



---

# IDP Series Feature Documentation



---

Published: 2012-11-28

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA  
408-745-2000  
www.juniper.net

Copyright © 2012, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

*IDP Series Feature Documentation*

Copyright © 2012, Juniper Networks, Inc.  
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

**END USER LICENSE AGREEMENT**

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

	About the Documentation . . . . .	xxv
	Documentation and Release Notes . . . . .	xxv
	Supported Platforms . . . . .	xxv
	Documentation Conventions . . . . .	xxv
	Documentation Feedback . . . . .	xxvii
	Requesting Technical Support . . . . .	xxvii
	Self-Help Online Tools and Resources . . . . .	xxviii
	Opening a Case with JTAC . . . . .	xxviii
<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Solution Overview . . . . .</b>	<b>3</b>
	Juniper Networks IDP Solutions . . . . .	3
	IDP Series Features Overview . . . . .	3
	IDP Series Operating System Overview . . . . .	7
	IDP Series Multicore Architecture . . . . .	7
	Auto-Recovery Feature . . . . .	9
	Flow Bypass Feature . . . . .	10
	Key Processes . . . . .	10
	IDP Series Network Interfaces Overview . . . . .	11
	Management Interface (eth0) . . . . .	13
	High Availability Interface (eth1) . . . . .	13
	Traffic Interfaces . . . . .	13
	Internal Bypass . . . . .	14
	External Bypass . . . . .	15
	Interface Signaling . . . . .	16
	Peer Port Modulation . . . . .	16
	Centralized Management with NSM Overview . . . . .	18
	J-Security Center Updates Overview . . . . .	19
<b>Chapter 2</b>	<b>Profiler and Monitoring Features Overview . . . . .</b>	<b>21</b>
	Profiler Overview . . . . .	21
	Application Volume Tracking Overview . . . . .	22
	IDP Logs Overview . . . . .	24
	NSM Reports Overview . . . . .	29
	IDP Reporter Overview . . . . .	31
<b>Chapter 3</b>	<b>Simulation Mode . . . . .</b>	<b>33</b>
	Simulation Mode Overview . . . . .	33
	Topology . . . . .	33
	Purpose . . . . .	33
	Configuration Overview . . . . .	34

	Logging .....	34
<b>Chapter 4</b>	<b>Security Policy Basics .....</b>	<b>37</b>
	Understanding Non-Policy-Based Drops .....	37
	Understanding the Components of an IDP Security Policy .....	41
	Understanding the Number of Available and Installed Policies .....	43
	Using Application Identification .....	43
	Understanding the Rule-Matching Algorithm .....	45
	Using the Recommended Security Policy .....	46
	Using Other Security Policy Templates .....	47
	Example: Fine-Tuning a Security Policy .....	48
	Fine-Tuning Security Policies Process Overview .....	48
	Getting Started with the Recommended Security Policy .....	49
	Refining Rule Matching Properties .....	50
	Reducing False Positives .....	50
	Adding Rulebases .....	53
<b>Chapter 5</b>	<b>The IDP Rulebase .....</b>	<b>55</b>
	Understanding the IDP Rulebase .....	55
	Understanding IDP Rulebase Rule Match Settings .....	56
	User-Role-Based Policy Feature Overview .....	58
	Using Attack Objects .....	60
	Attack Objects Overview .....	60
	Understanding Predefined Attack Objects and Attack Object Groups .....	61
	Using Attack Object Groups .....	61
	Using Custom Attack Objects .....	62
	Understanding IDP Rulebase Actions .....	63
	Understanding IDP Rulebase Notification Options .....	65
<b>Chapter 6</b>	<b>The Exempt Rulebase .....</b>	<b>67</b>
	Understanding the Exempt Rulebase .....	67
<b>Chapter 7</b>	<b>The APE Rulebase .....</b>	<b>69</b>
	Understanding the APE Rulebase .....	69
	Understanding APE Rulebase Match Conditions .....	70
	Using Application Objects .....	73
	Application Objects Overview .....	73
	Understanding Predefined Application Objects .....	73
	Using Application Groups .....	78
	Using Custom Application Objects .....	79
	Understanding APE Rulebase Actions .....	80
	Understanding APE Rulebase Notification Options .....	82
<b>Chapter 8</b>	<b>The Backdoor Rulebase .....</b>	<b>85</b>
	Understanding the Backdoor Rulebase .....	85
	Understanding Backdoor Rulebase Match Settings .....	87
	Understanding the Backdoor Rulebase Operation Setting .....	88
	Understanding Backdoor Rulebase Actions .....	88
	Understanding Backdoor Rulebase Notification Options .....	89

<b>Chapter 9</b>	<b>The SYN Protector Rulebase . . . . .</b>	<b>91</b>
	Understanding the SYN Protector Rulebase . . . . .	91
	Understanding SYN Protector Rulebase Match Settings . . . . .	93
	Understanding SYN Protector Rulebase Modes . . . . .	94
	Understanding SYN Protector Rulebase Notification Options . . . . .	95
<b>Chapter 10</b>	<b>The Traffic Anomalies Rulebase . . . . .</b>	<b>97</b>
	Understanding the Traffic Anomalies Rulebase . . . . .	97
	Understanding Traffic Anomalies Rulebase Match Conditions . . . . .	99
	Understanding Traffic Anomalies Rulebase Detection Settings . . . . .	100
	Understanding Traffic Anomalies Rulebase IP Actions . . . . .	100
	Understanding Traffic Anomalies Rulebase Notification Options . . . . .	101
<b>Chapter 11</b>	<b>The Network Honeypot Rulebase . . . . .</b>	<b>103</b>
	Understanding the Network Honeypot Rulebase . . . . .	103
	Understanding Network Honeypot Rulebase Match Settings . . . . .	104
	Understanding Network Honeypot Operation Setting . . . . .	104
	Understanding Network Honeypot Rulebase IP Actions . . . . .	105
	Understanding Network Honeypot Rulebase Notification Options . . . . .	106
<b>Chapter 12</b>	<b>Additional Security Features . . . . .</b>	<b>109</b>
	IP Spoof Attack Prevention Overview . . . . .	109
<b>Chapter 13</b>	<b>Inspection of Encapsulated and Encrypted Traffic . . . . .</b>	<b>111</b>
	Inspection of GRE Traffic Overview . . . . .	111
	Inspection of GTP Traffic Overview . . . . .	111
	Inspection of IPsec VPN Traffic Overview . . . . .	112
	Inspection of MPLS Traffic Overview . . . . .	112
	Inspection of SSL Traffic Overview . . . . .	113
	Using the SSL Server Private Keys . . . . .	113
	Using a Root Certificate Authority in SSL Forward Proxy Operations . . . . .	114
	Supported SSL Specifications . . . . .	115
<b>Part 2</b>	<b>Examples</b>	
<b>Chapter 14</b>	<b>Simulation Mode . . . . .</b>	<b>121</b>
	Example: Getting Started with Simulation Mode . . . . .	121
<b>Chapter 15</b>	<b>Using Profiler and Application Volume Tracking . . . . .</b>	<b>123</b>
	Example: Using Profiler to Set a Baseline . . . . .	124
	Example: Using Profiler to Alert You to New Hosts and Port Activity . . . . .	129
	Example: Identifying Services That Use Nonstandard Ports . . . . .	129
	Example: Responding to Vulnerability Announcements with Due Diligence . . . . .	130
	Example: Using Profiler to Investigate Unanticipated Attacks . . . . .	131
	Example: Using Profiler to Mitigate Risks from Laptops . . . . .	132
	Example: Using NSM to Enable and View Application Volume Tracking . . . . .	133
<b>Chapter 16</b>	<b>Logging . . . . .</b>	<b>139</b>
	Example: Using NSM Log Viewer Features . . . . .	139
	Using Predefined Views . . . . .	139
	Showing and Hiding Columns . . . . .	140

	Using Filters . . . . .	141
	Using Log Viewer Detail Panes . . . . .	142
	Using Flags and Comments . . . . .	143
	Using Custom Views . . . . .	144
	Example: Packet Logging Workflow . . . . .	145
	Using Packet Captures . . . . .	145
	Enabling Packet Capture in Security Policy Rules . . . . .	145
	Forwarding Packet Capture Logs to NSM . . . . .	146
	Viewing Packet Capture Logs . . . . .	147
	Using the NSM Packet Viewer . . . . .	147
	Using an External Viewer to View Packet Data . . . . .	148
	Example: Querying the IDP Series Device MIB . . . . .	151
<b>Chapter 17</b>	<b>IDP Rulebase Examples . . . . .</b>	<b>153</b>
	IDP Rulebase Example: Using Application Identification . . . . .	153
	IDP Rulebase Example: Specifying the Default Service . . . . .	154
	IDP Rulebase Example: Using Recommended Attack Objects . . . . .	155
	IDP Rulebase Example: Using Recommended Actions . . . . .	156
	IDP Rulebase Example: User-Role-Based Policies . . . . .	157
	Example: Fine-Tuning a Security Policy . . . . .	160
	Fine-Tuning Security Policies Process Overview . . . . .	160
	Getting Started with the Recommended Security Policy . . . . .	160
	Refining Rule Matching Properties . . . . .	161
	Reducing False Positives . . . . .	162
	Adding Rulebases . . . . .	164
<b>Chapter 18</b>	<b>APE Rulebase Examples . . . . .</b>	<b>165</b>
	APE Rulebase Example: Using Extended Application Objects . . . . .	165
	APE Rulebase Example: Aggregate and Per-Subscriber Rate Limits . . . . .	170
	APE Rulebase Example: Matching Custom Application Objects . . . . .	171
<b>Chapter 19</b>	<b>Exempt and Backdoor Rulebase Examples . . . . .</b>	<b>175</b>
	Exempt Rulebase Example: Exempting a Source Destination Pair . . . . .	175
	Exempt Rulebase Example: Exempting an Attack Object . . . . .	176
	Backdoor Rulebase Example: netcat . . . . .	176
<b>Chapter 20</b>	<b>Inspection of HTTPS Traffic . . . . .</b>	<b>179</b>
	Example: Implementing Inspection of Outbound SSL Traffic . . . . .	179
	Example: Exempting Outbound SSL Traffic from Inspection . . . . .	181
<b>Part 3</b>	<b>Configuration</b>	
<b>Chapter 21</b>	<b>Getting Started . . . . .</b>	<b>189</b>
	Supported Tools for Management Tasks . . . . .	189
	Connecting to ACM . . . . .	191
	Connecting to the Command-Line Interface (CLI Procedure) . . . . .	192
	Configuring Virtual Routers (ACM Procedure) . . . . .	192
	Getting Started with the Default Configuration . . . . .	194
	Developing Security Policies Task Summary . . . . .	195
	Using Predefined Security Policies . . . . .	197
	Using the New Policy Wizard (NSM Procedure) . . . . .	198

<b>Chapter 22</b>	<b>Simulation Mode</b> . . . . .	<b>201</b>
	Enabling Simulation Mode . . . . .	201
<b>Chapter 23</b>	<b>Configuring Profiler</b> . . . . .	<b>203</b>
	Profiler Task Summary . . . . .	203
	Configuring Profiler Options (NSM Procedure) . . . . .	204
	Configuring General Settings . . . . .	204
	Configuring Tracked Hosts . . . . .	205
	Configuring Context Targets . . . . .	208
	Configuring Alert Options . . . . .	209
	Modifying Profiler Settings . . . . .	210
<b>Chapter 24</b>	<b>Configuring the IDP Rulebase</b> . . . . .	<b>213</b>
	Modifying IDP Rulebase Rules (NSM Procedure) . . . . .	213
	Specifying Rule Match Conditions (NSM Procedure) . . . . .	215
	Specifying IDP Rulebase Attack Objects (NSM Procedure) . . . . .	216
	Specifying Rule Session Action (NSM Procedure) . . . . .	218
	Specifying IP Action (NSM Procedure) . . . . .	220
	Specifying Rule Notification Options (NSM Procedure) . . . . .	221
	Specifying Rule VLAN Matches (NSM Procedure) . . . . .	223
	Specifying Rule Targets (NSM Procedure) . . . . .	223
	Specifying Rule Severity (NSM Procedure) . . . . .	224
	Specifying Rule Comments (NSM Procedure) . . . . .	225
<b>Chapter 25</b>	<b>Configuring Additional Security Policy Rulebases</b> . . . . .	<b>227</b>
	Configuring Exempt Rulebase Rules (NSM Procedure) . . . . .	227
	Configuring the APE Rulebase (NSM Procedure) . . . . .	228
	Configuring Backdoor Rulebase Rules (NSM Procedure) . . . . .	233
	Configuring SYN Protector Rulebase Rules (NSM Procedure) . . . . .	235
	Configuring Traffic Anomalies Rulebase Rules (NSM Procedure) . . . . .	237
	Configuring Network Honeypot Rulebase Rules (NSM Procedure) . . . . .	240
<b>Chapter 26</b>	<b>Working with Attack Objects</b> . . . . .	<b>243</b>
	Using Attack Objects . . . . .	243
	Attack Objects Overview . . . . .	243
	Understanding Predefined Attack Objects and Attack Object Groups . . . . .	244
	Using Attack Object Groups . . . . .	245
	Using Custom Attack Objects . . . . .	246
	Attack Objects Task Summary . . . . .	246
	Viewing Predefined Attack Objects (NSM Procedure) . . . . .	247
	Working with Attack Groups (NSM Procedure) . . . . .	251
	Creating Dynamic Groups . . . . .	251
	Creating Static Groups . . . . .	252
	Creating a Signature Attack Object . . . . .	253
	Creating a Compound Attack Object . . . . .	273
<b>Chapter 27</b>	<b>Working with Application Objects</b> . . . . .	<b>279</b>
	Using Application Objects . . . . .	279
	Application Objects Overview . . . . .	279
	Understanding Predefined Application Objects . . . . .	280
	Using Application Groups . . . . .	284

	Using Custom Application Objects . . . . .	285
	Application Objects Task Summary . . . . .	286
	Viewing Predefined Application Objects (NSM Procedure) . . . . .	287
	Viewing Predefined Extended Application Objects (NSM Procedure) . . . . .	290
	Creating Application Groups (NSM Procedure) . . . . .	292
	Creating a Custom Application (NSM Procedure) . . . . .	292
<b>Chapter 28</b>	<b>Configuring Logging Features . . . . .</b>	<b>295</b>
	IDP Series Logs and Reports in NSM Task Summary . . . . .	295
	Configuring Interface Aliasing (ACM Procedure) . . . . .	296
	Configuring Log Storage Limits . . . . .	297
	Configuring Log Suppression (NSM Procedure) . . . . .	298
	Configuring an SNMP Agent (NSM Procedure) . . . . .	299
	Configuring Syslog Collection (NSM Procedure) . . . . .	301
	Enabling Collection of Packet Data in NSM Logs (NSM Procedure) . . . . .	303
<b>Chapter 29</b>	<b>Using the scio Command to Implement Advanced Features . . . . .</b>	<b>307</b>
	scio Configuration Commands Task Summary . . . . .	307
	Using the SSL Private Server Key to Enable Inspection of SSL Traffic . . . . .	308
	Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic . . . . .	311
	Exempting HTTPS Traffic from Inspection . . . . .	312
	Enabling Inspection of GRE Traffic . . . . .	313
	Enabling Inspection of GTP Traffic . . . . .	315
	Enabling Inspection of IPsec VPN Traffic . . . . .	317
	Enabling Inspection of MPLS Traffic . . . . .	318
	Enabling the Flow Bypass Feature . . . . .	319
	Configuring a Default Rate Limit . . . . .	321
	Enabling Per-User Rate Limiting for User-Role-Based Rules . . . . .	321
	Configuring Advanced Settings for the User-Role-Based Policy Feature . . . . .	322
<b>Part 4</b>	<b>Administration</b>	
<b>Chapter 30</b>	<b>Managing the Profiler . . . . .</b>	<b>327</b>
	Profiler Task Summary . . . . .	327
	Starting and Stopping the Profiler (NSM Procedure) . . . . .	328
	Managing the Profiler Database (NSM Procedure) . . . . .	328
	Displaying Profiler Database Information . . . . .	328
	Querying the Profiler Database . . . . .	328
	Purging the Profiler Database . . . . .	329
<b>Chapter 31</b>	<b>Logging . . . . .</b>	<b>331</b>
	Developing a Logging Strategy . . . . .	331
	Developing a Log Storage Strategy . . . . .	332
	Log Management Considerations . . . . .	332
	Local Log Files and Directories . . . . .	332
	NSM Log Collection . . . . .	334



<b>Chapter 32</b>	<b>Managing Security Policies . . . . .</b>	<b>335</b>
	Managing Security Policies Task Summary . . . . .	335
	Assigning a Security Policy to a Device (NSM Procedure) . . . . .	335
	Validating a Security Policy (NSM Procedure) . . . . .	336
	Loading J-Security Center Updates (NSM Procedure) . . . . .	336
	Pushing Security Policy Updates to an IDP Series Device (NSM Procedure) . . . . .	340
	Disabling Rules (NSM Procedure) . . . . .	342
	Exporting Security Policies (NSM Procedure) . . . . .	342
<b>Chapter 33</b>	<b>Managing the IDP Device Configuration with NSM . . . . .</b>	<b>343</b>
	NSM Device Configuration Management Task Summary . . . . .	343
	Adding IDP Series Devices to NSM Device Manager . . . . .	344
	Adding a Reachable Device . . . . .	344
	Adding an Unreachable Device . . . . .	345
	Modeling an IDP Series Device Configuration . . . . .	346
	Adding Device Clusters . . . . .	347
	Activating Devices (NSM Procedure) . . . . .	348
	Activating a Reachable IDP Series Device . . . . .	348
	Activating an Unreachable IDP Series Device . . . . .	349
	Pulling or Pushing Configuration Updates . . . . .	350
	Modifying the IDP Series Device Configuration . . . . .	351
	Modifying NSM Informational Properties . . . . .	352
	Modifying Antispoof Settings . . . . .	353
	Modifying Runtime Parameters . . . . .	355
	Modifying Load-Time Parameters . . . . .	365
	Modifying Protocol Anomaly Thresholds . . . . .	367
	Deleting an IDP Series Device Configuration from NSM Device Manager (NSM Procedure) . . . . .	382
<b>Chapter 34</b>	<b>Managing IDP Processes . . . . .</b>	<b>383</b>
	Restarting the IDP Engine . . . . .	383
	Rebooting and Shutting Down the IDP Series Appliance . . . . .	384
	idp.sh Command Reference . . . . .	385
<b>Chapter 35</b>	<b>Updating IDP Software . . . . .</b>	<b>389</b>
	Upgrading Software (CLI Procedure) . . . . .	389
	Updating IDP OS Software (NSM Procedure) . . . . .	390
	Loading J-Security Center Updates (NSM Procedure) . . . . .	392
<b>Chapter 36</b>	<b>Installing Traffic Interface I/O Modules . . . . .</b>	<b>397</b>
	Installing an I/O Module . . . . .	397
<b>Chapter 37</b>	<b>Enabling Bypass and Peer Port Modulation . . . . .</b>	<b>401</b>
	Configuring Virtual Routers (ACM Procedure) . . . . .	401
	Enabling the Flow Bypass Feature . . . . .	402
<b>Chapter 38</b>	<b>Configuring the Management Interface . . . . .</b>	<b>405</b>
	Changing the Management Interface IP Address . . . . .	405

<b>Part 5</b>	<b>Monitoring</b>	
<b>Chapter 39</b>	<b>Overview</b>	<b>411</b>
	Supported Tools for Monitoring Tasks	411
	Developing a Logging Strategy	412
	Developing a Log Storage Strategy	412
	Log Management Considerations	413
	Local Log Files and Directories	413
	NSM Log Collection	414
<b>Chapter 40</b>	<b>Using SNMP</b>	<b>417</b>
	SNMP Statistic Reporting and Traps Task Summary	417
	Configuring SNMP Reporting for the Interface Category of Statistics	418
	Configuring SNMP Reporting for the Resource Category of Statistics	428
	Configuring SNMP Reporting for the Rule Category of Statistics	433
	Configuring SNMP Reporting for the Sensor Category of Statistics	435
	Configuring SNMP Reporting for the Traffic Category of Statistics	438
<b>Chapter 41</b>	<b>Using NSM Logs and Reports</b>	<b>447</b>
	IDP Series Logs and Reports in NSM Task Summary	447
	Viewing Device Status (NSM Procedure)	448
	Using NSM Logs	453
	NSM Logs Overview	453
	Using NSM Log Viewer (NSM Procedure)	454
	Using NSM Log Investigator (NSM Procedure)	459
	Using NSM Audit Log Viewer (NSM Procedure)	460
	Viewing Simulation Mode Logs	461
	Using Profiler Viewer (NSM Procedure)	463
	Application Profiler Tab	463
	Protocol Profiler Tab	465
	Network Profiler Tab	466
	Violation Viewer Tab	468
	Viewing NSM Predefined Reports (NSM Procedure)	469
	Creating NSM Custom Reports (NSM Procedure)	472
<b>Chapter 42</b>	<b>Packet Logging</b>	<b>475</b>
	Example: Packet Logging Workflow	475
	Using Packet Captures	475
	Enabling Packet Capture in Security Policy Rules	476
	Forwarding Packet Capture Logs to NSM	477
	Viewing Packet Capture Logs	477
	Using the NSM Packet Viewer	478
	Using an External Viewer to View Packet Data	478
	Using tcpdump to Capture Packets	481
	Using jnetTcpdump to Capture Packets	481
<b>Chapter 43</b>	<b>Using the bypassStatus Utility to Monitor the Internal Bypass Daemon</b>	<b>483</b>
	bypassStatus Utility Task Summary	483
	bypassStatus Command Reference	484

<b>Chapter 44</b>	<b>Using the sctop Utility to Monitor Session Flow . . . . .</b>	<b>487</b>
	sctop Task Summary . . . . .	487
	Using the sctop Utility (CLI Procedure) . . . . .	487
	Understanding sctop Flow Table Reports . . . . .	490
<b>Chapter 45</b>	<b>Using the scio Utility to Verify Feature Implementation . . . . .</b>	<b>491</b>
	scio Monitoring Commands Task Summary . . . . .	491
	Verifying the APE Rulebase . . . . .	491
	Verifying Integration with an IC Series Unified Access Control Appliance . . . . .	493
	Verifying MPLS Decapsulation . . . . .	494
	Verifying the Flow Bypass Feature . . . . .	495
<b>Chapter 46</b>	<b>scio Commands . . . . .</b>	<b>499</b>
	scio agentconfig . . . . .	500
	scio app cache . . . . .	502
	scio app sig list . . . . .	504
	scio const . . . . .	505
	scio counter . . . . .	522
	scio getsystem . . . . .	524
	scio idp-cpu-utilization . . . . .	525
	scio logview . . . . .	526
	scio napp sig list . . . . .	527
	scio nic . . . . .	528
	scio sri . . . . .	529
	scio ssl . . . . .	531
	scio subs . . . . .	535
	scio sysconf . . . . .	539
	scio user . . . . .	544
	scio var . . . . .	546
	scio vc . . . . .	549
	scio version . . . . .	550
	scio vr . . . . .	551
<b>Chapter 47</b>	<b>IDP MIB Object ID Reference . . . . .</b>	<b>553</b>
	IDP Series MIB Object ID Reference . . . . .	553
<b>Part 6</b>	<b>Troubleshooting</b>	
<b>Chapter 48</b>	<b>Troubleshooting References . . . . .</b>	<b>575</b>
	Troubleshooting Tools Overview . . . . .	575
	IDP Processes Reference . . . . .	577
	Troubleshooting NSM Log Collection Issues . . . . .	578
<b>Chapter 49</b>	<b>Simulation Mode . . . . .</b>	<b>579</b>
	Example: Using Simulation Mode to Maximize Uptime . . . . .	579
<b>Chapter 50</b>	<b>Troubleshooting Feature Implementation . . . . .</b>	<b>581</b>
	Tuning the JNET Driver Failure Count . . . . .	581
	Viewing Auto-Recovery Logs . . . . .	583
	Disabling the Auto-Recovery Feature . . . . .	584
	Tuning the Auto-Recovery Policy Reload Setting . . . . .	584

Troubleshooting SNMP Statistic Reporting . . . . .	585
Viewing CPU Utilization . . . . .	586
Troubleshooting High CPU Usage . . . . .	588
Troubleshooting Erroneous CPU Utilization Reports . . . . .	590
Displaying Service Session Count . . . . .	592
Troubleshooting Configuration Push Errors (NSM Procedure) . . . . .	593
Troubleshooting Security Policy Validation Errors (NSM Procedure) . . . . .	594
Troubleshooting Application Identification . . . . .	595
Disabling the APE Rulebase . . . . .	598
Disabling the User Role-Based Policy Feature . . . . .	599
Disabling Support for Jumbo Frames . . . . .	599
Troubleshooting SSL Inspection . . . . .	600
Disabling SSL Inspection . . . . .	600
Disabling MPLS Decapsulation . . . . .	601

## Part 7

## Index

Index . . . . .	605
-----------------	-----

# List of Figures

<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Solution Overview</b>	<b>3</b>
	Figure 1: IDP Multicore Architecture	8
	Figure 2: IDP Series Network Interfaces	12
	Figure 3: Internal Bypass	15
	Figure 4: External Bypass	16
	Figure 5: Peer Port Modulation	17
	Figure 6: IDP-NSM Communication	18
<b>Chapter 3</b>	<b>Simulation Mode</b>	<b>33</b>
	Figure 7: Packet Processing in Simulation Mode	34
	Figure 8: NSM Log Viewer: Simulation Mode Logs	35
<b>Chapter 4</b>	<b>Security Policy Basics</b>	<b>37</b>
	Figure 9: Security Policy Components	41
	Figure 10: Using NSM Log Viewer Attack Reference Information	51
	Figure 11: Using NSM Log Viewer Flag and Comment Features	52
<b>Chapter 7</b>	<b>The APE Rulebase</b>	<b>69</b>
	Figure 12: NSM Object Manager: Predefined Application Objects	74
	Figure 13: NSM Object Manager: Predefined Application: General Tab	75
	Figure 14: NSM Object Manager: Predefined Application: Detector Tab	76
	Figure 15: NSM Object Manager: Predefined Extended Application Objects	77
	Figure 16: NSM Object Manager: Extended Application Details	78
	Figure 17: NSM Object Manager: Extended Application Member Details	78
	Figure 18: NSM Object Manager: Application Group Dialog Box	79
	Figure 19: NSM Object Manager: Custom Application Dialog Box	80
<b>Chapter 8</b>	<b>The Backdoor Rulebase</b>	<b>85</b>
	Figure 20: NSM Device Manager: Sensor Settings > Run-Time Parameters	86
<b>Chapter 9</b>	<b>The SYN Protector Rulebase</b>	<b>91</b>
	Figure 21: NSM Device Manager: Sensor Settings > Run-Time Parameters	92
<b>Chapter 13</b>	<b>Inspection of Encapsulated and Encrypted Traffic</b>	<b>111</b>
	Figure 22: SSL Inspection Using SSL Server Private Keys	114
	Figure 23: SSL Inspection Using a Root CA	114
<b>Part 2</b>	<b>Examples</b>	
<b>Chapter 15</b>	<b>Using Profiler and Application Volume Tracking</b>	<b>123</b>
	Figure 24: NSM Network Address Object Editor	124

	Figure 25: NSM Group Object Editor . . . . .	125
	Figure 26: Starting Profiler from NSM Device Manager . . . . .	125
	Figure 27: NSM Profiler Configuration Tabs . . . . .	126
	Figure 28: NSM Profiler Tracked Hosts Tab . . . . .	126
	Figure 29: NSM Profiler Update Job Information Window . . . . .	127
	Figure 30: Profiler: Network Profiler Tab . . . . .	128
	Figure 31: Profiler Settings: Enable AVT . . . . .	134
	Figure 32: Profiler Viewer: Application Profiler Tab . . . . .	134
	Figure 33: Profiler Viewer: Application Profiler Tab: Nested Applications . . . . .	135
	Figure 34: NSM AVT Report . . . . .	136
<b>Chapter 16</b>	<b>Logging . . . . .</b>	<b>139</b>
	Figure 35: NSM Log Viewer: Predefined View . . . . .	140
	Figure 36: NSM Log Viewer: Choose Columns . . . . .	140
	Figure 37: NSM Log Viewer: Filters . . . . .	141
	Figure 38: NSM Log Viewer: Filters . . . . .	142
	Figure 39: Using NSM Log Viewer Attack Reference Information . . . . .	142
	Figure 40: Using NSM Log Viewer Flag and Comment Features . . . . .	143
	Figure 41: NSM Log Viewer: Custom View . . . . .	144
	Figure 42: Notification Options: Packet Logging . . . . .	146
	Figure 43: NSM Log Viewer: Has Packet Data Column . . . . .	146
	Figure 44: NSM Device Configuration Editor: Report Settings . . . . .	147
	Figure 45: NSM Packet Capture Viewer . . . . .	148
	Figure 46: Specifying an External Viewer . . . . .	149
	Figure 47: Wireshark Packet Viewer . . . . .	150
<b>Chapter 17</b>	<b>IDP Rulebase Examples . . . . .</b>	<b>153</b>
	Figure 48: A Simplified Rule Enabled by the Application Identification Feature . . . . .	154
	Figure 49: Default Service . . . . .	154
	Figure 50: Recommended Attack Objects . . . . .	155
	Figure 51: Recommended Action . . . . .	157
	Figure 52: IC Series Admin Console: Configuring User Roles . . . . .	158
	Figure 53: ACM: Generating a One-Time Password for the Connection from the IC Series Appliance . . . . .	158
	Figure 54: IC Series Admin Console: Configuring the Connection to the IDP Appliance . . . . .	159
	Figure 55: IDP Rulebase: User-Role-Based Rules . . . . .	159
	Figure 56: Using NSM Log Viewer Attack Reference Information . . . . .	162
	Figure 57: Using NSM Log Viewer Flag and Comment Features . . . . .	163
<b>Chapter 18</b>	<b>APE Rulebase Examples . . . . .</b>	<b>165</b>
	Figure 58: NSM Object Manager: Predefined Extended Application Objects . . . . .	166
	Figure 59: NSM Object Manager: Extended Application Details . . . . .	167
	Figure 60: NSM Object Manager: Extended Application Details . . . . .	168
	Figure 61: APE Rulebase: Using Extended Applications . . . . .	168
	Figure 62: NSM Object Manager: Creating Application Groups . . . . .	169
	Figure 63: NSM Object Manager: Creating Application Groups . . . . .	169
	Figure 64: APE Rulebase: Using Application Groups . . . . .	169
	Figure 65: APE Rulebase: User-Role-Based Rules to Support Tiered Access . . . . .	170

	Figure 66: APE Rulebase: User-Role-Based Rules to Support Tiered Access . . .	171
	Figure 67: NSM Object Manager: Custom Application Object . . . . .	172
	Figure 68: NSM Object Manager: Custom Application Object . . . . .	173
	Figure 69: APE Rulebase: Adding a Custom Application Object . . . . .	173
	Figure 70: APE Rulebase: Rule Order . . . . .	174
<b>Chapter 19</b>	<b>Exempt and Backdoor Rulebase Examples . . . . .</b>	<b>175</b>
	Figure 71: Exempt Rulebase Rule . . . . .	175
	Figure 72: Backdoor Rulebase . . . . .	177
<b>Chapter 20</b>	<b>Inspection of HTTPS Traffic . . . . .</b>	<b>179</b>
	Figure 73: Firefox: Displaying the Server Certificate for a Website . . . . .	183
	Figure 74: Internet Explorer: Displaying the Server Certificate for a Website . . . .	184
<b>Part 3</b>	<b>Configuration</b>	
<b>Chapter 21</b>	<b>Getting Started . . . . .</b>	<b>189</b>
	Figure 75: ACM Configure Virtual Routers Page . . . . .	193
<b>Chapter 23</b>	<b>Configuring Profiler . . . . .</b>	<b>203</b>
	Figure 76: Profiler Settings: Enable AVT . . . . .	204
	Figure 77: NSM Profiler Tracked Hosts Tab . . . . .	206
	Figure 78: NSM Profiler Context to Profile Tab . . . . .	208
	Figure 79: Profiler Alert Tab . . . . .	209
	Figure 80: New Preferences Dialog Box . . . . .	210
<b>Chapter 24</b>	<b>Configuring the IDP Rulebase . . . . .</b>	<b>213</b>
	Figure 81: NSM Security Policy Editor: IDP Rulebase . . . . .	213
<b>Chapter 25</b>	<b>Configuring Additional Security Policy Rulebases . . . . .</b>	<b>227</b>
	Figure 82: NSM Security Policy Editor: Exempt Rulebase . . . . .	227
	Figure 83: NSM Security Policy Editor: APE Rulebase . . . . .	228
	Figure 84: NSM Security Policy Editor: Backdoor Rulebase . . . . .	233
	Figure 85: NSM Device Manager: Sensor Settings > Run-Time Parameters . . . .	235
	Figure 86: NSM Security Policy Editor: SYN Protector Rulebase . . . . .	236
	Figure 87: NSM Device Manager: Sensor Settings > Run-Time Parameters . . . .	237
	Figure 88: NSM Security Policy Editor: Traffic Anomalies Rulebase . . . . .	238
	Figure 89: NSM Security Policy Editor: Network Honeypot Rulebase . . . . .	240
<b>Chapter 26</b>	<b>Working with Attack Objects . . . . .</b>	<b>243</b>
	Figure 90: NSM Object Manager: Predefined Attack Objects . . . . .	248
	Figure 91: NSM Object Manager Predefined Attack Object Details . . . . .	249
	Figure 92: NSM Object Manager Predefined Attack Object Extended Details . .	250
	Figure 93: Custom Attack Object: General Tab . . . . .	254
	Figure 94: Custom Attack Object: Extended Tab . . . . .	255
	Figure 95: Custom Attack: Target Platform and Type Page . . . . .	257
	Figure 96: Custom Attack - General Properties Page . . . . .	259
	Figure 97: Custom Attack – Attack Pattern Page . . . . .	263
	Figure 98: Custom Attack – IP Settings and Header Matches Page . . . . .	267
	Figure 99: Custom Attack Object: TCP Packet Header Fields . . . . .	269
	Figure 100: Custom Attack Object: UDP Packet Header Fields . . . . .	271
	Figure 101: Custom Attack Object: ICMP Packet Header Fields . . . . .	272

	Figure 102: Custom Attack – Compound Members . . . . .	275
<b>Chapter 27</b>	<b>Working with Application Objects . . . . .</b>	<b>279</b>
	Figure 103: NSM Object Manager: Predefined Application Objects . . . . .	280
	Figure 104: NSM Object Manager: Predefined Application: General Tab . . . . .	281
	Figure 105: NSM Object Manager: Predefined Application: Detector Tab . . . . .	282
	Figure 106: NSM Object Manager: Predefined Extended Application Objects . . . . .	283
	Figure 107: NSM Object Manager: Extended Application Details . . . . .	284
	Figure 108: NSM Object Manager: Extended Application Member Details . . . . .	284
	Figure 109: NSM Object Manager: Application Group Dialog Box . . . . .	285
	Figure 110: NSM Object Manager: Custom Application Dialog Box . . . . .	286
	Figure 111: NSM Object Manager: Predefined Application Objects . . . . .	288
	Figure 112: NSM Object Manager: Predefined Application: General Tab . . . . .	288
	Figure 113: NSM Object Manager: Predefined Application: Detector Tab . . . . .	289
	Figure 114: NSM Object Manager: Predefined Extended Application Objects . . . . .	290
	Figure 115: NSM Object Manager: Extended Application Details . . . . .	291
	Figure 116: NSM Object Manager: New Application Group . . . . .	292
	Figure 117: NSM Object Manager: New Custom Application Dialog Box . . . . .	293
<b>Chapter 28</b>	<b>Configuring Logging Features . . . . .</b>	<b>295</b>
	Figure 118: ACM Configure Network Interface Hardware Page . . . . .	296
	Figure 119: NSM Device Configuration Editor: Report Settings . . . . .	297
	Figure 120: NSM Device Configuration Editor: Report Settings . . . . .	300
	Figure 121: NSM Device Configuration Editor: Report Settings . . . . .	302
	Figure 122: NSM Device Configuration Editor: Report Settings . . . . .	304
<b>Chapter 29</b>	<b>Using the scio Command to Implement Advanced Features . . . . .</b>	<b>307</b>
	Figure 123: NSM Device Manager: SSL Decryption Setting . . . . .	310
	Figure 124: NSM Device Manager: GRE Support Setting . . . . .	314
	Figure 125: NSM Device Manager: GTP Support Setting . . . . .	316
<b>Part 4</b>	<b>Administration</b>	
<b>Chapter 31</b>	<b>Logging . . . . .</b>	<b>331</b>
	Figure 126: IDP Log Storage and Log Forwarding . . . . .	333
<b>Chapter 33</b>	<b>Managing the IDP Device Configuration with NSM . . . . .</b>	<b>343</b>
	Figure 127: NSM Device Configuration Editor: Info Page . . . . .	352
	Figure 128: NSM Device Configuration Editor: Anti-Spoof Settings Page . . . . .	354
	Figure 129: NSM Device Configuration Editor: Run-time Parameters Tab . . . . .	355
	Figure 130: NSM Device Configuration Editor: Load Time Parameters Tab . . . . .	366
	Figure 131: NSM Device Configuration Editor: Protocol Thresholds and Configuration Tab . . . . .	368
<b>Chapter 36</b>	<b>Installing Traffic Interface I/O Modules . . . . .</b>	<b>397</b>
	Figure 132: I/O Module Blank . . . . .	397
	Figure 133: IDP-1GE-4COP-BYP . . . . .	397
	Figure 134: IDP-1GE-4SFP* . . . . .	398
	Figure 135: IDP-1GE-4SX-BYP . . . . .	398
	Figure 136: IDP-10GE-2XFP* . . . . .	398
	Figure 137: IDP-10GE-2SR-BYP . . . . .	398



	Figure 138: Replacing a Blank Tray with an I/O Module Tray . . . . .	399
<b>Chapter 37</b>	<b>Enabling Bypass and Peer Port Modulation . . . . .</b>	<b>401</b>
	Figure 139: ACM Configure Virtual Routers Page . . . . .	401
<b>Part 5</b>	<b>Monitoring</b>	
<b>Chapter 39</b>	<b>Overview . . . . .</b>	<b>411</b>
	Figure 140: IDP Log Storage and Log Forwarding . . . . .	414
<b>Chapter 41</b>	<b>Using NSM Logs and Reports . . . . .</b>	<b>447</b>
	Figure 141: NSM Device Monitor . . . . .	448
	Figure 142: NSM Device Detail . . . . .	450
	Figure 143: NSM Device Monitor: Process Status Page . . . . .	452
	Figure 144: NSM Device Monitor: Device Statistics . . . . .	453
	Figure 145: NSM Log Viewer . . . . .	454
	Figure 146: NSM Packet Viewer . . . . .	458
	Figure 147: NSM Log Viewer: Simulation Mode Logs . . . . .	462
	Figure 148: Profiler Viewer: Application Profiler Tab . . . . .	463
	Figure 149: Profiler Viewer: Application Profiler Tab: Nested Applications . . . . .	463
	Figure 150: Profiler Viewer: Protocol Profiler Tab . . . . .	465
	Figure 151: Profiler Viewer: Network Profiler Tab . . . . .	467
	Figure 152: Profiler Viewer: Violation Viewer Tab . . . . .	469
<b>Chapter 42</b>	<b>Packet Logging . . . . .</b>	<b>475</b>
	Figure 153: Notification Options: Packet Logging . . . . .	476
	Figure 154: NSM Log Viewer: Has Packet Data Column . . . . .	476
	Figure 155: NSM Device Configuration Editor: Report Settings . . . . .	477
	Figure 156: NSM Packet Capture Viewer . . . . .	478
	Figure 157: Specifying an External Viewer . . . . .	479
	Figure 158: Wireshark Packet Viewer . . . . .	480
<b>Chapter 45</b>	<b>Using the scio Utility to Verify Feature Implementation . . . . .</b>	<b>491</b>
	Figure 159: NSM Log Viewer: MPLS Label Information . . . . .	495
<b>Part 6</b>	<b>Troubleshooting</b>	
<b>Chapter 50</b>	<b>Troubleshooting Feature Implementation . . . . .</b>	<b>581</b>
	Figure 160: SNMP Statistic Report Processes . . . . .	585



# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xxv</b>
	Table 1: Notice Icons . . . . .	xxvi
	Table 2: Text and Syntax Conventions . . . . .	xxvi
<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Solution Overview</b> . . . . .	<b>3</b>
	Table 3: IDP Series Features . . . . .	3
	Table 4: Intrusion Detection Methods . . . . .	6
	Table 5: IDP Multicore Architecture . . . . .	8
	Table 6: Processes . . . . .	10
<b>Chapter 2</b>	<b>Profiler and Monitoring Features Overview</b> . . . . .	<b>21</b>
	Table 7: Application Volume Tracking Data . . . . .	23
	Table 8: Application Volume Tracking Log Viewing Tools . . . . .	23
	Table 9: IDP Logging Options . . . . .	24
	Table 10: NSM Log Viewer: Log Columns . . . . .	25
	Table 11: NSM DI/IDP Predefined Reports . . . . .	29
	Table 12: NSM Profiler Predefined Reports . . . . .	30
	Table 13: NSM: Application Volume Tracking Reports . . . . .	31
<b>Chapter 4</b>	<b>Security Policy Basics</b> . . . . .	<b>37</b>
	Table 14: Non-Policy-Based Drops . . . . .	38
	Table 15: IDP Security Policy Rulebases . . . . .	41
	Table 16: Recommended Security Policy Definition . . . . .	46
	Table 17: IDP Security Policy Templates . . . . .	47
	Table 18: Recommended Security Policy Definition . . . . .	49
	Table 19: Actions to Take To Reduce False Positives . . . . .	52
<b>Chapter 5</b>	<b>The IDP Rulebase</b> . . . . .	<b>55</b>
	Table 20: IDP Rulebase Match Condition Guidelines . . . . .	56
	Table 21: Predefined Attack Object Groups . . . . .	61
	Table 22: Recommended Action by Attack Severity . . . . .	63
	Table 23: IDP Rulebase Actions . . . . .	64
	Table 24: IDP Rulebase IP Actions . . . . .	65
	Table 25: IDP Rulebase Notification Options . . . . .	66
<b>Chapter 7</b>	<b>The APE Rulebase</b> . . . . .	<b>69</b>
	Table 26: APE Rulebase Match Condition Guidelines . . . . .	71
	Table 27: IDP Rulebase Actions . . . . .	81
	Table 28: APE Rulebase Notification Options . . . . .	83
<b>Chapter 8</b>	<b>The Backdoor Rulebase</b> . . . . .	<b>85</b>

	Table 29: Backdoor Detection Runtime Parameters . . . . .	86
	Table 30: Backdoor Rulebase Actions . . . . .	89
	Table 31: Backdoor Rulebase Notification Options . . . . .	89
<b>Chapter 9</b>	<b>The SYN Protector Rulebase . . . . .</b>	<b>91</b>
	Table 32: SYN Protector Thresholds . . . . .	92
	Table 33: SYN Flood Detection Runtime Parameters . . . . .	93
	Table 34: SYN Protector Rulebase Modes . . . . .	94
	Table 35: SYN Protector Rulebase Notification Options . . . . .	96
<b>Chapter 10</b>	<b>The Traffic Anomalies Rulebase . . . . .</b>	<b>97</b>
	Table 36: Traffic Anomalies Rulebase Detection Settings . . . . .	97
	Table 37: Traffic Signature Runtime Settings . . . . .	98
	Table 38: Traffic Anomalies Rulebase IP Actions . . . . .	101
	Table 39: Traffic Anomalies Rulebase Notification Options . . . . .	102
<b>Chapter 11</b>	<b>The Network Honeypot Rulebase . . . . .</b>	<b>103</b>
	Table 40: Network Honeypot Rulebase IP Actions . . . . .	105
	Table 41: Network Honeypot Rulebase Notification Options . . . . .	106
<b>Chapter 13</b>	<b>Inspection of Encapsulated and Encrypted Traffic . . . . .</b>	<b>111</b>
	Table 42: Supported SSL Cipher Suites . . . . .	115
<b>Part 2</b>	<b>Examples</b>	
<b>Chapter 15</b>	<b>Using Profiler and Application Volume Tracking . . . . .</b>	<b>123</b>
	Table 43: Application Profiler Session Table . . . . .	135
	Table 44: NSM: Application Volume Tracking Reports . . . . .	137
<b>Chapter 17</b>	<b>IDP Rulebase Examples . . . . .</b>	<b>153</b>
	Table 45: Recommended Security Policy Definition . . . . .	161
	Table 46: Actions to Take To Reduce False Positives . . . . .	163
<b>Part 3</b>	<b>Configuration</b>	
<b>Chapter 21</b>	<b>Getting Started . . . . .</b>	<b>189</b>
	Table 47: Management Tools by Task . . . . .	189
	Table 48: IDP Security Policy Rulebases . . . . .	195
	Table 49: Recommended Security Policy Settings . . . . .	197
	Table 50: IDP Security Policy Templates . . . . .	198
	Table 51: New Policy Wizard: Page One . . . . .	199
	Table 52: New Policy Wizard: Page Two . . . . .	199
	Table 53: New Policy Wizard: Preconfiguration Options . . . . .	199
<b>Chapter 23</b>	<b>Configuring Profiler . . . . .</b>	<b>203</b>
	Table 54: Profiler Settings: General Tab . . . . .	205
	Table 55: Profiler Settings: Tracked Hosts or Exclude List . . . . .	207
	Table 56: Profiler Alert Tab . . . . .	210
	Table 57: Profiler Settings . . . . .	211
<b>Chapter 24</b>	<b>Configuring the IDP Rulebase . . . . .</b>	<b>213</b>
	Table 58: IDP Rulebase Rule Properties . . . . .	214

	Table 59: IDP Rulebase Match Condition Settings . . . . .	215
	Table 60: Attack Object Group Hierarchy . . . . .	217
	Table 61: IDP Rulebase Actions . . . . .	218
	Table 62: IDP Rulebase Actions: Recommended Actions by Severity . . . . .	219
	Table 63: IDP Rulebase IP Actions . . . . .	221
	Table 64: IDP Rulebase Notification Options . . . . .	222
	Table 65: IDP Rulebase VLAN Tag Settings . . . . .	223
	Table 66: IDP Rulebase Severity . . . . .	224
<b>Chapter 25</b>	<b>Configuring Additional Security Policy Rulebases . . . . .</b>	<b>227</b>
	Table 67: Exempt Rulebase Rule Properties . . . . .	228
	Table 68: APE Rulebase Rule Properties . . . . .	229
	Table 69: Backdoor Rulebase Rule Settings . . . . .	234
	Table 70: SYN Protector Rulebase Rule Properties . . . . .	236
	Table 71: Traffic Anomalies Rulebase Rule Properties . . . . .	238
	Table 72: Traffic Anomalies Rulebase Detection Settings . . . . .	239
	Table 73: Network Honeypot Rulebase Rule Properties . . . . .	240
<b>Chapter 26</b>	<b>Working with Attack Objects . . . . .</b>	<b>243</b>
	Table 74: Predefined Attack Object Groups . . . . .	244
	Table 75: Dynamic Attack Group Filters . . . . .	252
	Table 76: Custom Attack Dialog Box: General Tab Settings . . . . .	254
	Table 77: Custom Attack Dialog Box: Extended Tab Settings . . . . .	256
	Table 78: Attack Object Types . . . . .	258
	Table 79: Custom Attack – General Properties . . . . .	259
	Table 80: Custom Attack – Attack Pattern . . . . .	263
	Table 81: Custom Attack – IP Settings and Header Matches Page . . . . .	268
	Table 82: Custom Attack Object: TCP Packet Header Fields . . . . .	269
	Table 83: Custom Attack Object: UDP Header Fields . . . . .	271
	Table 84: Custom Attack Object: ICMP Packet Header Fields . . . . .	272
	Table 85: Custom Attack – General Properties . . . . .	274
	Table 86: Compound Attack Parameters . . . . .	275
<b>Chapter 27</b>	<b>Working with Application Objects . . . . .</b>	<b>279</b>
	Table 87: NSM Object Manager: Custom Application Objects . . . . .	294
<b>Chapter 28</b>	<b>Configuring Logging Features . . . . .</b>	<b>295</b>
	Table 88: IDP Series Device Configuration: Log Storage Limit Settings . . . . .	298
	Table 89: IDP Series Device Configuration: Log Suppression Settings . . . . .	299
	Table 90: IDP Series Device Configuration: SNMP Settings . . . . .	300
<b>Part 4</b>	<b>Administration</b>	
<b>Chapter 31</b>	<b>Logging . . . . .</b>	<b>331</b>
	Table 91: IDP Local Log Directories . . . . .	332
<b>Chapter 32</b>	<b>Managing Security Policies . . . . .</b>	<b>335</b>
	Table 92: IDP Detector Engine and NSM Attack Database Update Procedures . . . . .	337
	Table 93: Devices Update Job Options . . . . .	340
	Table 94: Device Update Job Options . . . . .	341

<b>Chapter 33</b>	<b>Managing the IDP Device Configuration with NSM</b> . . . . .	<b>343</b>
	Table 95: Pulling and Pushing Configuration Updates . . . . .	350
	Table 96: Device Update Job Options . . . . .	351
	Table 97: IDP Series Device Configuration: Info Settings . . . . .	352
	Table 98: IDP Series Device Configuration: Antispoof Settings . . . . .	354
	Table 99: IDP Series Device Configuration: Runtime Parameters . . . . .	356
	Table 100: IDP Series Device Configuration: Load Time Parameters . . . . .	366
	Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings . . . . .	369
<b>Chapter 34</b>	<b>Managing IDP Processes</b> . . . . .	<b>383</b>
	Table 102: Operations Requiring Use of Particular IDP Series User Interfaces . .	383
	Table 103: IDP Series Appliance Reboot and Shutdown Commands . . . . .	384
	Table 104: Command Reference: idp.sh . . . . .	385
<b>Chapter 35</b>	<b>Updating IDP Software</b> . . . . .	<b>389</b>
	Table 105: IDP Detector Engine and NSM Attack Database Update Procedures . . . . .	393
<b>Part 5</b>	<b>Monitoring</b>	
<b>Chapter 39</b>	<b>Overview</b> . . . . .	<b>411</b>
	Table 106: Supported Tools for Monitoring Tasks . . . . .	411
	Table 107: IDP Local Log Directories . . . . .	413
<b>Chapter 40</b>	<b>Using SNMP</b> . . . . .	<b>417</b>
	Table 108: System Resource Instrumentation Categories . . . . .	417
	Table 109: System Resource Instrumentation: Network Interface Traps . . . . .	425
	Table 110: System Resource Instrumentation: Resource SNMP Traps . . . . .	431
	Table 111: System Resource Instrumentation: Sensor SNMP Traps . . . . .	437
<b>Chapter 41</b>	<b>Using NSM Logs and Reports</b> . . . . .	<b>447</b>
	Table 112: NSM Device Monitor Status Data . . . . .	448
	Table 113: NSM Device Monitor: Device Details Page . . . . .	450
	Table 114: Log Viewing Options . . . . .	453
	Table 115: NSM Log Viewer: Log Columns . . . . .	454
	Table 116: NSM Log Viewer: Predefined Views . . . . .	459
	Table 117: NSM Audit Log Viewer Table . . . . .	460
	Table 118: NSM Audit Log Viewer: Target View Table . . . . .	460
	Table 119: NSM Audit Log Viewer: Device View Table . . . . .	461
	Table 120: Application Profiler Session Table . . . . .	464
	Table 121: Protocol Profiler Data . . . . .	465
	Table 122: Network Profiler Data . . . . .	467
	Table 123: NSM DI/IDP Predefined Reports . . . . .	470
	Table 124: NSM Profiler Predefined Reports . . . . .	471
	Table 125: NSM: Application Volume Tracking Reports . . . . .	471
	Table 126: Custom Report Configuration Options . . . . .	473
<b>Chapter 43</b>	<b>Using the bypassStatus Utility to Monitor the Internal Bypass Daemon</b> . . . . .	<b>483</b>
	Table 127: Command Reference: bypassStatus . . . . .	484

<b>Chapter 44</b>	<b>Using the sctop Utility to Monitor Session Flow . . . . .</b>	<b>487</b>
	Table 128: Command Key Reference: sctop Utility . . . . .	488
	Table 129: sctop Flow Table Report . . . . .	490
	Table 130: sctop Flow Table: Flag Column . . . . .	490
<b>Chapter 45</b>	<b>Using the scio Utility to Verify Feature Implementation . . . . .</b>	<b>491</b>
	Table 131: APE-Related scio Commands . . . . .	492
	Table 132: scio counters Related to Flow Bypass . . . . .	497
<b>Chapter 46</b>	<b>scio Commands . . . . .</b>	<b>499</b>
	Table 133: Command Reference: scio agentconfig . . . . .	500
	Table 134: Command Reference: scio app cache . . . . .	502
	Table 135: Command Reference: scio const . . . . .	505
	Table 136: scio const Arguments Related to the Application Identification Feature . . . . .	509
	Table 137: scio const Arguments Related to the APE Rulebase . . . . .	511
	Table 138: scio const Arguments Related to the Application Volume Tracking Feature . . . . .	512
	Table 139: scio const Arguments Related to Flow Bypass . . . . .	513
	Table 140: scio const Arguments Related to Policy Load . . . . .	514
	Table 141: scio const Arguments Related to GRE Decapsulation . . . . .	515
	Table 142: scio const Arguments Related to GTP Decapsulation . . . . .	515
	Table 143: scio const Arguments Related to IPsec ESP NULL Decapsulation . . . . .	517
	Table 144: scio const Arguments Related to MPLS Decapsulation . . . . .	517
	Table 145: scio const Arguments Related to SSL Inspection . . . . .	518
	Table 146: scio const Arguments Related to Maximum Frame Size . . . . .	519
	Table 147: scio const Arguments Related to the SYN Protector Rulebase . . . . .	519
	Table 148: scio const Arguments Related to the User Role-Based Policy Feature . . . . .	521
	Table 149: Command Reference: scio var . . . . .	522
	Table 150: Command Reference: scio idp-cpu-utilization . . . . .	525
	Table 151: Command Reference: scio nic . . . . .	528
	Table 152: Command Reference: scio sri . . . . .	529
	Table 153: Command Reference: scio ssl . . . . .	531
	Table 154: Command Reference: scio subs . . . . .	535
	Table 155: Command Reference: scio sysconf . . . . .	539
	Table 156: Command Reference: scio ca . . . . .	544
	Table 157: Command Reference: scio var . . . . .	547
	Table 158: Command Reference: scio vc . . . . .	549
	Table 159: Command Reference: scio vr . . . . .	551
<b>Chapter 47</b>	<b>IDP MIB Object ID Reference . . . . .</b>	<b>553</b>
	Table 160: snmpwalk Results by Name and by Number . . . . .	553
	Table 161: IDP Series MIB Objects . . . . .	559
	Table 162: IDP Series Traps . . . . .	567
<b>Part 6</b>	<b>Troubleshooting</b>	
<b>Chapter 48</b>	<b>Troubleshooting References . . . . .</b>	<b>575</b>
	Table 163: IDP Series Troubleshooting Tools . . . . .	575

**Chapter 50**

Table 164: Troubleshooting: IDP Processes Reference .....	577
<b>Troubleshooting Feature Implementation .....</b>	<b>581</b>
Table 165: Auto-Recovery Logs .....	583
Table 166: Diagnosing Problems with SNMP Reporting .....	586
Table 167: CPU Monitoring Tools .....	587
Table 168: Troubleshooting: Configuration Push Errors .....	593
Table 169: Troubleshooting: Security Policy Validation Errors .....	594



# About the Documentation

- Documentation and Release Notes on page xxv
- Supported Platforms on page xxv
- Documentation Conventions on page xxv
- Documentation Feedback on page xxvii
- Requesting Technical Support on page xxvii

## Documentation and Release Notes

---

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

## Supported Platforms

---

For the features described in this document, the following platforms are supported:

- IDP Series

## Documentation Conventions

---

Table 1 on page xxvi defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xxvi defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  <code>user@host&gt; configure</code>
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host&gt; show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies book names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS System Basics Configuration Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name:  <code>[edit]</code> <code>root@# set system domain-name <i>domain-name</i></code>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> <li>To configure a stub area, include the <b>stub</b> statement at the <code>[edit protocols ospf area area-id]</code> hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>
< > (angle brackets)	Enclose optional keywords or variables.	<code>stub &lt;default-metric <i>metric</i>&gt;;</code>

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	<b>broadcast   multicast</b>  <i>(string1   string2   string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	<b>community name members [ community-ids ]</b>
Indentation and braces ( { } )	Identify a level in the configuration hierarchy.	<b>[edit]</b> routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
<b>J-Web GUI Conventions</b>		
<b>Bold text like this</b>	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net), or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

## Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract,

or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf> .
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/> .
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

## Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

## Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/> .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html> .

## PART 1

# Overview

- [Solution Overview on page 3](#)
- [Profiler and Monitoring Features Overview on page 21](#)
- [Simulation Mode on page 33](#)
- [Security Policy Basics on page 37](#)
- [The IDP Rulebase on page 55](#)
- [The Exempt Rulebase on page 67](#)
- [The APE Rulebase on page 69](#)
- [The Backdoor Rulebase on page 85](#)
- [The SYN Protector Rulebase on page 91](#)
- [The Traffic Anomalies Rulebase on page 97](#)
- [The Network Honeypot Rulebase on page 103](#)
- [Additional Security Features on page 109](#)
- [Inspection of Encapsulated and Encrypted Traffic on page 111](#)



## CHAPTER 1

# Solution Overview

- [Juniper Networks IDP Solutions on page 3](#)
- [IDP Series Features Overview on page 3](#)
- [IDP Series Operating System Overview on page 7](#)
- [IDP Series Network Interfaces Overview on page 11](#)
- [Centralized Management with NSM Overview on page 18](#)
- [J-Security Center Updates Overview on page 19](#)

## Juniper Networks IDP Solutions

---

Juniper Networks provides intrusion detection services and intrusion detection and prevention (IDP) technology in the following product families:

- Juniper Networks IDP Series Intrusion Detection and Prevention Appliances
- Juniper Networks ISG Series Integrated Security Gateways
- Juniper Networks SRX Series Services Gateways

This guide describes the IDP Series appliances.

**Related Documentation** The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Features Overview on page 3](#)

## IDP Series Features Overview

---

[Table 3 on page 3](#) briefly describes Juniper Networks IDP Series features.

**Table 3: IDP Series Features**

Feature	Description	Documentation
Application-Based Management		

---

Table 3: IDP Series Features (*continued*)

Feature	Description	Documentation
Application identification	<p>Port-independent application identification enhances both security and manageability by eliminating the need to manually and comprehensively configure application-port mapping for the service objects and application objects used in the IDP rulebase and APE rulebase rules.</p> <p>Beginning with IDP OS Release 5.1, the application identification feature can match extended application signatures used in APE rulebase rules. <i>Extended application</i> signatures are also called <i>nested application</i> signatures. The predefined extended application signatures developed for IDP OS Release 5.1 include the most prevalent Web 2.0 applications running over HTTP.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Using Application Identification on page 43</a></li> </ul>
User-defined application signatures	If the predefined signatures do not address all of your use cases, you can use the NSM Object Manager to create custom application signatures.	<ul style="list-style-type: none"> <li>• <a href="#">Using Application Objects on page 73</a></li> </ul>
Application policy enforcement	The application policy enforcement (APE) rulebase enables you to mark, limit, or drop traffic that matches application signatures.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the APE Rulebase on page 69</a></li> </ul>
Application volume tracking	The application volume tracking (AVT) feature leverages Profiler functionality to collect statistics about application usage.	<ul style="list-style-type: none"> <li>• <a href="#">Application Volume Tracking Overview on page 22</a></li> </ul>
<b>Intrusion Detection and Prevention</b>		
Multimethod attack detection	The IDP Series uses eight methods to detect malicious traffic.	See <a href="#">Table 4 on page 6</a> for a description of detection methods.
Zero-day protection	The IDP rulebase attack objects detect protocol usages that violate published RFCs. Protocol anomaly detection protects your network from undiscovered vulnerabilities.	<ul style="list-style-type: none"> <li>• <a href="#">J-Security Center Updates Overview on page 19</a></li> </ul>
Protocol decoding	Juniper Networks Security Center (J-Security Center) provides a robust protocol detection engine that can decode more than 60 protocols and analyze and enforce proper usage in more than 500 contexts.	<ul style="list-style-type: none"> <li>• <a href="#">J-Security Center Updates Overview on page 19</a></li> </ul>
Recommended security policy and predefined attack objects	<p>J-Security Center provides a robust default security policy (called Recommended) and a comprehensive set of predefined attack objects (including those flagged as Recommended for various categories of attacks).</p> <p>The J-Security Center attack database includes more than 5500 signatures for identifying anomalies, attacks, spyware, and applications.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Using the Recommended Security Policy on page 46</a></li> <li>• <a href="#">Using Attack Objects on page 60</a></li> </ul>



Table 3: IDP Series Features (*continued*)

Feature	Description	Documentation
User-defined security policies and attack objects	<p>If you choose, you can use the default security policy or other predefined templates as a basis for your own user-defined security policy.</p> <p>Similarly, you can use the predefined attack objects as a basis for your own user-defined attack objects.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the Components of an IDP Security Policy on page 41</a></li> <li>• <a href="#">Using Attack Objects on page 60</a></li> </ul>
Active response methods	<p>J-Security Center attack objects are coded with recommended actions to take on the instant session, including drop packet, drop connection, close client, close server, and close client/server. You can rely on these or set your own.</p> <p>In addition, when the IDP Series device detects an attack from a particular IP address, it can block connections from the IP address for a configurable duration of time.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding IDP Rulebase Actions on page 63</a></li> </ul>
Passive response methods	The IDP Series supports several passive responses, including logging and TCP reset.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding IDP Rulebase Notification Options on page 65</a></li> </ul>
Traffic decryption and decapsulation	The IDP Series can decrypt or decapsulate traffic and then inspect the payload. We support decryption of SSL and decapsulation of GRE, GTP, IPsec ESP NULL, and MPLS traffic.	<ul style="list-style-type: none"> <li>• <a href="#">Inspection of SSL Traffic Overview on page 113</a></li> <li>• <a href="#">Inspection of GRE Traffic Overview on page 111</a></li> <li>• <a href="#">Inspection of GTP Traffic Overview on page 111</a></li> <li>• <a href="#">Inspection of IPsec VPN Traffic Overview on page 112</a></li> <li>• <a href="#">Inspection of MPLS Traffic Overview on page 112</a></li> </ul>
<b>Centralized Management and Logging</b>		
Centralized management	The IDP Series is compatible with Juniper Networks Network and Security Manager (NSM).	<ul style="list-style-type: none"> <li>• <a href="#">Centralized Management with NSM Overview on page 18</a></li> </ul>
Network profiling	The Profiler captures accurate and granular detail of your network traffic.	<ul style="list-style-type: none"> <li>• <a href="#">Profiler Overview on page 21</a></li> </ul>
Robust logging, reporting, and notification	The IDP Series includes useful predefined log views and reports and enables you to create custom views and reports.	<ul style="list-style-type: none"> <li>• <a href="#">IDP Logs Overview on page 24</a></li> <li>• <a href="#">NSM Reports Overview on page 29</a></li> <li>• <a href="#">IDP Reporter Overview on page 31</a></li> </ul>
<b>Simulation Mode, Autorecovery, Bypass, and Failover</b>		
Simulation mode	In simulation mode, the IDP Series inspects traffic according to your security policy but only simulates policy actions, generating logs of the action dictated by the security policy. You can use simulation mode when you first adopt the IDP Series solution to learn expected behavior without risk of traffic disruption.	<ul style="list-style-type: none"> <li>• <a href="#">Simulation Mode Overview on page 33</a></li> </ul>

Table 3: IDP Series Features (*continued*)

Feature	Description	Documentation
Autorecovery	If an IDP process engine experiences failure, the IDP Series device buffers the next packets in the flow and restarts the process engine.	<ul style="list-style-type: none"> <li>• <a href="#">Auto-Recovery Feature on page 9</a></li> </ul>
Bypass	<p>You can configure network interfaces to enter a bypass state in case of failure or graceful shutdown, or if the JNET driver encounters problems processing packets.</p> <p>The IDP Series also supports flow bypass to forward traffic when traffic exceeds IDP session capacity.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Internal Bypass on page 14</a></li> <li>• <a href="#">External Bypass on page 15</a></li> <li>• <a href="#">Flow Bypass Feature on page 10</a></li> </ul>
High availability	Feature set that operates well with third-party high availability solutions where you have deployed redundant network paths and use the failure detection features of a firewall, router, or switch to manage the cutover from the primary path to the backup path in cases of failure.	<ul style="list-style-type: none"> <li>• <a href="#">IDP Series Deployment Scenarios</a></li> </ul>
<b>Compatibility with Juniper Networks Access Solutions</b>		
User role-based policies	When integrated with Juniper Networks IC Series Unified Access Control (UAC) appliance, the IDP Series appliance supports security policy rules based on UAC user roles. This feature enables you to more easily configure focused rules to implement your business security policy.	<ul style="list-style-type: none"> <li>• <a href="#">IDP Series Deployment Scenarios</a></li> </ul>
Coordinated threat control	The IDP Series is compatible with Juniper Networks SA Series SSL VPN appliances and IC Series devices. The SA Series and IC Series devices can “subscribe” to IDP Series logs and use the logs as a basis for access rules.	<ul style="list-style-type: none"> <li>• <a href="#">IDP Series Deployment Scenarios</a></li> </ul>

[Table 4 on page 6](#) briefly describes IDP detection methods and provides a reference to detailed information.

Table 4: Intrusion Detection Methods

Feature	Description	Documentation
Stateful signature	The IDP rulebase attack object signatures are bound to protocol context. As a result, this detection method produces few false positives.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the IDP Rulebase on page 55</a></li> <li>• <a href="#">Using Attack Objects on page 60</a></li> </ul>
Protocol anomaly	The IDP rulebase attack objects detect protocol usages that violate published RFCs. This method protects your network from undiscovered vulnerabilities.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the IDP Rulebase on page 55</a></li> <li>• <a href="#">Using Attack Objects on page 60</a></li> </ul>

Table 4: Intrusion Detection Methods (*continued*)

Feature	Description	Documentation
Traffic anomaly	The Traffic Anomalies rulebase uses heuristic rules to detect unexpected traffic patterns that might indicate reconnaissance or attacks. This method blocks distributed denial-of-service (DDoS) attacks and prevents reconnaissance activities.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the Traffic Anomalies Rulebase on page 97</a></li> </ul>
Backdoor	The Backdoor rulebase uses heuristic-based anomalous traffic patterns and packet analysis to detect Trojans and rootkits. These methods prevent proliferation of malware in case other security measures have been compromised.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the Backdoor Rulebase on page 85</a></li> </ul>
IP spoofing	The IDP Series device checks the validity of allowed addresses inside and outside the network, permitting only authentic traffic and blocking traffic with a disguised source.	<ul style="list-style-type: none"> <li>• <a href="#">IP Spoof Attack Prevention Overview on page 109</a></li> </ul>
Denial of service (DoS)	The SYN Protector rulebase provides two, alternative methods to prevent SYN-flood attacks.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the SYN Protector Rulebase on page 91</a></li> </ul>
Network honeypot	The IDP Series device impersonates vulnerable ports so you can track attacker reconnaissance activity.	<ul style="list-style-type: none"> <li>• <a href="#">Understanding the Network Honeypot Rulebase on page 103</a></li> </ul>

**Related Documentation** The following additional related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Juniper Networks IDP Solutions on page 3](#)

## IDP Series Operating System Overview

The following topics explain the features of the IDP Series operating system:

- [IDP Series Multicore Architecture on page 7](#)
- [Auto-Recovery Feature on page 9](#)
- [Flow Bypass Feature on page 10](#)
- [Key Processes on page 10](#)

### IDP Series Multicore Architecture

The IDP Series operating system separates control plane and data plane processes, making the IDP Series devices resilient to periods of heavy load, such as a denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack). In high-end platforms, control plane and data plane processes run on separate CPU, bolstering resiliency and performance. [Figure 1 on page 8](#) shows the processes that run on each core CPU for IDP Series platforms.

Figure 1: IDP Multicore Architecture

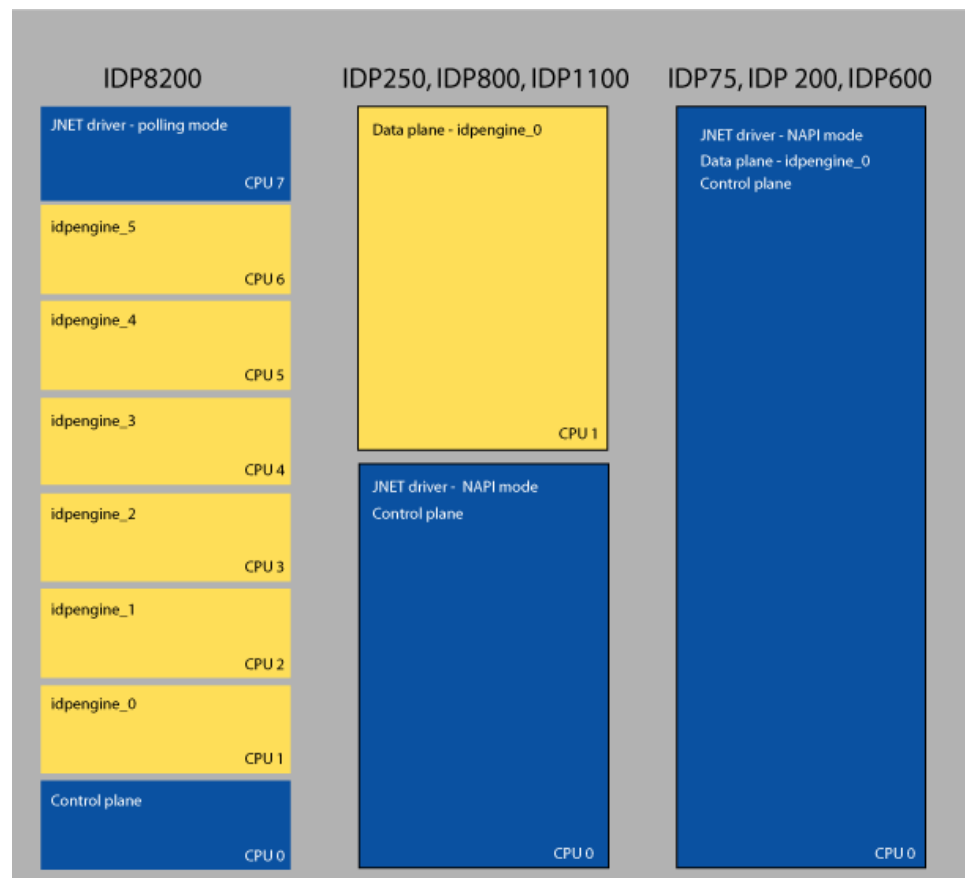


Table 5 on page 8 describes how CPUs are dedicated on IDP Series platforms.

Table 5: IDP Multicore Architecture

Platform	Multicore Architecture
IDP8200	<p>The IDP8200 appliance has eight core CPUs:</p> <ul style="list-style-type: none"> <li>One CPU is dedicated to control plane processes, including configuration and logging processes.</li> <li>One CPU is dedicated to the JNET driver processes, which handle traffic transmission and buffering. The JNET driver runs in polling mode.</li> <li>The remaining CPUs are dedicated to data plane processes, including the IDP engine processes that inspect traffic and take action against attacks.</li> </ul> <p><b>NOTE:</b> When you use the <b>scio</b> or <b>sctop</b> utilities to monitor IDP processes, you can specify the <b>-c CPU</b> option to display statistics related to processing on a particular IDP engine. For example, to display CPU utilization for CPU 2, the syntax is <b>scio -c 2 idp-cpu-utilization</b>. Without the <b>-c</b> option, <b>scio</b> displays an aggregate for all IDP engines.</p>

Table 5: IDP Multicore Architecture (*continued*)

Platform	Multicore Architecture
IDP1100, IDP800, IDP250	<p>The IDP1100, IDP800, and IDP250 appliances have two core CPUs:</p> <ul style="list-style-type: none"> <li>One CPU is used for both control plane and JNET driver processes. The JNET driver runs in NAPI mode: when traffic is low, the JNET driver operates in interrupt mode; when traffic is high, the JNET driver switches to polling mode.</li> <li>The second CPU is dedicated to data plane processes.</li> </ul>
IDP600, IDP200, IDP75	<p>The IDP600, IDP200, and IDP75 appliances have one core CPU. The JNET driver runs in NAPI mode: when traffic is low, the JNET driver operates in interrupt mode; when traffic is high, the JNET driver switches to polling mode.</p>



**NOTE:** If you use port monitoring utilities such as the Linux `lsof` command to view port activity, you will notice activity by host 127.0.0.1 (example below). These entries identify internal system communication. Communication on port 9101 and above and activity identifying Bacula processes is related to the internal communication between the control plane and data plane. This activity is essential to proper functioning of the IDP OS.

```
[root@idp-75-172-22-151-70 ~]# lsof -n -i
COMMAND    PID  USER  FD  TYPE DEVICE SIZE NODE NAME
idpengine  3164 root   4u  IPv4 39333      TCP *:bacula-dir (LISTEN)
idpengine  3164 root   5u  IPv4 40576      TCP
127.0.0.1:bacula-dir->127.0.0.1:39472 (ESTABLISHED)
idpengine  3164 root   6u  IPv4 39336      TCP 127.0.0.1:50000
(LISTEN)
idpengine  3164 root   7u  IPv4 302480     TCP
127.0.0.1:50000->127.0.0.1:37684 (ESTABLISHED)
idpengine  3164 root   9u  IPv4 302372     TCP
127.0.0.1:bacula-dir->127.0.0.1:59114 (ESTABLISHED)
..
```

## Auto-Recovery Feature

The auto-recovery feature monitors the status of individual IDP engines. In the event an IDP engine is terminated, the auto-recovery daemon logs the event, attempts to restart the IDP engine, and logs recovery. The auto-recovery feature is enabled by default.

The auto-recovery feature bolsters resiliency of IDP Series devices by enabling restart per IDP engine. In case of failure by an IDP engine, the other IDP engines in the data plane do not need to be restarted. If an IDP engine becomes overloaded or terminates, the JNET driver buffers packets for the active sessions while the IDP engine restarts. If the IDP engine does not restart after six attempts, the IDP Series device may enter internal bypass (if enabled).

The auto-recovery feature is complemented by the bypass under congestion feature. When the IDP engine recovers, it processes the buffered packets. If the buffer becomes large enough, it triggers bypass under congestion instead of resulting in subsequent failure.



**NOTE:** The auto-recovery process ensures the IDP engine restarts with the same device configuration, feature configuration, and security policy that were in place before the restart. However, because the application identification feature uses the first packets of a session as a key to determining the application, the auto-recovery process cannot reliably identify the application for buffered sessions. As a result, in processing buffered traffic, the application identification feature is unavailable and application rate limiting cannot be applied. In addition, the latest interval of application volume tracking data is discarded.

## Flow Bypass Feature

The flow bypass feature prevents the IDP Series device from becoming a point of failure when the network is congested. With flow bypass enabled, when the IDP system packet receive queue reaches a rising threshold that you specify, the IDP engine marks the flow as a bypass flow and passes it through, uninspected. The IDP Series device passes through subsequent flows until the IDP system receive queue falls below a reset threshold that you also specify. On IDP8200, which has multiple IDP engines, the flow bypass feature operates per IDP engine; that is, when the IDP packet receive queue for an individual IDP engine reaches its rising threshold, the individual IDP engine takes the flow bypass action and other IDP engines continue to inspect flows.

The flow bypass feature is disabled by default. It is intended for use in networks that prioritize availability over security.

For procedures for enabling the flow bypass feature, configuring its thresholds, and monitoring the feature, see the *IDP Series Administration Guide*.

## Key Processes

Table 6 on page 10 describes the key processes.

**Table 6: Processes**

Process	Description
agent	Manages communication between the IDP Series device and Network and Security Manager.
idpengine	Performs packet processing, including decapsulation or decryption, defragmentation, reassembly, inspection, and rule actions.
idpHMD	Generates SNMP alerts when thresholds are crossed for tracked resources on the device. Responds to SNMP poll requests. Resources are CPU, memory, hard disk space, and session count.
idpLogReader	Gathers logs generated by idpengine and stores the information in a local log database. The agent process forwards the data to NSM, where you can view records.
pkid	Inspects SSL traffic, if SSL inspection is turned on.

Table 6: Processes (*continued*)

Process	Description
profiler	Gathers information about hosts and applications in your network. Profiler stores the information in the Profiler database. The agent process forwards the data to NSM, where you can view records.
sciod	Handles policy push, information retrieval, Profiler status, and so on.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Network Interfaces Overview on page 11](#)

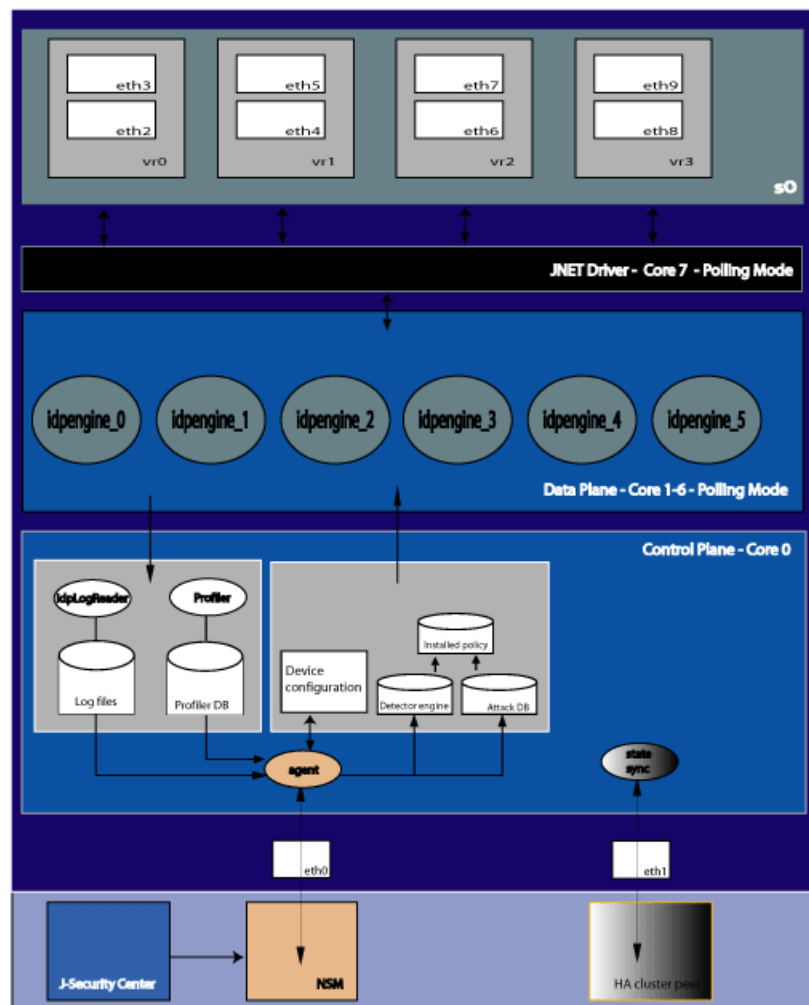
The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing Auto-Recovery Logs on page 583](#)
- [Enabling the Flow Bypass Feature on page 319](#)

## [IDP Series Network Interfaces Overview](#)

In [Figure 2 on page 12](#), eth0, eth1, eth2, eth3, and so forth are the network interfaces.

Figure 2: IDP Series Network Interfaces



The following topics explain the features of these network interfaces:

- [Management Interface \(eth0\)](#) on page 13
- [High Availability Interface \(eth1\)](#) on page 13
- [Traffic Interfaces](#) on page 13
- [Internal Bypass](#) on page 14
- [External Bypass](#) on page 15
- [Interface Signaling](#) on page 16
- [Peer Port Modulation](#) on page 16



## Management Interface (eth0)

In [Figure 2 on page 12](#), eth0 is a dedicated management interface used for communication with Network and Security Manager (NSM). The agent process is a control plane process. It manages communication between the IDP Series device and NSM. The agent process handles the following functionality:

- Device configuration—You set part of the active configuration with the Appliance Configuration Manager (ACM), part with the CLI, and part with NSM. The agent process pushes changes you make from NSM to the IDP Series device.
- Security policy—You configure policies with NSM. You push a single policy to the IDP Series device to be installed and used by the IDP process engines. The installed policy is the policy used to determine which traffic the IDP engine inspects, what to look for, and what actions to take.
- Detector engine—The IDP detector engine is a code base that contains the application signatures and protocol decoder definitions used by the IDP engine in packet analysis. J-Security Center periodically updates the IDP detector engine. In [Figure 2 on page 12](#), note the process flow: first, you download updates from J-Security Center to NSM; then, you push updates from NSM to IDP Series devices.
- Attack database—The attack database includes the attack objects used by the IDP rulebase to match attack signatures and protocol anomalies. J-Security Center updates predefined attack object definitions as often as necessary. As with detector engine updates, you download them from J-Security Center to NSM and then push them from NSM to the IDP Series device.
- Logging—The IDP process engines generate logs and packet captures related to security policy and application policy enforcement rules. The Profiler generates profiling and application volume logs. The agent process sends these logs to NSM so you can use NSM monitoring features to monitor security events and application usage.

## High Availability Interface (eth1)

In [Figure 2 on page 12](#), eth1 is a dedicated high availability (HA) interface used for sync-state communication with a cluster peer in a high availability deployment.

## Traffic Interfaces

In [Figure 2 on page 12](#), eth2, eth3, and so forth are the network interfaces you connect to the network devices that route traffic in your network.

The IDP Series implements the following abstract objects to manage network interfaces:

- Virtual circuit—A virtual circuit corresponds with the physical interface. For example, physical interface eth2 is a virtual circuit. You use the Appliance Configuration Manager (ACM) to configure speed and duplex, as well as optional interface alias settings for each interface.
- Virtual router—A virtual router contains a logical pair of virtual circuits. For example, virtual router vr0 contains eth2 and eth3. In transparent mode, traffic arrives in one

interface and is forwarded through the other. You use ACM to configure the deployment mode (sniffer or transparent) and bypass options (internal, external, or off) for each virtual router. You can use the command-line interface to display information and status for each virtual router, including Address Resolution Protocol (ARP) and media access control (MAC) tables.

- **Subscriber**—A single subscriber named s0 contains all virtual routers. The subscriber maintains process and status of all traffic that flows through the device. You can use the command-line interface to view information and status maintained by subscriber s0. We test and support only configurations where the default subscriber is used.

## Internal Bypass

The Internal Bypass feature is intended for deployments where a network security policy privileges availability over security. In the event of failure or graceful shutdown, traffic bypasses the IDP processing engine and is passed through the IDP Series device uninspected.

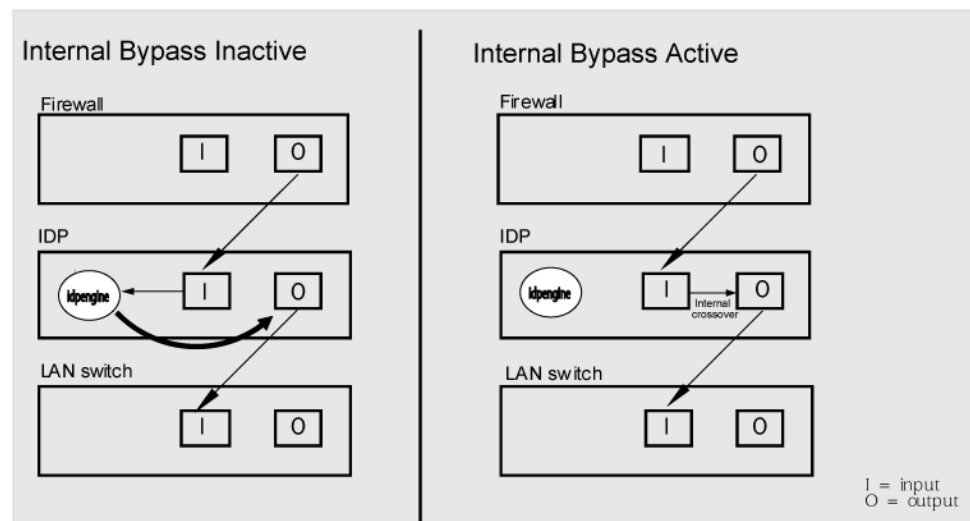
The Internal Bypass feature operates through a timing mechanism. When enabled, the timer on traffic interfaces counts down to a bypass trigger point. When the IDP Series appliance is turned on and available, it sends a reset signal to the traffic interface timer so that it does not reach the bypass trigger point. If the IDP OS encounters failure, then it fails to send the reset signal, the timer counts down to the trigger point, and the traffic interfaces enter a bypass state. If the IDP Series appliance is shut down gracefully, the traffic interfaces immediately enter bypass.

With copper NICs, the bypass mechanism joins the interfaces mechanically to form a circuit that bypasses IDP processing. Packets traverse the IDP Series device as if the path from eth2 (receiving interface) to eth3 (transmitting interface) were a crossover cable. No packet inspection or processing occurs.

With fiber NICs, the bypass mechanism uses optical relays instead of copper relays. During normal operations, the optical relays send light to the built-in optical transceivers. When bypass is triggered, the relays flip state, and the light signal is redirected to optically connect the two external ports.

[Figure 3 on page 15](#) compares the data path when Internal Bypass is enabled but not activated with the data path when Internal Bypass is activated.

Figure 3: Internal Bypass

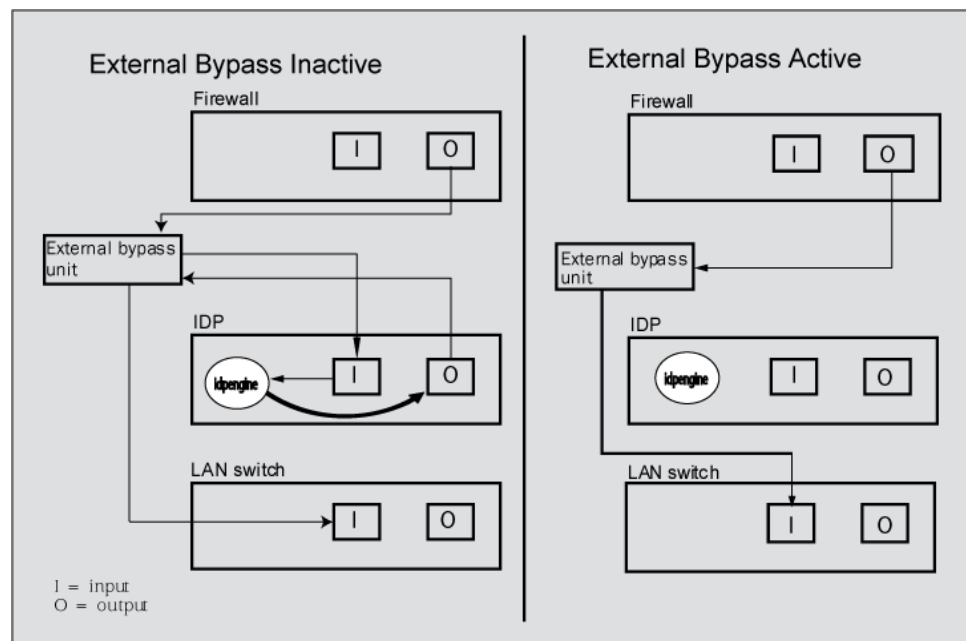


When the IDP operating system resumes healthy operations, it sends a reset signal to the traffic interfaces, and the interfaces resume normal operation.

## External Bypass

The External Bypass setting supports third-party external bypass units. Deployments with external bypass units depend on the functionality of the external bypass unit to check the status of the IDP Series appliance and make the determination whether to send packets through or around the IDP Series device. Most external bypass units test for availability by sending heartbeat packets through the device. If the packets reach the expected destination, the external bypass unit allows the traffic to continue through the IDP Series appliance. If the packets fail to reach the expected destination, the external bypass unit determines the IDP Series is unavailable, so it forwards traffic around the IDP Series device. The IDP Series supports external bypass solutions by allowing the heartbeat traffic to pass through the device regardless of the Layer 2 Bypass setting. In other words, if you disable Layer 2 Bypass and enable External Bypass, most Layer 2 traffic will be dropped but the heartbeat traffic used in the external bypass deployment will be passed through. [Figure 4 on page 16](#) compares the data path when External Bypass is enabled but not activated with the data path when External Bypass is activated.

Figure 4: External Bypass



## Interface Signaling

The interface signaling feature supports high-availability deployments where there are redundant network paths, and a firewall, router, or switch chooses the active path. The interface signaling script monitors the state of the following IDP Series components:

- Traffic interfaces (eth2, eth3, and so on). In case of interface failure, the script brings down all peer interfaces so that a third-party link detection mechanism can properly detect failure.
- IDP engines (idpengine0, idpengine1, and so on). In case of IDP engine failure, the auto-recovery feature attempts to restart the IDP engine. If the IDP engine cannot be restarted after six attempts, the auto-recovery process runs an **idp.sh stop** command. The interface signaling script then brings down all traffic interfaces.

After bringing down the peer interfaces, the interface signaling script sleeps for 30 seconds to avoid link flapping issues. After 30 seconds, the script checks the state of the IDP engine or interface that had encountered the failure. When the underlying problem has been resolved and the interface is up, the interface signaling script brings up the peer interfaces.

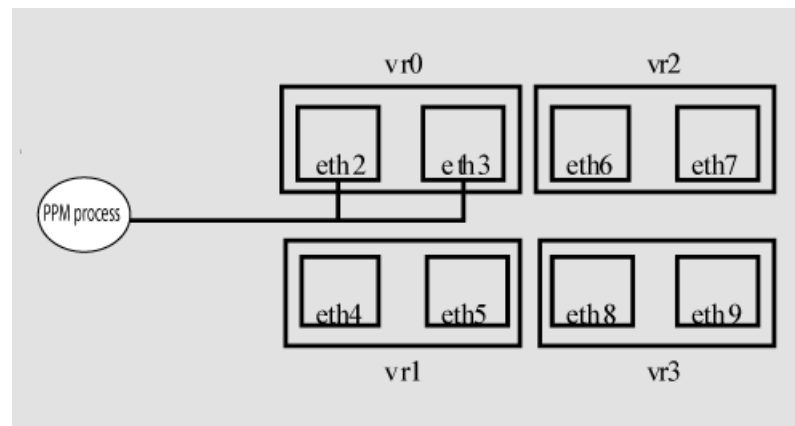
## Peer Port Modulation

The peer port modulation (PPM) feature supports deployments where routers monitor link state to make routing decisions. In these deployments, a router might be set to monitor link state on only one side of the IDP Series device. Suppose, for example, the router monitors only the IDP inbound interface. Suppose the inbound interface remains up but the outbound interface goes down. The router watching the inbound link would detect an available link and forward traffic to the IDP Series device. Traffic would be dropped

at the point of failure—the outbound link. PPM propagates a link loss state for one traffic interface to all interfaces in the IDP virtual router.

When PPM is enabled, a PPM daemon monitors the health of IDP traffic interfaces belonging to the same virtual router. If a traffic interface loses link, the PPM process turns off any associated network interfaces in the same virtual router so that other network devices detect that the virtual router is down and route around it. For example, assume you have enabled PPM and configured IDP virtual routers as shown in [Figure 5 on page 17](#).

**Figure 5: Peer Port Modulation**



Suppose there is a network problem and eth3 goes down. The PPM daemon detects this and turns off the other interface in vr0: eth2. The interfaces in vr1, vr2, and vr3 are unaffected. After you fix the problem with eth3, the PPM daemon detects this, and turns on eth2.



**NOTE:** The PPM feature is independent of the bypass feature (NIC state setting). PPM is related to the *status of the link*, not the status of the IDP operating system. A link can be down even when the IDP operating system is healthy. Note, however, that PPM runs as a control plane process and operates only when the IDP Series device is turned on and the control plane is available. If the IDP operating system is unavailable, the PPM feature is also unavailable, regardless of the setting for the NIC state.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)
- [Centralized Management with NSM Overview on page 18](#)

The following related topics are included in the *IDP Series Administration Guide*:

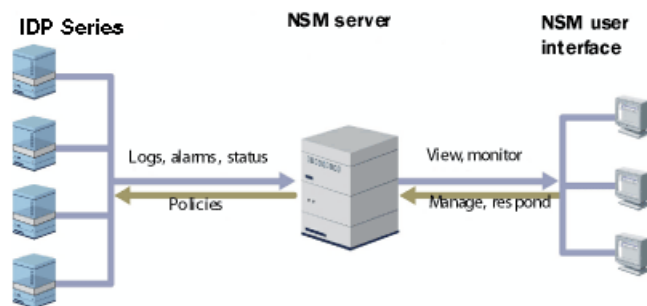
- [Configuring Virtual Routers \(ACM Procedure\) on page 192](#)
- [Tuning the JNET Driver Failure Count on page 581](#)
- [Configuring Interface Aliasing \(ACM Procedure\) on page 296](#)

## Centralized Management with NSM Overview

Juniper Networks Network and Security Manager (NSM) is a central management server capable of managing hundreds of IDP Series devices and other Juniper Networks devices, such as ScreenOS firewalls, SA Series devices, and IC Series devices. You typically deploy NSM in a management subnet accessible to the NSM-managed devices.

Figure 6 on page 18 illustrates the flow of information between the tiers of the central management solution: the NSM user interface, the NSM server, and IDP Series devices.

Figure 6: IDP-NSM Communication



The IDP Series configuration, security policies, attack objects, and log records are stored in NSM server databases and administered using the NSM user interface. Communication between the NSM server and IDP Series devices, and between the NSM server and the NSM user interface, is encrypted and authenticated.

For IDP Series deployments, centralized management provides the following benefits:

- Centralized management for IDP Series devices and other network devices
- Consolidated logs from different devices in a single repository
- Centralized management of enterprise security policies
- Simplified management for attack signature updates
- Role-based administration

For information about installing NSM and using NSM distributed management features, management objects (such as address objects, service objects, and templates), and navigational and display features, see the NSM documentation.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [J-Security Center Updates Overview on page 19](#)
- [Management Interface \(eth0\) on page 13](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [NSM Device Configuration Management Task Summary on page 343](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## J-Security Center Updates Overview

The Juniper Networks Security Center (J-Security Center) routinely makes important updates available to IDP security policy components:

- **Detector engine.** The IDP detector engine is a dynamic protocol decoder that includes support for decoding more than 60 protocols and more than 500 service contexts. You should update IDP detector engine when you first install IDP, whenever you upgrade, and whenever alerted to do so by Juniper Networks. You can view release notes for detector engine updates at <http://www.juniper.net/techpubs/software/management/idp/de/>.
- **Attack database.** The [attack signature database](#) stores data definitions for attack objects. Attack objects are patterns comprising stateful signatures and traffic anomalies. You specify attack objects in IDP rulebase rules.
- **Application signature database.** The [application signature database](#) stores data definitions for application objects. Application objects are patterns used to identify applications and match APE rulebase rules.

J-Security Center updates are packaged and released separately from the IDP operating system and software code base to ensure IDP products protect your network against recently discovered vulnerabilities. We recommend you schedule automatic updates for the attack database and application database. For IDP Series devices, both databases are distributed in “signature database updates”.

After you have completed the update, any new attack objects and application objects are available in the security policy editor. If you use dynamic groups in IDP rulebase rules and a new attack object belongs to the dynamic group, the rule automatically inherits the new attacks.



**NOTE:** We recommend you subscribe to the IDP Signature Updates technical bulletin to be notified when J-Security Center releases IDP detector engine updates. Go to <https://www.juniper.net/alerts/>.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Centralized Management with NSM Overview on page 18](#)
- [Using Attack Objects on page 60](#)
- [Using Application Objects on page 73](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)
- [Application Objects Task Summary on page 286](#)
- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)





## CHAPTER 2

# Profiler and Monitoring Features Overview

- [Profiler Overview on page 21](#)
- [Application Volume Tracking Overview on page 22](#)
- [IDP Logs Overview on page 24](#)
- [NSM Reports Overview on page 29](#)
- [IDP Reporter Overview on page 31](#)

## Profiler Overview

---

The Profiler is a network-analysis tool that helps you learn about your internal network so you can create effective security policies and minimize unnecessary log records. The Profiler queries and correlates information from multiple IDP Series devices.

After you configure the Profiler, it automatically learns about your internal network and the elements that constitute it, including hosts, peers (which host is talking to which other host), ports (non-IP protocols, TCP/UDP ports, RPC programs), and Layer 7 data that uniquely identifies hosts, applications, commands, users, and filenames. You can use this data to investigate and analyze potential problems in the network and to resolve security incidents.

During profiling, the IDP Series device records network activity at Layer 3, Layer 4, and Layer 7 and stores this information in a searchable database called the Profiler DB. The Profiler uses session creation, session teardown, and protocol contexts to generate this database, which defines all unique activities occurring on your network. Unique activities include attempts, probes, and successful connections.

The device logs normal events only once, and it logs all unique events as often as they occur.

A *normal event* is an event that reoccurs frequently and does not change. For example, suppose Wendy holds a meeting every Tuesday at 4:00 PM in conference room A. Every meeting, she connects her laptop to the network and accesses documents on the primary fileserver. Because the same event occurs multiple times, the device logs the event once and includes a timestamp that indicates the first and last times Wendy accessed the network from conference room A.

A *unique event* is an event that is new, unexpected, or does not match the normal traffic patterns of your network. For example, suppose that in her weekly meeting, Wendy

accesses documents from a different fileserver or has a colleague lead the meeting when she is on vacation. Because the network session information differs, the device logs these activities separately from the normal Tuesday afternoon meeting.

When you configure the Profiler, you can specify:

- General settings, such as whether to collect application volume data and whether to record the OS fingerprint of network hosts
- Network and host IP addresses to track in Profiler logs
- Network and host IP addresses to exclude in Profiler logs
- Contexts to retrieve additional data
- Alerts in cases where you want to track new hosts and applications

For complete procedures on setting Profiler options, see the *IDP Series Administration Guide*.

#### **Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Example: Using Profiler to Set a Baseline on page 124](#)
- [Example: Using Profiler to Alert You to New Hosts and Port Activity on page 129](#)
- [Example: Identifying Services That Use Nonstandard Ports on page 129](#)
- [Example: Responding to Vulnerability Announcements with Due Diligence on page 130](#)
- [Example: Using Profiler to Investigate Unanticipated Attacks on page 131](#)
- [Example: Using Profiler to Mitigate Risks from Laptops on page 132](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

---

## **Application Volume Tracking Overview**

The application volume tracking (AVT) feature uses application identification and the Profiler to collect application statistics aggregated at 15-minute and 1-hour intervals. The AVT database stores up to four sets of each interval at a time (four 15-minute intervals and four 1-hour intervals). After it has accumulated four intervals, it begins dropping the oldest interval as it collects a new one.

The AVT process writes data files to the following directories:

- `/usr/idp/device/var/stat/1hour`
- `/usr/idp/device/var/stat/15min`

The data is collected and parsed for reporting in NSM or IDP Reporter.

[Table 7 on page 23](#) describes the columns of data in AVT records for each session.

Table 7: Application Volume Tracking Data

Data Field	Description
Session ID	Unique ID for the session.
Source IP address	IP address for the host that initiated the session.
Source port	The port number for the source host.
Destination IP address	IP address for the destination server.
Destination port	The port number of the destination host.
VLAN ID	VLAN ID (if any).
Protocol	The IP protocol: TCP, UDP, or ICMP.
Application ID	The application identified by the application identification feature. Extended applications (also called nested applications) are reported separately from HTTP results. A 0 indicates the application was not identified.
Bytes	Throughput in bytes for sessions during the interval. AVT tracks both server-to-client and client-to-server bytes.
Packets	Number of packets for sessions during the interval. AVT tracks both server-to-client and client-to-server packets.

Table 8 on page 23 lists documentation references for AVT log viewing tools.

Table 8: Application Volume Tracking Log Viewing Tools

AVT Log Viewing Tools	Documentation
NSM Profiler Viewer > Application Profiler tab (logs) NSM Report (reports)	<i>IDP Series Administration Guide</i>
IDP Reporter	<i>IDP Reporter User's Guide</i>



**NOTE:** To avoid issues with reports, we highly recommend that you synchronize the network clocks for all devices to the same NTP server. For example, the network clocks for all IDP Series devices and NSM clients should be synchronized to the NTP server specified in the NSM configuration.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Example: Using NSM to Enable and View Application Volume Tracking on page 133](#)
- [Profiler Overview on page 21](#)

- [NSM Reports Overview on page 29](#)
- [IDP Reporter Overview on page 31](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)
- IDP Reporter Task Summary

## IDP Logs Overview

The IDP system generates logs for device events and security events.

Device event logs are related to the operation of the IDP Series device. IDP OS Release 5.1 supports extensive system resource instrumentation, so you can use SNMP utilities to monitor device health and load.

Security event logs are related to traffic that matches security policy rules or IP spoof attack settings.

[Table 9 on page 24](#) summarizes options for viewing and managing logs.

**Table 9: IDP Logging Options**

Option	Description
Network and Security Manager (NSM)	<p>IDP Series devices automatically send device event logs to NSM. IDP Series devices send security event logs when traffic matches security policy rules for which logging has been enabled. You can use the NSM log viewer to sort through and analyze logs. If you enable packet logging for a security policy rule, you can use the NSM packet viewer to display packet data.</p> <p>For details on using NSM log utilities, see the <i>Network and Security Manager Administration Guide</i>.</p>
Syslog	<p>You can configure IDP Series devices to forward logs to a syslog server, a commonly used method for storing logs for troubleshooting or record-keeping.</p> <p>For details on configuring syslog collection, see the <i>IDP Series Administration Guide</i>.</p>
SNMP	<p>SNMP reporting is enabled by default. You can use SNMP methods to track the following metrics:</p> <ul style="list-style-type: none"> <li>• CPU usage</li> <li>• Memory usage</li> <li>• Disk usage</li> <li>• Packet buffer usage</li> <li>• Session statistics</li> <li>• Network interface statistics</li> <li>• Traffic statistics</li> </ul> <p>For details on configuring SNMP reporting, see the <i>IDP Series Administration Guide</i>.</p>

Table 9: IDP Logging Options (*continued*)

Option	Description
Log suppression	<p>You can configure log suppression to reduce the volume of logs. The log suppression feature eliminates multiple log entries for events with the same properties. Instead, in NSM Log Viewer, a single entry appears along with a count of all such matching events.</p> <p>For details on configuring log suppression, see the <i>IDP Series Administration Guide</i>.</p>



**NOTE:** To avoid issues with reports, we highly recommend that you synchronize the network clocks for all devices to the same NTP server. For example, the network clocks for all IDP Series devices and NSM clients should be synchronized to the NTP server specified in the NSM configuration.

Table 10 on page 25 describes the fields that appear in log entries.

Table 10: NSM Log Viewer: Log Columns

Column	Description
Log ID	Unique ID for the log entry, derived by combining the date and log number.
Time Received	Date and time that the management system received the log entry.
Alert	NSM-defined alert for this type of log entry. Configure alerts in policy rules.
User Flag	<p>To set a flag, right-click the log row, select Flag, and then select one of the following flags:</p> <ul style="list-style-type: none"> <li>• High</li> <li>• Medium</li> <li>• Low</li> <li>• Closed</li> <li>• False Positive</li> <li>• Assigned</li> <li>• Investigate</li> <li>• Follow-up</li> <li>• Pending</li> </ul>
Src Addr	Source IP address of the packet that generated the log entry.
Dst Addr	Destination IP address of the packet that generated the log entry.

Table 10: NSM Log Viewer: Log Columns (*continued*)

Column	Description
Action	<p>Action the security device performed on the packet/connection that generated this log entry:</p> <ul style="list-style-type: none"> <li>Accepted—Did not block the packet.</li> <li>Closed Client—Closed the connection and sent an RST packet to the client, but did neither to the server.</li> <li>Closed Server—Closed the connection and sent an RST packet to the server, but did neither to the client.</li> <li>Closed—Closed the connection and sent an RST packet to both the client and the server.</li> <li>Dropped—Dropped the connection without sending an RST packet to the sender, preventing the traffic from reaching its destination.</li> <li>Dropped Packet—Dropped a matching packet before it could reach its destination but did not close the connection.</li> <li>Ignored—Matched the attack, did not take action, and ignored the remainder of the connection.</li> </ul> <p><b>NOTE:</b> Beginning in IDP OS Release 5.1, IDP logs show the action that was taken, rather than the action that was specified in the rule. For TCP events, these are the same. Close actions are not possible for UDP or ICMP packets. For UDP and ICMP events, the IDP logs show the action take—drop—instead of a close client, close server, or close client and server actions that might have been configured for the rule.</p>
Protocol	Protocol that the packet that generated the log entry used.
Dst Port	Destination port of the packet that generated the log entry.
Rule #	Security policy rule that generated the log entry.
Nat Src Addr	NAT source address of the packet that generated the log entry.
Nat Dst Addr	NAT destination address of the packet that generated the log entry.
Details	Miscellaneous string associated with log entry.
Category	<p>Type of log entry:</p> <ul style="list-style-type: none"> <li>Admin</li> <li>Alarm—The device generates event alarms for any security event that has a predefined severity level of emergency, critical, or alert. Additionally, the device generates traffic alarm log entries when it detects network traffic that exceeds the specified alarm threshold in a rule (the traffic alarm log entry describes the security event that triggered the alarm).</li> <li>Config—A configuration change occurred on the device.</li> <li>Custom—A match with a custom attack object was detected.</li> <li>Implicit—An implicit rule was matched.</li> <li>Info—General system information.</li> <li>Predefined—A match with a predefined attack object was detected.</li> <li>Profiler—Traffic matches a Profiler alert setting.</li> <li>Screen—Not applicable for IDP Series devices. Generated by ScreenOS firewall devices.</li> <li>Self—The device generated this log for a non-traffic related reason.</li> <li>Sensor.</li> <li>Traffic—Traffic matches a rule you have configured for harmless traffic.</li> <li>URL Filtering—Not applicable for IDP Series devices. Generated by ScreenOS firewall devices.</li> <li>User.</li> </ul>

Table 10: NSM Log Viewer: Log Columns (*continued*)

Column	Description
Subcategory	Category-specific type of log entry (examples are "Reboot" or message ID).
Severity	Severity rating associated (if any) with this type of log entry: <ul style="list-style-type: none"> <li>• Not Set (the device could not determine a severity for this log entry)</li> <li>• Info</li> <li>• Device_warning_log</li> <li>• Minor</li> <li>• Major</li> <li>• Device_critical_log</li> <li>• Emergency</li> <li>• Error</li> <li>• Notice</li> <li>• Informational</li> <li>• Debug</li> </ul>
Device	Device that generated this log entry.
Comment	User-defined comment about the log entry.
Application Name	Application associated with the current log.
Bytes In	For sessions, specifies the number of inbound bytes.
Bytes Out	For sessions, specifies the number of outbound bytes.
Bytes Total	For sessions, specifies the combined number of inbound and outbound bytes.
Dev Domain Ver	Domain version that generated this log entry.
Device Domain	Domain for the device that generated this log entry.
Device family	Family of the device that generated this log entry.
Dst Intf	Name of the outbound interface of the packet that generated this log entry.
Dst Zone	Destination zone associated with a traffic log entry.
Elapsed Secs	For sessions, specifies how long the session lasted.
Has Packet Data	Indicates whether the log entry has associated packet data.
NAT Dst Port	The NAT destination port of the packet that generated the log entry.
NAT Src Port	The NAT source port of the packet that generated the log entry.

Table 10: NSM Log Viewer: Log Columns (*continued*)

Column	Description
Packets In	For sessions, specifies the number of inbound packets.
Packets Out	For sessions, specifies the number of outbound packets.
Packets Total	For sessions, specifies the combined number of inbound and outbound packets.
Policy	Security policy that generated the log entry.
Roles	Role group associated with this log entry.
Rule Domain	The domain of the rule that generated the log entry.
Rule Domain Ver	The domain version of the rule that generated the log entry.
Rulebase	Security policy rulebase that generated the log entry.
Src Intf	Name of the inbound interface of the packet that generated this log entry.
Src Port	Source port of the packet that generated the log entry.
Src Zone	Source zone associated with a traffic log entry.
Time Generated	Date and time the device generated the log entry.
User	User associated with this log entry.

The following example shows a syslog message record:

```
[syslog@juniper.net dayId="20061012" recordId="0" timeRecv="2006/10/12 21:52:21"
timeGen="2006/10/12 21:52:21"
domain="" devDomVer2="0" device_ip="10.209.83.4" cat="Predefined"
attack="TROJAN:SUBSEVEN:SCAN"
srcZn="NULL" srcIntf="NULL" srcAddr="192.168.170.20" srcPort="63396"
natSrcAddr="NULL" natSrcPort="0"
dstZn="NULL" dstIntf="NULL" dstAddr="192.168.170.10" dstPort="27374"
natDstAddr="NULL" natDstPort="0" protocol="TCP"
ruleDomain="" ruleVer="5" policy="Policy2" rulebase="IDS" ruleNo="4" action="NONE"
severity="LOW" alert="no"
elapsedTime="0" inbytes="0" outbytes="0" totBytes="0" inPak="0" outPak="0"
totPak="0" repCount="0" packetData="no"
varEnum="31" misc="<017>'interface=eth2" user="NULL" app="NULL" uri="NULL"]
```

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Developing a Logging Strategy on page 331](#)
- [Developing a Log Storage Strategy on page 332](#)
- [Example: Using NSM Log Viewer Features on page 139](#)



- [Example: Packet Logging Workflow on page 145](#)
- [Understanding IDP Rulebase Notification Options on page 65](#)
- [Understanding Backdoor Rulebase Notification Options on page 89](#)
- [Understanding SYN Protector Rulebase Notification Options on page 95](#)
- [Understanding Traffic Anomalies Rulebase Notification Options on page 101](#)
- [Understanding Network Honeypot Rulebase Notification Options on page 106](#)
- [IP Spoof Attack Prevention Overview on page 109](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)
- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)

## NSM Reports Overview

IDP reports are representations of log data, aggregated and sorted to facilitate network and security analysis. The standalone IDP solution supports both centralized, aggregated reporting through NSM, and on-box reporting for a single IDP Series device through IDP Reporter.

NSM Report Manager contains a set of predefined network and security reports, including a group of deep inspection (DI) and intrusion detection and prevention (IDP) reports.

[Table 11 on page 29](#) summarizes NSM DI/IDP predefined reports.

**Table 11: NSM DI/IDP Predefined Reports**

Report	Description
Top 100 Attacks (last 24 hours)	Those attacks that are detected most frequently within the last 24 hours.
Top 100 Attacks Prevented (last 24 hours)	Those attacks that are prevented most frequently within the last 24 hours.
Top 20 Attackers (All Attacks - last 24 hours)	IP addresses that have most frequently been the source of an attack during the last 24 hours.
Top 20 Attackers Prevented (All Attacks - last 24 hours)	IP addresses that have most frequently been prevented from attacking the network during the last 24 hours.
Top 20 Targets (last 24 hours)	IP addresses that have most frequently been the target of an attack during the last 24 hours.
Top 20 Targets Prevented (last 24 hours)	IP addresses that have most frequently prevented attacks during the last 24 hours.
All Attacks by Severity (last 24 hours)	Number of attacks by severity level (set in attack objects) during the last 24 hours.
All Attacks Prevented by Severity (last 24 hours)	Number of attacks prevented by severity level (set in attack objects) during the last 24 hours.

Table 11: NSM DI/IDP Predefined Reports (*continued*)

Report	Description
All Attacks Over Time (last 7 days)	All attacks detected during the last 7 days.
All Attacks Prevented Over Time (last 7 days)	All attacks prevented during the last 7 days.
All Attacks Over Time (last 30 days)	All attacks detected during the last 30 days.
All Attacks Prevented Over Time (last 30 days)	All attacks prevented during the last 30 days.
Critical Attacks (last 24 hours)	All attacks categorized as "critical" detected during the past 24 hours.
Critical Attacks Prevented (last 24 hours)	All attacks categorized as "critical" prevented during the past 24 hours.
Critical through Medium Attacks (last 24 hours)	All attacks categorized as either "critical" or "medium" detected during the past 24 hours.
Critical through Medium Attacks Prevented (last 24 hours)	All attacks categorized as either "critical" or "medium" prevented during the past 24 hours.
Top 50 Scan Sources (last 7 days)	IP addresses that have most frequently been the source of a scan during the past 7 days.
Top 50 Scan Targets (last 7 days)	IP addresses that have most frequently been the target of a scan over the last 7 days.
Profiler - New Hosts (last 7 days)	New hosts listed in the Profiler over the last 7 days.
Profiler - New Ports (last 7 days)	New ports listed in the Profiler over the last 7 days.
Profiler - New Protocols (last 7 days)	New protocols listed in the Profiler over the last 7 days.
Top IDP Rules	The total number of log entries generated by specific rules in your IDP policies. You can use the Top Rules report to identify those rules that are generating the most log events. This enables you to better optimize your rulebases by identifying those rules that are most and least effective. You can then modify or remove those rules from your security policies.

[Table 12 on page 30](#) describes Profiler predefined reports. These reports are related to activity by hosts in your network.

Table 12: NSM Profiler Predefined Reports

Report	Description
Top 10 Peers by Count	Source and destination IP addresses that appeared most frequently in the Profiler logs.

Table 12: NSM Profiler Predefined Reports (*continued*)

Report	Description
Top 10 Peers with maximum hits	Source and destination IP addresses that had the highest number of hits in the Profiler logs.

[Table 13 on page 31](#) describes the predefined application volume tracking (AVT) reports. The reports are related to application use in your network.

Table 13: NSM: Application Volume Tracking Reports

Report	Description
Top 10 Applications by Volume	Applications with the highest volume in bytes in the past 24 hours.
Top 10 Application Categories by Volume	Application categories with the highest volume in bytes in the past 24 hours.
Top 5 Applications by Volume over Time (last 1 hour)	Applications with the highest volume in bytes in the past hour.
Top 5 Application Categories by Volume (last 1 hour)	Application categories with the highest volume in bytes in the past hour.
Top 5 Source by Volume over Time (last 1 hour)	Source IP addresses with the highest volume in bytes in the past hour.
Top 5 Destination by Volume over Time (last 1 hour)	Destination IP addresses with the highest volume in bytes in the past hour.

In addition to these predefined reports, you can create custom reports based on IDP log fields. For details on creating custom reports, see the *IDP Series Administration Guide*.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## IDP Reporter Overview

IDP reports are representations of log data, aggregated and sorted to facilitate network and security analysis. IDP Series devices support both centralized, aggregated reporting through NSM, and on-box reporting for a singular IDP Series device through IDP Reporter.

IDP Reporter is a Java application that has been preinstalled on your IDP Series device. You can access IDP Reporter through a Web interface. Like NSM Report Manager, IDP Reporter contains predefined reports and enables you to create custom reports based on log fields.

For details on accessing and using IDP Reporter, see the [IDP Reporter User's Guide](#).

**Related  
Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)
- [Application Volume Tracking Overview on page 22](#)

The following related topic is included in the *IDP Series Administration Guide*:

- IDP Reporter Task Summary

## CHAPTER 3

# Simulation Mode

- [Simulation Mode Overview on page 33](#)

### Simulation Mode Overview

---

Simulation mode is not a deployment mode, but rather an operational mode. The following sections give an overview of simulation mode:

- [Topology on page 33](#)
- [Purpose on page 33](#)
- [Configuration Overview on page 34](#)
- [Logging on page 34](#)

### Topology

The purpose of simulation mode is to enable you to evaluate expected results when you deploy the IDP Series device in transparent mode or sniffer mode. Therefore, in your network topology, you install and connect the IDP Series device where you intend to deploy it in transparent (in-path) or sniffer mode (out-of-path).

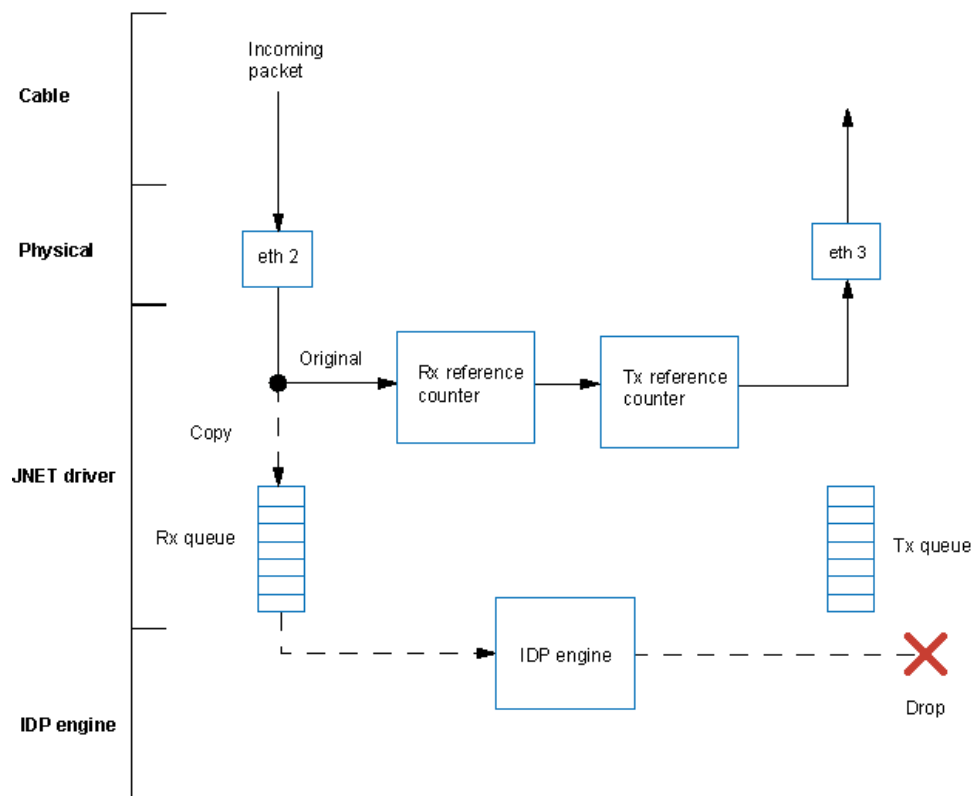
### Purpose

You operate an IDP Series device in simulation mode in the following situations:

- When you first deploy the IDP Series device in your network and you want to evaluate the security actions it takes without disrupting traffic.
- When you implement a new feature or change a security policy and you want to evaluate the impact without disrupting traffic.
- As a workaround to avoid traffic outages when IDP processing is resulting in crashes and other failures.

In simulation mode, when the IDP Series device receives a packet, it makes a copy. It transmits the original packet uninspected through the egress interface and enqueues the duplicate packet into the JNET driver receive queue to be processed by the IDP engine. The IDP engine inspects the traffic against your security policy rules and implicit rules, and it generates logs when rules match. The IDP engine then drops the copy of the packet. [Figure 7 on page 34](#) illustrates packet processing in simulation mode.

Figure 7: Packet Processing in Simulation Mode



**NOTE:** Because of packet queueing, when simulation mode is turned on, a few packets that are queued for processing and forwarding might be dropped. This results in retransmission depending on Layer 4 or Layer 7 behavior. When simulation mode is turned off, a few duplicate packets might be forwarded.

## Configuration Overview

You use the CLI to enable or disable simulation mode. Simulation mode is disabled by default. You do not need to restart the IDP engine (`idp.sh`) or push a policy to enable or disable simulation mode.

## Logging

In logs, the string [Simulation Mode] appears in the Details column, along with the details of the event. [Figure 8 on page 35](#) shows a simulation mode log in the NSM log viewer. You can use NSM log and report filters to create log views and reports that filter for (or filter out) simulation mode logs.

Figure 8: NSM Log Viewer: Simulation Mode Logs

**Log Viewer [3-IDP/DI]**

Src Addr	Nat Dst Addr	Details	Category	Subcategory	Severity	Device	Comment
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth2' ,alias=eth2' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth2' ,alias=eth2' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth2' ,alias=eth2' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...]	Predefined	Unnamed	Info	idp-10.209.85.9	

Timeline: Jul 12, Jul 13, Jul 14, Jul 15, Jul 16. Current time: 7/16/10 2:15:24 AM. Tailing Logs:

Summary: **Predefined :: Unnamed**

No Description

Variable Data:

Header Checksum	0xec5a
Header Code	210
Header Type	14
N Id	1234
N Seq	1

#### Related Documentation

The following related topics are included in *IDP Series Deployment Scenarios*:

- Sniffer Mode Overview
- Transparent Mode Overview

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Example: Getting Started with Simulation Mode on page 121](#)
- [Example: Using Simulation Mode to Maximize Uptime on page 579](#)

The following related topic is included in the *IDP Series Administration Guide*:

- Simulation Mode Task Summary





## CHAPTER 4

# Security Policy Basics

- [Understanding Non-Policy-Based Drops on page 37](#)
- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Understanding the Number of Available and Installed Policies on page 43](#)
- [Using Application Identification on page 43](#)
- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Using the Recommended Security Policy on page 46](#)
- [Using Other Security Policy Templates on page 47](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

### Understanding Non-Policy-Based Drops

---

The IDP Series device inspects traffic that traverses it and takes action according to:

- Implicit rules
- Protocol anomaly threshold settings
- Security policy rules

[Table 14 on page 38](#) summarizes implicit rules that drop traffic. The related topics listed provide information about protocol anomaly threshold settings and IDP security policy rules.

Table 14: Non-Policy-Based Drops

Implicit Rule	Description
Layer 2 traffic (when bypass not enabled)	<p>Enabled: Dropped by default. Configurable in ACM.</p> <p>When the IDP Series device is turned on and is operating normally, the traffic interfaces process TCP/IP traffic according to implicit traffic anomaly rules and explicit security policy rules. For Layer 2 connections, the interfaces process traffic, drop it, or pass it through (uninspected), according to the following rules:</p> <ul style="list-style-type: none"> <li>The interfaces process Address Resolution Protocol (ARP) and Internet Protocol (IPv4) traffic for inspection and process according to implicit and explicit rules.</li> <li>By default, the interfaces drop all other Layer 2 traffic.</li> </ul> <p>When Layer 2 bypass is enabled, the IDP Series device passes through Layer 2 packets related to bypass and high availability deployments (such as heartbeats or Bridge Protocol Data Unit (BPDU) packets), and non-IPv4 packets and packets related to switching and routing protocols, such as IPv6, internetwork packet exchange (IPX), Cisco Discovery Protocol (CDP), and interior gateway routing protocol (IGRP), and so forth.</p> <hr/> <p>Counter: If you do not enable Layer 2 bypass, you can use the following counters to observe Layer 2 drops:</p> <pre>[root@default host ~]# scio counter get kpp   grep sc_kpp_jpkt_free sc_kpp_jpkt_free          1374077</pre> <pre>[root@default host ~]# scio counter get kpp   grep sc_kpp_other sc_kpp_other              305</pre>
Invalid IP header	<p>Enabled: By default.</p> <hr/> <p>Counter:</p> <pre>[root@default host ~]# scio counter get kpp   grep sc_kpp_bad_ip_header sc_kpp_bad_ip_header      0</pre> <hr/> <p>Event logs: None</p> <hr/> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <b>/var/idp/device/sysinfo/logs/</b>.</p>
Invalid TCP header	<p>Enabled: By default.</p> <hr/> <p>Counter:</p> <pre>[root@default host ~]# scio counter get reass   grep sc_reass_bad_tcp_header sc_reass_bad_tcp_header   0</pre> <hr/> <p>Event logs: Yes</p> <hr/> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <b>/var/idp/device/sysinfo/logs/</b>.</p>

Table 14: Non-Policy-Based Drops (*continued*)

Implicit Rule	Description
TCP checksum error	<p>Enabled: Logging for this event is disabled by default. Use the following command to enable logging for this event:</p> <pre>[root@default host ~]# scio const set sc_log_implicit_pkt_drop 1</pre> <p>Counter:</p> <pre>[root@default host ~]# scio counter get reass   grep sc_reass_bad_tcp_csum sc_reass_bad_tcp_csum      0</pre> <p>Event logs: If you enable logging, event logs are generated and sent to NSM and/or a syslog server.</p> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <b>/var/idp/device/sysinfo/logs/</b>.</p>
UDP checksum error	<p>Enabled: The counter for this event is disabled by default. Use the following command to enable it:</p> <pre>[root@default host ~]# scio const set sc_enable_udp_csum 1</pre> <p>Counter:</p> <pre>[root@default host ~]# scio counter get flow   grep sc_flow_bad_udp_csum sc_flow_bad_udp_csum      0</pre> <p>Event logs: If you enable logging, event logs are generated and sent to NSM and/or a syslog server.</p> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <b>/var/idp/device/sysinfo/logs/</b>.</p>
ICMP source quench	<p>Dropping ICMP source quench messages is disabled by default. Use the following command to enable it:</p> <pre>[root@default host ~]# scio const -s s0:flow set sc_icmp_drop_source_quench 1</pre> <p>Event logs: Event logs are generated and sent to NSM or a syslog server.</p> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <b>/var/idp/device/sysinfo/logs/</b>.</p>
TTL error	<p>Enabled: By default.</p> <p>Counter:</p> <pre>[root@default host ~]# scio counter get kpp   grep sc_kpp_ttl_error sc_kpp_ttl_error          0</pre> <p>Event logs: None</p> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <b>/var/idp/device/sysinfo/logs/</b>.</p>

Table 14: Non-Policy-Based Drops (*continued*)

Implicit Rule	Description
Memory limit of busy packet list	<p>Enabled: By default.</p> <hr/> <p>Counter:</p> <pre>[root@defaulthost ~]# scio counter get kpp   grep sc_kpp_busy_drop sc_kpp_busy_drop          0</pre> <hr/> <p>Event logs: None</p> <hr/> <p>Debug logs: None</p>
Dropped by reassembly module when per flow memory overflows	<p>Enabled: By default.</p> <hr/> <p>Counter:</p> <pre>[root@defaulthost ~]# scio counter get kpp   grep sc_kpp_fdrops sc_kpp_fdrops             0</pre> <hr/> <p>Event logs: None</p> <hr/> <p>Debug logs: None</p>
Global reassembly memory overflow	<p>Enabled: By default.</p> <hr/> <p>Counter:</p> <pre>[root@defaulthost ~]# scio counter get reass   grep sc_reass_ovflw_drop sc_reass_ovflw_drop       0</pre> <hr/> <p>Event logs: None</p> <hr/> <p>Debug logs: If debugging is enabled, debug logs are generated and saved to <code>/var/idp/device/sysinfo/logs/</code>.</p>
Transmit failure where packet has already been freed	<p>Enabled: By default.</p> <hr/> <p>Counter:</p> <pre>[root@defaulthost ~]# scio counter get kpp   sc_kpp_transmit_error sc_kpp_transmit_error     0</pre> <hr/> <p>Event logs: None</p> <hr/> <p>Debug logs: None</p>

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)

The following related topic is included in the *IDP Series Administration Guide*:

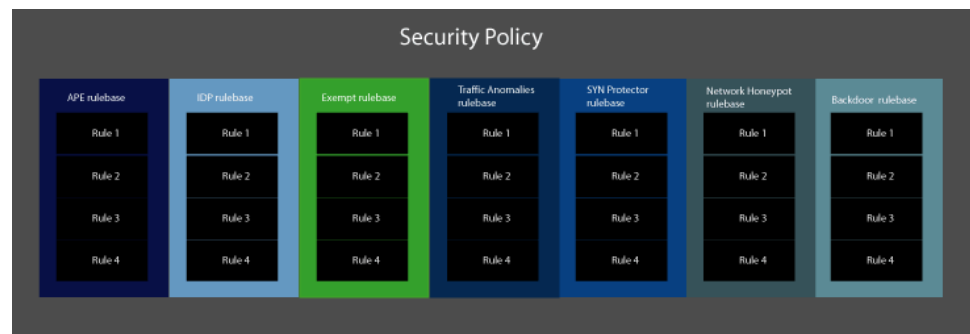
- [Modifying Protocol Anomaly Thresholds on page 351](#)

## Understanding the Components of an IDP Security Policy

An IDP security policy defines how an IDP Series device handles network traffic. It allows you to enforce various attack detection and prevention techniques on traffic that traverses your network.

[Figure 9 on page 41](#) illustrates the components of an IDP security policy.

**Figure 9: Security Policy Components**



A security policy is made up of one or more rulebases. A *rulebase* is an ordered set of rules that use a particular detection method to identify and prevent attacks.

[Table 15 on page 41](#) describes the IDP security policy rulebases. A security policy can contain only one instance of any rulebase type.

**Table 15: IDP Security Policy Rulebases**

Rulebase	Description
APE rulebase	<p>Enables you to implement application policy enforcement rules. You can use APE rules to manage sessions based on application and/or user role. You can terminate matching sessions or limit bandwidth available to them.</p> <p>See <a href="#">“Understanding the APE Rulebase” on page 69</a>.</p>
IDP rulebase	<p>Protects your network from attacks by using attack objects to detect known and unknown attacks. Juniper Networks Security Center (J-Security Center) provides predefined attack objects that you can use in IDP rules. You can also configure your own custom attack objects.</p> <p>See <a href="#">“Understanding the IDP Rulebase” on page 55</a>.</p>
Exempt rulebase	<p>You configure rules in this rulebase to exclude known false positives or to exclude a specific source, destination, or attack object from matching an IDP rule. If traffic matches a rule in the IDP rulebase, IDP attempts to match the traffic against the Exempt rulebase before performing the action specified.</p> <p>See <a href="#">“Understanding the Exempt Rulebase” on page 67</a>.</p>

Table 15: IDP Security Policy Rulebases (*continued*)

Rulebase	Description
Traffic Anomalies rulebase	<p>Protects your network from attacks by using traffic flow analysis to identify attacks that occur over multiple connections and sessions (such as scans).</p> <p>See <a href="#">“Understanding the Traffic Anomalies Rulebase” on page 97.</a></p>
SYN Protector rulebase	<p>Protects your network from SYN-floods by ensuring that the three-way handshake is performed successfully for specified TCP traffic. If your network is vulnerable to SYN-flood attacks, use the SYN-Protector rulebase to prevent it.</p> <p>See <a href="#">“Understanding the SYN Protector Rulebase” on page 91.</a></p>
Network Honeypot rulebase	<p>Protects your network by impersonating open ports on existing servers on your network, alerting you to attackers performing port scans and other information-gathering activities.</p> <p>See <a href="#">“Understanding the Network Honeypot Rulebase” on page 103.</a></p>
Backdoor rulebase	<p>Protects your network from mechanisms installed on a host computer that facilitate unauthorized access to the system. Attackers who have already compromised a system typically install backdoors (such as Trojans) to make future attacks easier. When attackers send and retrieve information to and from the backdoor program (as when typing commands), they generate interactive traffic that IDP can detect.</p> <p>See <a href="#">“Understanding the Backdoor Rulebase” on page 85.</a></p>



**NOTE:** Firewall rulebases, visible in NSM, do not apply to standalone IDP Series devices.

*Rules* are instructions that provide context to detection methods. Rules specify:

- A match condition that determines which traffic to inspect
- Attack objects that determine what to look for (IDP rulebase and Exempt rulebase)
- Actions and operation modes that determine what to do when traffic matches a rule
- Notification options, including logs, alerts, and packet captures

#### Related Documentation

The following additional related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Number of Available and Installed Policies on page 43](#)
- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following additional related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

## Understanding the Number of Available and Installed Policies

In NSM, you can create and save an unlimited number of security policies, and these policies are available to be installed on IDP Series devices.

You can install the same security policy on an unlimited number of IDP Series devices.

You can install one security policy on an IDP Series device. However, when you push a security policy update, you might observe more than one policy in place for a period after the update.

By default, the IDP system resets the flow table when you install a new policy. When the flow table is reset, existing sessions are passed through uninspected. For IDP75 and IDP200, you cannot override the default.

For high-end appliances, you can unset this default to avoid passing through sessions uninspected. Go to NSM Device Manager > Run-time Parameters and unselect **Reset flow table with policy load/unload**. If you unset this default, when you load a new policy, the IDP flow table maintains sessions belonging to the previously installed policy as well as the newly installed policy. The IDP process engine continues to use the previously installed security policy to inspect previous sessions; and uses the newly installed security policy to inspect new sessions. When the previously installed policy is no longer in use, it is unloaded and all traffic is inspected using the newly installed policy. For IDP8200 and IDP250, the IDP engine can maintain flows for as many as two security policies. For IDP1100, IDP800, and IDP600, the IDP engine can maintain flows for as many as four security policies.

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)
- [Modifying the IDP Series Device Configuration on page 351](#)

## Using Application Identification

The application identification feature enables the IDP engine to detect applications running on standard or nonstandard ports. Port-independent application identification enhances both security and manageability by eliminating the need to manually and comprehensively configure application-port mapping for the service objects and application objects used in the IDP rulebase and APE rulebase rules.

The application identification feature uses application signatures provided by the Juniper Security Center team (J-Security Center) to identify the session application. Beginning with IDP OS Release 5.1, the application identification feature can match extended application signatures used in APE rulebase rules. *Extended application* signatures are also called *nested application* signatures. The predefined extended application signatures

developed for IDP OS Release 5.1 include the most prevalent Web 2.0 applications running over HTTP. If your security policy includes APE rules configured to match extended application signatures, the application identification process identifies and generates the following HTTP contexts: http-url-parsed, http-url-parsed-param-parsed, http-header-host, and http-header-content-type. The application identification feature can then match application signature patterns in those contexts.

J-Security Center updates application signatures and develops new ones as necessary. Beginning with IDP OS Release 5.1, you can use NSM to browse predefined application objects, predefined extended application objects, and application groups. You can also use NSM to create custom application definitions, if needed. You cannot, however, create custom extended application definitions.

When the application identification feature identifies a new application, it caches the result (the destination address, port, protocol, and service) to reduce processing for subsequent sessions. The application cache and extended application cache are maintained separately.

When the IDP engine processes security policy rules, it examines the session, beginning with the first packet, to identify a match. To match service or application, the IDP engine first compares the session against the application identification cache to identify the application. If the session does not match the application identification cache, the IDP engine processes the session against the application signatures. If the IDP engine is still unable to determine the application, it uses the standard application protocol and port.

In IDP rulebase rules, with application identification enabled, you set the service object in rules to **Default** to allow the application identification feature to identify the correct service. If you set service to a specific service object, application identification is not applied and the rule is processed using the service object properties.

In APE rulebase rules, with application identification enabled, you set the service object in rules to **Default** and specify rules based on application or extended application. If you disable application identification and specify a match based on application, the IDP engine uses the standard application protocol and port for the application. If the application you are interested in is not listed, you can create a custom application object to match against application properties that you define.

The application identification feature is enabled by default, and we recommend you use this feature. To support lab experimentation and troubleshooting, you can disable application identification and extended application identification, and you can tune the following settings:

- Maximum number of sessions that utilize application identification
- Maximum memory used by application identification
- Maximum memory for saving TCP or UDP packets per session

For information on tuning these parameters, see the [scio const](#) reference page in the *IDP Series Administration Guide*.



**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Rulebase Example: Using Application Identification on page 153](#)
- [Using Application Objects on page 73](#)
- [Understanding the IDP Rulebase on page 55](#)
- [J-Security Center Updates Overview on page 19](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)
- [Specifying Rule Match Conditions \(NSM Procedure\) on page 215](#)
- [scio const on page 505](#)

---

## Understanding the Rule-Matching Algorithm

---

The IDP process engine processes rulebases in the following order:

1. Application Policy Enforcement (APE) rulebase (terminal)
2. Traffic Anomalies rulebase (terminal)
3. SYN Protector rulebase (terminal)
4. Network Honeypot rulebase (terminal)
5. IDP rulebase (nonterminal)
6. Exempt rulebase (nonterminal)
7. Backdoor rulebase (terminal)

For terminal rulebases, the IDP rule-matching algorithm evaluates rules according to the ordered list to identify matches. As soon as the algorithm identifies a match, it applies the rule and terminates rule matching. For example, if a terminal rule 1 matches, it is applied, and rule 2 is not consulted.

For nonterminal rulebases, the IDP rule-matching algorithm also evaluates rules according to the ordered list to identify matches. However, even if it finds a match, it continues down the list to identify additional matches.

The IDP rulebase includes the option to mark a rule as a terminal rule. When a terminal rule matches source, destination, and service, IDP applies the rule and terminates rule matching. It does not matter whether the traffic matches the attack objects.

In the IDP rulebase, you can set the terminal rule flag for the following purposes:

- To disregard traffic that originates from a particular trusted source (however, an Exempt rulebase rule might be a better choice).
- To exit rule processing when you want a particular match to always trigger a particular action and no other, such as a drop connection action.

- To exit rule processing when your rule specifies precise destination addresses and precise services, and you know that the subsequent rules do not apply.

Use caution when specifying the terminal flag. You can inadvertently leave your network open to attacks by creating an inappropriate terminal rule. Be particularly careful about terminal rules using the value **Any** for both the source and destination. Terminal rules should appear near the top of the rulebase, before other rules that would match the same traffic.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)
- [Understanding APE Rulebase Match Conditions on page 70](#)
- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [Understanding Backdoor Rulebase Match Settings on page 87](#)
- [Understanding SYN Protector Rulebase Match Settings on page 93](#)
- [Understanding Traffic Anomalies Rulebase Match Conditions on page 99](#)
- [Understanding Network Honeypot Rulebase Match Settings on page 104](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

## Using the Recommended Security Policy

The highly respected Juniper Networks Security Center team (J-Security Center) provides the default IDP security policy—named Recommended. We advise that you use this policy (or customize it) to protect your network from the likeliest and most dangerous attacks.

[Table 16 on page 46](#) summarizes the properties of the Recommended security policy.

**Table 16: Recommended Security Policy Definition**

Property	Value
Rulebase	IDP rulebase
Rules	Nine rules, distinguished by attack object
Source	Any
Service	Default, meaning the matching property is based on the service bindings of the attack object specified by the rule
Destination	Any

Table 16: Recommended Security Policy Definition (*continued*)

Property	Value
Attack objects	Recommended IP, Recommended TCP, Recommended ICMP, Recommended HTTP, Recommended SMTP, Recommended DNS, Recommended FTP, Recommended POP3, Recommended IMAP, Recommended Trojan, Recommended Virus, Recommended Worm  <i>NOTE:</i> All of the attack objects included in the predefined policies are client-to-server attacks.
Action	Recommended, meaning the action is specified by the attack object
Notification	Logging

If you prefer, you can copy this security policy and use it as a template for a custom security policy tailored for your network.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the IDP Rulebase on page 55](#)
- [IDP Rulebase Example: Specifying the Default Service on page 154](#)
- [IDP Rulebase Example: Using Application Identification on page 153](#)
- [IDP Rulebase Example: Using Recommended Attack Objects on page 155](#)
- [IDP Rulebase Example: Using Recommended Actions on page 156](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

## Using Other Security Policy Templates

NSM includes security policy templates you can use as the basis for a custom security policy tailored for your network. Template rules include a set of attack objects and logically associated IDP actions. If you choose to use these templates, we advise you to customize them for your deployment. At a minimum, you should change the destination IP setting from Any to the IP addresses for specific servers you want to protect.

[Table 17 on page 47](#) describes IDP security policy templates.

Table 17: IDP Security Policy Templates

Template	Description
all_with_logging	Includes all attack objects and enables packet logging for all rules. This policy is provided for lab use and is not recommended in production.
all_without_logging	Includes all attack objects but does not enable packet logging.

Table 17: IDP Security Policy Templates (*continued*)

Template	Description
dmz_services	Protects a typical DMZ environment.
dns_server	Protects DNS services.
file_server	Protects file sharing services, such as SMB, NFS, FTP, and others.
getting_started	Contains very open rules. Useful in controlled lab environments, but should not be deployed on heavy traffic live networks.
idp_default	Contains a set of attack groups that balances security and performance.
web_server	Protects HTTP servers from remote attacks.



**NOTE:** All of the attack objects included in the predefined policies are client-to-server attacks.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using the Recommended Security Policy on page 46](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

## Example: Fine-Tuning a Security Policy

This topic provides a suggested workflow for getting started and fine-tuning a security policy. It includes the following subtopics:

- [Fine-Tuning Security Policies Process Overview on page 48](#)
- [Getting Started with the Recommended Security Policy on page 49](#)
- [Refining Rule Matching Properties on page 50](#)
- [Reducing False Positives on page 50](#)
- [Adding Rulebases on page 53](#)

### Fine-Tuning Security Policies Process Overview

You want to tune a security policy so that it is:

- Comprehensive—Detects all possible attacks on specific hosts in your network.
- Optimized—Each attack object specified in an IDP rulebase rule has a performance cost. In general, you want more rules with a few attack objects in each rather than fewer rules with many attack objects. In addition, we recommend that a single rule

includes only the attack objects that are applicable to the rule destination server and only those of a severity that concerns you.

- **Precise**—Generates few false positives.
- **Maintainable**—As you refine your rules, you want to leverage as much of the predefined logic as possible. In your IDP rulebase rules, for example, you want to use the application identification feature, dynamic attack object groups, recommended attack objects, and recommended actions as much as possible, knowing the Juniper Networks Security Center team updates these as needed (even daily).

Fine-tuning is an iterative process. The process involves the following steps:

1. [Getting Started with the Recommended Security Policy on page 49](#)
2. [Refining Rule Matching Properties on page 50](#)
3. [Reducing False Positives on page 50](#)
4. [Adding Rulebases on page 53](#)

## Getting Started with the Recommended Security Policy

When you add the IDP Series device to the NSM Device Manager, NSM automatically pushes the recommended policy to the IDP Series device. The recommended policy protects destination servers from the most frequent and severe attacks.

[Table 18 on page 49](#) summarizes the properties of the Recommended security policy.

**Table 18: Recommended Security Policy Definition**

Property	Value
Rulebase	IDP rulebase
Rules	Nine rules, distinguished by attack object
Source	Any
Destination	Any
Service	Default, meaning the matching property is based on the service bindings of the attack object specified by the rule
Attack objects	Recommended IP, Recommended TCP, Recommended ICMP, Recommended HTTP, Recommended SMTP, Recommended DNS, Recommended FTP, Recommended POP3, Recommended IMAP, Recommended Trojan, Recommended Virus, Recommended Worm  <b>NOTE:</b> All of the attack objects included in the predefined policies are client-to-server attacks.
Action	Recommended, meaning the action is specified by the attack object
Notification	Logging

## Refining Rule Matching Properties

The source and destination matching parameters for template rules are set to **Any**. These broad settings provide comprehensive protection, but they entail a performance cost and might result in more logs than are necessary. We recommend you customize these settings.

Run the Profiler for several days to gather information about the hosts and applications running in your network. After several days, you should have the data you need to complete the following tasks:

- Create NSM address objects that identify groups of internal servers. When you configure rules to examine client to server traffic, you specify the internal servers as the rule's destination servers.
- Create an address object that defines your internal network. When you configure rules to examine traffic from your network to hosts on the world wide web, you can specify the internal network address object as the rule's source.
- Create NSM service objects to identify services running on internal servers. In most cases, you can specify **Default** for service so the rule uses the service relevant to the attack object. However, there might be cases where you want to specify a service object or service group.
- Identify predefined attack groups related to services (or create your own attack group, if necessary).
- Refine the IDP rulebase rule set so that each rule is focused on a single destination server (client to server traffic) or service (server to client traffic).

## Reducing False Positives

A *false positive* is a log record that reflects an event on your network that you are not concerned about and no longer want to see in your logs. The IDP security policy found traffic that matched your rule, but over time you realize you do not need to track such events.

To determine whether a log is a false positive, you need to understand why the IDP Series device triggered the log and whether or not the traffic poses a real risk to the target server.

Suppose your security policy rule includes the following attack object: Predefined :: HTTP: Windows Media Services NSISlog.DLL Buffer Overflow. It generates a log when it identifies the attack pattern in traffic through the IDP Series device. Use the reference information in the details pane below the log table to learn more about the attack. You can click the hypertext linked name of the attack object in the summary tab to display reference information for the attack, as shown in [Figure 10 on page 51](#).

Figure 10: Using NSM Log Viewer Attack Reference Information

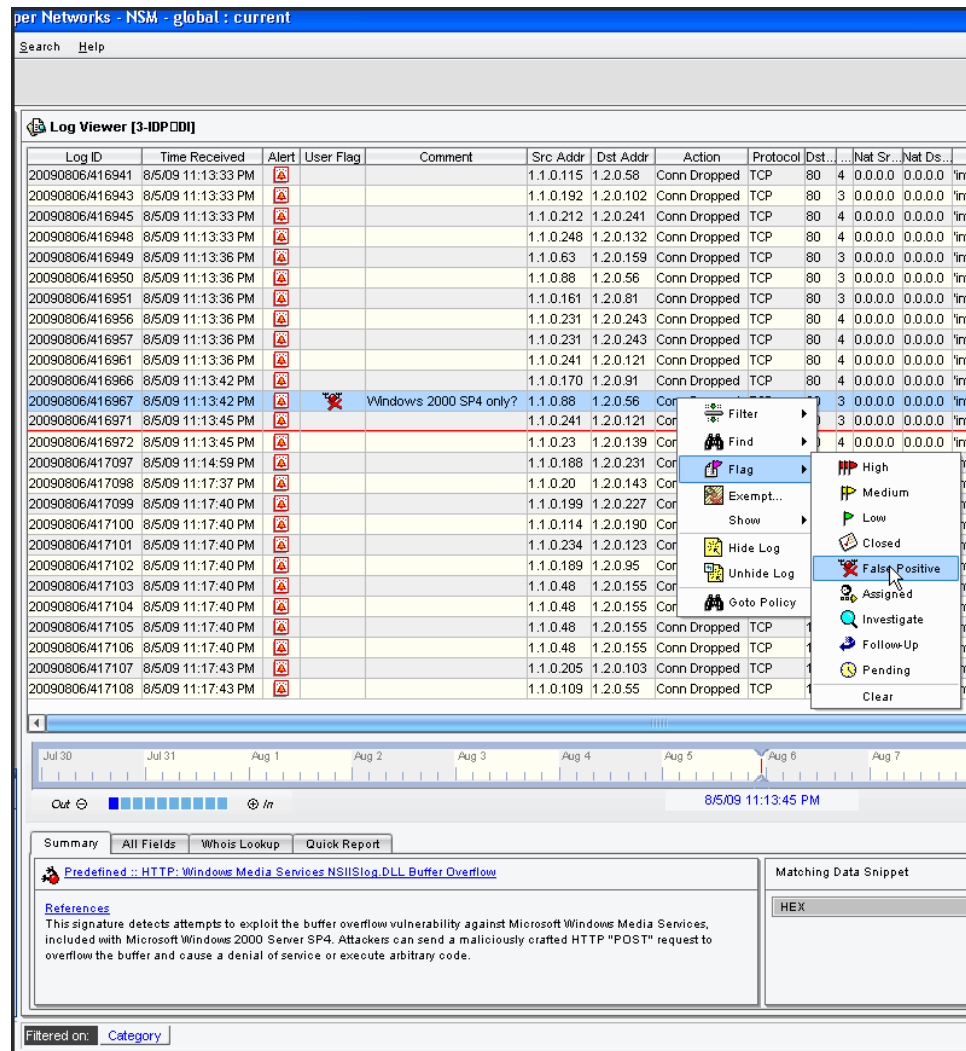
The screenshot shows the NSM Log Viewer interface. The main window displays a table of logs with columns: Log ID, Time Received, Alert, User Flag, Comment, Src Addr, Dst Addr, Action, Protocol, Dst, Src, Net, Dst, Details, Category, and Subcategory. A log entry with ID 20090806/416967 is highlighted, showing a 'Windows 2000 SP4 only?' comment and a red 'X' in the Alert column.

The detailed view of the selected log entry shows the following information:

- References:**
  - <http://online.securityfocus.com/bid/3035/discussion/>
  - <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0349>
  - <http://www.kb.cert.org/vuls/id/113716>
  - <http://www.microsoft.com/technet/security/bulletin/MS03-022.mspx>
  - <http://secunia.com/advisories/9115>
- Extended Description:**
  - Last Modified:** 2009-08-13
  - Impact:** Windows Media Services may expose IIS to remote arbitrary code execution if media logging is enabled.
  - Description:** Microsoft has reported a buffer overflow vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension handles incoming client requests. This could cause arbitrary code execution in IIS, which is exploitable through Media Services.
  - Technical Information:** Microsoft Media Services provides functionality for providing streaming media content to clients from IIS. It ships with a number of Microsoft Windows 2000 server releases and is also available for download for Windows NT. Microsoft has reported a buffer overflow vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension (msiislog.dll) handles incoming client requests. The logging facility may attempt to write excessive data to an undersized buffer when handling a malformed HTTP client request. This could trigger a denial of service or remote arbitrary code execution in IIS, which is exploitable through Media Services. The issue would occur in servers that are configured to provide logging of media requests. It is possible to exploit this issue by sending an overflow from HTTP POST request to the

In this example, we learn that the threat detected applies only to Microsoft Windows 2000 Server SP4. It is a false positive because all of the Windows servers in our network are Windows Server 2008. You can use the NSM Log Viewer flag and comment features to mark logs as false positives. In Figure 11 on page 52, we have marked the log ID 20090806/416967 as a false positive because the attack targets server versions not present in our network.

Figure 11: Using NSM Log Viewer Flag and Comment Features



There are a number of ways you can tune your security policy to reduce false positives. [Table 19 on page 52](#) summarizes some basic tunings.

Table 19: Actions to Take To Reduce False Positives

Type of False Positive	Tuning Required
You trust the source.	Add an Exempt rulebase rule to “whitelist” the trusted source.



Table 19: Actions to Take To Reduce False Positives (*continued*)

Type of False Positive	Tuning Required
The attack applies to a hardware or software version that does not match your destination server.	<p>You have many options:</p> <ul style="list-style-type: none"> <li>• Delete the attack from the rule.</li> <li>• Modify an attack group to exclude the object.</li> <li>• Add an Exempt rulebase rule to whitelist the non-offending attack object.</li> <li>• Modify rule action so traffic is stopped or permitted differently from before.</li> <li>• Modify the rule severity so that you can filter these events differently from before.</li> </ul>
Your team has already patched the vulnerability detected by the attack.	Same as previous.
Upon examination, benign traffic crosses thresholds that trigger protocol anomaly events.	Use the NSM Device Manager to modify protocol anomaly thresholds.

## Adding Rulebases

The IDP rulebase is the primary rulebase in an IDP security policy. When you have sufficiently tuned your IDP rulebase rules so that the security policy generates the level of logs you want, you can add additional rulebases to enable additional detection methods.

Take the same approach to tuning these additional rulebases. Instead of refining the group of attack objects that are relevant, you tune the IDP runtime parameters that set thresholds for detection mechanisms.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)
- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Understanding the Exempt Rulebase on page 67](#)
- [Using Attack Objects on page 60](#)
- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying the IDP Series Device Configuration on page 351](#)



## CHAPTER 5

# The IDP Rulebase

- [Understanding the IDP Rulebase on page 55](#)
- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [User-Role-Based Policy Feature Overview on page 58](#)
- [Using Attack Objects on page 60](#)
- [Understanding IDP Rulebase Actions on page 63](#)
- [Understanding IDP Rulebase Notification Options on page 65](#)

## Understanding the IDP Rulebase

---

The IDP rulebase employs an attack object database to support two robust detection methods: stateful signatures and protocol anomalies.

A *stateful signature* combines an attack pattern with service, context, and other properties into a signature attack object. As a result, the IDP system does not need to expend resources inspecting huge sections of network traffic where attacks cannot possibly be, and IDP produces very few false positives.

A *protocol anomaly* is a deviation from protocol standards established by the Internet Engineering Taskforce (IETF) Request for Comment (RFC) process. Traffic that does not adhere to these standards is suspicious because most legitimate applications adhere to the standards, and anomalies can fairly be regarded as purposeful attempts to evade an intrusion detection system (IDS). IDP protocol-anomaly attack objects find traffic that deviates from IETF RFC standards.

When you create rules for the IDP rulebase, you specify:

- A source/destination/service match condition
- Attack objects
- Action
- Notification options

For complete procedures on configuring IDP rulebase rules, see the *IDP Series Administration Guide*.

**Related Documentation** The following additional related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [Using Application Identification on page 43](#)
- [Using Attack Objects on page 60](#)
- [Understanding IDP Rulebase Actions on page 63](#)
- [Understanding IDP Rulebase Notification Options on page 65](#)
- [IDP Rulebase Example: Using Application Identification on page 153](#)
- [IDP Rulebase Example: Specifying the Default Service on page 154](#)
- [IDP Rulebase Example: Using Recommended Attack Objects on page 155](#)
- [IDP Rulebase Example: Using Recommended Actions on page 156](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)

## Understanding IDP Rulebase Rule Match Settings

The IDP engine inspects the session beginning with the first packet to determine whether the session matches a rule. If the session matches all rule settings for source, destination, service, and VLAN tag ID, the IDP system decodes the traffic and inspects the session packets for the attack objects specified in the rule. If the session matches only some of the rule settings, the rule is not a match.

[Table 20 on page 56](#) provides guidelines for setting IDP rulebase match conditions.

**Table 20: IDP Rulebase Match Condition Guidelines**

Setting	Guideline
From zone/To zone	Not applicable for standalone IDP Series devices.
Source	<p>Requires one of the specified source IP addresses to match the session for the rule to be applied. You can add address objects for hosts, groups, or network address ranges.</p> <p>In most cases, to detect incoming attacks that target your internal network, specify <b>Any</b>. Specifying <b>Any</b> means you are not using source as a key to your match.</p> <p>To detect traffic from spyware that has affected hosts in your internal network, specify internal network addresses as the source.</p> <p>To detect attacks between two networks you manage, specify multiple addresses. The more specific you are in defining the source and destination of an attack, the more you reduce false positives.</p> <p><b>NOTE:</b> You must choose between source IP address or user role as match criteria for a rule. You cannot configure both for one rule.</p>

Table 20: IDP Rulebase Match Condition Guidelines (*continued*)

Setting	Guideline
User Role	<p>Requires one of the specified user roles to match the session for the rule to be applied. If a value for User Role matches, the Source parameter is not consulted.</p> <p>You must choose to configure either source IP address or user role as match criteria for a rule. User role-based rules are evaluated before IP address-based rules. If a user-role based rule matches, the rule is applied and IP address-based rules are not consulted.</p> <p><b>NOTE:</b> Matching based on user role depends on integration with the Juniper Networks IC Series Unified Access Controller (UAC) appliance.</p>
Destination	<p>Requires one of the specified destination IP addresses to match the session for the rule to be applied. You can add address objects for hosts, groups, or network address ranges.</p> <p>In most cases, specify the hosts or servers you want to protect.</p> <p>Specify <b>Any</b> to not use destination as a key to your match. For example, it would be impossible to predict the destination IP address for traffic resulting from spyware in your internal network. Specify <b>Any</b> for rules that target spyware attacks.</p>
Service	<p>Requires one of the specified services to match the session for the rule to be applied. Services are Application Layer protocols that define how data is structured as it travels across the network. IDP can inspect services that use TCP, UDP, RPC, and ICMP transport layer protocols. If the application running on the destination server uses standard ports, you can select from predefined services. If the application running on the destination server uses nonstandard ports, you must create a custom service object.</p> <p><b>TIP:</b> Specify <b>Default</b> to match the service(s) specified in the rule attack object(s). If the application identification feature is enabled, the IDP process engine identifies services even if they are running on nonstandard ports.</p> <p>If you disable application identification and specify <b>Default</b>, the IDP process engine assumes that standard ports are used for the service.</p> <p><b>NOTE:</b> If you disable application identification and your service uses nonstandard ports, you must create custom service objects. For procedures, see the <i>IDP Series Administration Guide</i>.</p> <p>Specify <b>Any</b> to not use service as a key to your match.</p>
VLAN	<p>Requires one of the specified VLAN tags to match the session for the rule to be applied.</p> <p>Specify <b>Any</b> to not use VLAN tag as a key to your match.</p>



**TIP:** You can use Profiler to identify the hosts and services that are included in your network. In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM online Help.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the IDP Rulebase on page 55](#)

- [User-Role-Based Policy Feature Overview on page 58](#)
- [Using Application Identification on page 43](#)
- [IDP Rulebase Example: Specifying the Default Service on page 154](#)
- [IDP Rulebase Example: Using Application Identification on page 153](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Specifying Rule Match Conditions \(NSM Procedure\) on page 215](#)

---

## User-Role-Based Policy Feature Overview

The user role-based policy feature depends on integration with the Juniper Networks IC Series Unified Access Control (UAC) appliance. This feature requires collaboration with the UAC administrator.

The user role-based policy feature enables you to specify user roles as match criteria in IDP rulebase and application policy enforcement (APE) rulebase rules. Matching based on user role rather than IP address both simplifies and finely tunes your rules. In many networks, the IP address is dynamically assigned. To protect your network, you would have to cast a wide net for traffic sources. In most cases, you would specify a subnet mask or specify **Any** source (in the latter case, this means you really are not matching on source). For the purpose of intrusion detection and prevention, a wide net is not necessarily a bad thing: you do want to inspect any session that could potentially contain an attack. Use of role-based rules with a terminal match, however, will improve performance by providing faster matching with specific source targets and rulebase termination. In addition, you are likely to find that user role-based logs are easier to analyze because they provide visibility into the user role associated with an attack event or application usage.

UAC integration with IDP Series devices also improves end user experience authenticating to your network. In a UAC deployment, you use the Host Checker feature to quarantine users with vulnerable hosts. Instead of using a firewall to shut down access to network resources, you can use IDP security policies to enable access and inspect the traffic to guard against threats.

In the APE rulebase, role-based rules are indispensable to supporting the business cases that demand a nuanced approach to application policy enforcement. They enable you to enforce business policies such as “Contractors, Part-Time, and Temporary employees may not use peer-to-peer filesharing applications; full-time employees may use them, but only with a limited pool of bandwidth.”

To deploy the user role-based feature:

1. Read the release notes for the IDP OS and UAC releases to verify version compatibility requirements.
2. Deploy a UAC solution for user access to the network. For details, see the *Unified Access Control Administration Guide*.

3. Use the UAC user interface to create the user roles you want to use in your security policy:
  - For security rules, you want to leverage results of the Host Checker to map users to roles that identify vulnerabilities, such as “Laptop Users,” “Unauthorized Instant Messenger Installed,” or “Windows XP Patch Required.”
  - For application policy enforcement rules, you want to map users to roles that reflect the business rule, such as Contractor, Part-Time, and Temporary.

For details, see the *Unified Access Control Administration Guide* or UAC online Help.

4. Configure communication between the UAC appliance and the IDP Series device so you can use the IDP role-based policy feature:
  - From the IDP Series side, you use the Appliance Configuration Manager (ACM) to generate a one-time password the UAC appliance can use to connect to the IDP Series device.
  - From the UAC side, you configure the connection to the IDP Series device, specifying the IP address, port 7103, and the one-time password.

For details, see the UAC online Help.

5. Use the IDP Series command-line interface to verify integration. You can use CLI commands to verify connectivity with the IC Series device and to display the user session table, which is populated by the IC Series device.

For details, see the *IDP Series Administration Guide*.

6. In NSM, create a security policy with role-based rules. The roles you specify are the roles created and managed in UAC.
7. Push the security policy from NSM to the IDP Series device.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [IDP Rulebase Example: User-Role-Based Policies on page 157](#)
- [APE Rulebase Example: Aggregate and Per-Subscriber Rate Limits on page 170](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Specifying Rule Match Conditions \(NSM Procedure\) on page 215](#)
- [Configuring Advanced Settings for the User-Role-Based Policy Feature on page 322](#)
- [Verifying Integration with an IC Series Unified Access Control Appliance on page 493](#)

The following related topic is included in the *IDP Series Deployment Scenarios*:

- [Deploying IDP Series with an IC Series Device to Implement User-Role-Based Security Policies](#)

## Using Attack Objects

---

If the session matches rule settings for source, destination, service, and VLAN tag ID, the IDP engine decodes the traffic and inspects the session packets for the attack objects specified in the rule. The following topics provide guidelines for using attack objects in IDP rulebase rules:

- [Attack Objects Overview on page 60](#)
- [Understanding Predefined Attack Objects and Attack Object Groups on page 61](#)
- [Using Attack Object Groups on page 61](#)
- [Using Custom Attack Objects on page 62](#)

### Attack Objects Overview

When traffic matches an IDP rulebase source/destination/service condition, the IDP engine inspects the traffic for the attack objects you specify.

A *signature attack object* detects known attacks using stateful attack signatures. An attack signature is a pattern that always exists within an attack; if the attack is present, so is the attack signature. With stateful signatures, the IDP engine can look for the specific protocol or service used to perpetrate the attack, the direction and flow of the attack, and the context in which the attack occurs. Stateful signatures produce few false positives because the context of the attack is defined, eliminating huge sections of network traffic in which the attack would not occur.

A *protocol anomaly* identifies unusual activity on the network. It detects abnormal or ambiguous messages within a connection according to the set of rules for the particular protocol being used. Protocol anomaly detection works by finding deviations from protocol standards, most often defined by RFCs and common RFC extensions. Most legitimate traffic adheres to established protocols. Traffic that does not, produces an anomaly, which may be created by attackers for specific purposes, such as evading an IPS. You cannot create protocol anomaly objects. You can specify a predefined protocol anomaly object as a component of a compound attack object.

A *compound attack object* combines multiple signatures and/or protocol anomalies into a single object. Traffic must match all of the combined signatures and/or protocol anomalies to match the compound attack object; you can specify the order in which signatures or anomalies must match. Use compound attack objects to refine your IDP policy rules, reduce false positives, and increase detection accuracy. A compound attack object enables you to be very specific about the events that need to occur before the IDP engine identifies traffic as an attack. You can use **And**, **Or**, and **Ordered and** operations to define the relationship among different attack objects within a compound attack and the order in which events occur.

Attack object definitions also include data fields to help you group and manage attack objects and use them in security policies. These data fields include category, severity, keywords, and a recommended flag.



*Predefined attack objects* provided by the Juniper Networks Security Center (J-Security Center) team also contain a recommended action for the IDP Series device to take against the attack session.

*Custom attack objects* are ones you create, if your security policy requires more or less protection, or more or less accounting than what the predefined attack objects provide.

Both predefined and custom attack objects are stored in the attack object database.

When you add attack objects to an IDP rulebase rule, you can add attack objects by group or individually.

## Understanding Predefined Attack Objects and Attack Object Groups

The Juniper Networks Security Center (J-Security Center) team has developed more than 600 attack objects and these are included in the attack object database used in IDP security policies.

[Table 21 on page 61](#) describes the attack object groups provided by the J-Security Center.

**Table 21: Predefined Attack Object Groups**

Group	Contents
Attack Type	Contains two subgroups: anomaly and signature. Within each subgroup, attack objects are grouped by severity.
Category	Contains subgroups based on category. Within each category, attack objects are grouped by severity.
Operating System	Contains the following subgroups: BSD, Linux, Solaris, and Windows. Within each operating system, attack objects are grouped by services and severity.
Severity	Contains the following subgroups: Critical, Major, Minor, Warning, Info. Within each severity, attack objects are grouped by category. Our severity rating is not based on CVSS (Common Vulnerability Scoring System). We do include data from Bugtraq (Symantec) and CVE (Common Vulnerabilities and Exposures).
Web Services	Contains subgroups based on Web services. Within services, attacked objects are grouped by severity.
Miscellaneous	Contains attack objects that have a significant affect on IDP performance.
Response	Contains attack objects where the attack is detected in the server-to-client direction. This group contains a hierarchy of subgroups that includes all of the above categories.

J-Security Center updates the attack object database to provide new attack objects, to revise severities or recommendations, or to remove obsolete attack objects. We recommend you schedule routine, automatic updates.

## Using Attack Object Groups

A *dynamic group* contains members that match properties you specify for the group. You use dynamic groups so that an attack database update automatically populates the group with relevant members. This eliminates the need to review each new signature to

determine if you need to use it in your existing security policy. A predefined or custom dynamic group can only contain attack objects and not attack groups. Dynamic group members can be either predefined or custom attack objects.

A *static group* is not automatically updated with new members. It contains only the attack objects or groups you have added. Use static groups when you do not want your attack group dynamically populated during NSM updates. For example, if you customize the action for predefined attack objects to meet your company's security policy guidelines, you can create one or more static groups to contain these attack objects. When you perform an NSM attack object update, your static group will not be affected.

There are two types of static groups: predefined static groups and custom static groups. Predefined static groups are categories of groups provided by default.

A custom static group can include the same members as a predefined static group (predefined attack objects, predefined static groups, and predefined dynamic groups), plus the following members:

- Custom attack objects
- Custom dynamic groups
- Other custom static groups

Static groups require more maintenance than dynamic groups because you must manually add or remove attack objects in a static group to manage the members. However, you can include a dynamic group within a static group to automatically update some attack objects. For example, the predefined attack object group Operating System is a static group that contains four predefined static groups: BSD, Linux, Solaris, and Windows. The BSD group contains the predefined dynamic group BSD-Services-Critical, to which attack objects can be added during an attack database update.

## Using Custom Attack Objects

The attack objects provided by the Juniper Networks Security Center (J-Security Center) team cover most cases for small business, enterprise, and service provider networks. Your business might encounter cases where you must modify a predefined attack object or create a new one. For example:

- You read a security advisory about a known attack and want to create an attack object that detects the malicious traffic described in that advisory.
- You need to update or improve an existing third-party signature (such as a Snort signature).
- You want to customize an existing signature or protocol anomaly attack object for your local environments. For example, you might need to customize a signature to prevent false positives generated by a specific application running on your network.
- You want to detect specific activity on your network. For example, you might want to detect abnormal traffic (possibly malicious), remote log-ins, or brute force attacks that attempt to guess usernames and passwords.

For a complete tutorial on creating custom attack objects, see the [IDP Series Custom Attack Objects Reference and Examples Guide](#).

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [J-Security Center Updates Overview on page 19](#)
- [Understanding the IDP Rulebase on page 55](#)
- [IDP Rulebase Example: Using Recommended Attack Objects on page 155](#)
- [Exempt Rulebase Example: Exempting an Attack Object on page 176](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)

## Understanding IDP Rulebase Actions

Actions are responses to sessions that match the source/destination/service condition and the attack object. Actions are what protect your network from attacks.

If a packet triggers multiple rule actions, the IDP Series device takes the most severe action. For example, if the rules dictate that a packet receive a DiffServ marking and be dropped, the IDP Series device will take the more severe action, which is dropping the packet.

Predefined attack objects include a recommended action. The recommended action is generally related to attack severity, but other factors are considered. [Table 22 on page 63](#) lists the recommended actions by attack severity.

**Table 22: Recommended Action by Attack Severity**

Severity	Description	Recommended Action
Critical	Attacks attempt to evade an intrusion prevention system, crash a machine, or gain system-level privileges.	Drop Packet, Drop Connection
Major	Attacks attempt to crash a service, perform a denial of service, install or use a Trojan, or gain user-level access to a host.	Drop Packet, Drop Connection
Minor	Attacks attempt to obtain critical information through directory traversal or information leaks.	None
Warning	Attacks are obsolete or attempt to obtain noncritical information or scan the network.	None
Info	Attacks are normal, harmless traffic containing URLs, DNS lookup failures, and SNMP public community strings. You can use informational attack objects to obtain information about your network.	None

If you choose, you can set a different action. [Table 23 on page 64](#) describes the actions you can set for IDP rulebase rules.

Table 23: IDP Rulebase Actions

Action	Description
None	Inspects for attacks but takes no action against the connection if an attack is found.
Ignore	Does not take action and ignores the remainder of the session.
Diffserv Marking	<p>Assigns the indicated service-differentiation value to the packet, and then passes it on normally. Set the service-differentiation value in the dialog box that appears when you select this action in the rulebase.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. Therefore, this action has no effect in sniffer mode.</p>
Drop Packet	<p>Drops a matching packet before it can reach its destination but does not close the connection. Use this action in rules focused on traffic that is prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a DoS that prevents you from receiving traffic from a legitimate source address.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. Therefore, this action has no effect in sniffer mode.</p>
Drop Connection	<p>Drops the connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. Therefore, this action has no effect in sniffer mode.</p>
Close Client and Server	<p>Closes the connection and sends an RST packet to both the client and the server.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and server but does not close the connection.</p>
Close Client	<p>Closes the connection to the client but not to the server.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and server but does not close the connection.</p> <p><b>NOTE:</b> In VLAN tagged MPLS traffic, the Close Client action drops the connection instead of closing it.</p>
Close Server	<p>Closes the connection to the server but not to the client.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and server but does not close the connection.</p>

If the IDP engine matches an attack, it can take action not only against the current session but also against subsequent network traffic from the same IP address. Such actions are called *IP actions*. By default, the specified IP action is permanent (timeout = 0). If you prefer, you can set a timeout.

[Table 24 on page 65](#) describes IDP rulebase IP actions.

Table 24: IDP Rulebase IP Actions

IP Action	Description
IP Block	<p>Blocks the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP address</li> <li>• Source subnet</li> <li>• Protocol</li> <li>• Destination IP address</li> <li>• Destination subnet</li> <li>• Destination port</li> <li>• From zone</li> </ul>
IP Close	<p>Closes the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP address</li> <li>• Source subnet</li> <li>• Protocol</li> <li>• Destination IP address</li> <li>• Destination subnet</li> <li>• Destination port</li> <li>• From zone</li> </ul>
IP Notify	Does not take any action against future traffic but logs the event or sends an alert.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the IDP Rulebase on page 55](#)
- [IDP Rulebase Example: Using Recommended Actions on page 156](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Specifying Rule Session Action \(NSM Procedure\) on page 218](#)

## Understanding IDP Rulebase Notification Options

You use notification features to help you manage your network, analyze your network security, validate your security policy, and capture forensic evidence of attacks. You can set notification options per rule.

The first time you design a security policy, you might be tempted to log all data for all attacks and let the policy run indefinitely. We recommend you take a more refined approach. Some attack objects are informational only, and others can generate false positives and redundant logs. If you become overloaded with data, you can miss something important. Remember that security policies that generate too many log records are hazardous to the security of your network, as you might discover an attack too late or miss a security breach entirely as a result of having to sift through hundreds of log records.

Excessive logging can also affect throughput, performance, and available disk space. A good security policy generates enough logs to fully document only the important security events on your network.

By default, logging is enabled for IDP rulebase rules. [Table 25 on page 66](#) describes the notification options you can configure. You also have the option to disable logging.

**Table 25: IDP Rulebase Notification Options**

Option	Description
Event logs and alerts	<p>You can enable the following delivery and handling options for logs:</p> <ul style="list-style-type: none"> <li>• Send to NSM log viewer.</li> <li>• Send to NSM log viewer and flag as an alert.</li> <li>• Send to an e-mail address list.</li> <li>• Send to syslog.</li> <li>• Send to SNMP trap.</li> <li>• Save in XML format.</li> <li>• Save in CVS format.</li> <li>• Process with a script.</li> </ul>
Packet captures	<p>Viewing the packets used in an attack on your network can help you determine the extent of the attempted attack and its purpose, whether or not the attack was successful, and any possible damage to your network.</p> <p>If multiple rules with packet capture enabled match the same attack, the IDP system captures the maximum specified number of packets. For example, you configure rule 1 to capture 10 packets before and after the attack, and you configure rule 2 to capture 5 packets before and after the attack. If both rules match the same attack, the IDP system attempts to capture 10 packets before and after the attack.</p> <p>You can capture up to 256 packets before the event and 256 packets after the event.</p> <p><b>NOTE:</b> If necessary, you can improve performance by logging only the packets received after the attack.</p>

For complete procedures on setting IDP rulebase notification options, see the *IDP Series Administration Guide*.

**Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the IDP Rulebase on page 55](#)
- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## CHAPTER 6

# The Exempt Rulebase

- [Understanding the Exempt Rulebase on page 67](#)

### Understanding the Exempt Rulebase

---

The Exempt rulebase enhances manageability of the IDP solution by enabling you to categorically exempt traffic segments you know to be safe from processing by the IDP rulebase.

A *false positive*, also known as a false alert, is a situation in which benign traffic causes an intrusion detection system (IDS) to generate an alert. Too many false positives can degrade performance and produce oversized log files.

The IDP engine reduces false positives by using stateful signatures to detect known attacks. A stateful signature knows the pattern and location of the attack, and produces fewer false positives than regular attack signatures because it does not inspect network traffic that cannot contain the attack.

To further increase detection accuracy and reduce false positives, the IDP engine uses:

- Flow tracking to correlate multiple TCP/UDP connections into a single flow to determine the validity of the traffic.
- IP defragmentation and TCP reassembly to reconstruct fragmented traffic.
- Protocol normalization to normalize traffic to a common format for analysis.

Still, a few false positives from your IDS are normal, especially when you are testing new security policies. You can use the Exempt rulebase to manage these cases.

When you create rules for the Exempt rulebase, you specify:

- A source/destination/service match condition
- At least one attack object



---

**NOTE:** The Exempt rulebase is a non-terminal rulebase. That is, the IDP process engine processes all rules in the rulebase.

---

**Related  
Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*.

- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Understanding the IDP Rulebase on page 55](#)
- [Exempt Rulebase Example: Exempting a Source Destination Pair on page 175](#)
- [Exempt Rulebase Example: Exempting an Attack Object on page 176](#)

The following related topics are included in the *IDP Series Administration Guide*.

- [Configuring Exempt Rulebase Rules \(NSM Procedure\) on page 227](#)



## CHAPTER 7

# The APE Rulebase

- [Understanding the APE Rulebase on page 69](#)
- [Understanding APE Rulebase Match Conditions on page 70](#)
- [Using Application Objects on page 73](#)
- [Understanding APE Rulebase Actions on page 80](#)
- [Understanding APE Rulebase Notification Options on page 82](#)

### Understanding the APE Rulebase

---

The APE rulebase (application policy enforcement) leverages the application identification feature to enable you to manage network traffic based on application. APE rules match source-destination-application criteria. APE rules do not use attack objects.

You can configure rule actions to meet application policy enforcement objectives. For example:

- To use the IDP Series device like an application firewall, you can specify drop or close actions. Matching traffic is terminated at the IDP Series device.
- To set a cap on available bandwidth for disfavored applications or use of certain applications by certain users, you can specify a rate limiting action. When the limit is reached, the IDP Series device begins dropping matching traffic.
- To support deployments where you use other network equipment to implement quality-of-service (QoS) guarantees, you can specify a DiffServ marker action. If a session matches a rule, the IDP engine applies the DSCP marker to the session packets before transmitting them.

Any traffic not terminated by APE rules can be inspected subsequently by the IDP rulebase and other rulebases.

When you create rules for the APE rulebase, you specify:

- Match conditions
- An action
- Notification options

**Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding APE Rulebase Match Conditions on page 70](#)
- [Using Application Objects on page 73](#)
- [Understanding APE Rulebase Actions on page 80](#)
- [Understanding APE Rulebase Notification Options on page 82](#)
- [APE Rulebase Example: Aggregate and Per-Subscriber Rate Limits on page 170](#)
- [APE Rulebase Example: Using Extended Application Objects on page 165](#)
- [APE Rulebase Example: Matching Custom Application Objects on page 171](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)

---

## Understanding APE Rulebase Match Conditions

The APE rulebase is a terminal rulebase. Rules are evaluated in numerical order. The first rule to match is applied, and subsequent rules are not processed.

If an APE rule matches but the action does not drop the connection, the IDP system also processes additional rulebases to inspect for attacks. If an attack rule identifies the connection to be closed or dropped, that action is taken and the rate-limiting action is not required.

The matching tuple for APE rules includes the following elements:

- Source or user role
- Destination
- Service or the combined list of applications and extended applications
- VLAN tag

The Boolean logic of the matching tuple is as follows:

(src OR user role) AND destination AND vlan AND (service OR application list)



**NOTE:** You can use the Any wildcard to “remove” a property from the tuple. For example, if you specify Any for source, destination, or VLAN tag, you are creating a “traffic lane” that treats all traffic matching the specified application the same. However, Any has a different significance when building the service or application list. When setting service or application guidelines, be sure to follow the guidelines below.

---

[Table 26 on page 71](#) provides guidelines for setting IDP rulebase match conditions.

Table 26: APE Rulebase Match Condition Guidelines

Setting	Guideline
From zone/To zone	Not applicable to IDP Series devices.
Source	<p>Requires one of the specified source IP addresses to match the session in order for the rule to be applied. You can add address objects for hosts, groups, or network address ranges.</p> <p>A rule can specify matching criteria for Source IP or user role, but not both. A policy can include rules that match on Source IP and rules that match on user role.</p> <p><b>NOTE:</b> If a value for user role matches, the source parameter is not used.</p>
User Role	<p>Requires one of the specified user roles to match the session in order for the rule to be applied.</p> <p>A rule can specify matching criteria for Source IP or user role, but not both. In a rulebase, the user role-based rules are evaluated before the IP address-based rules. If a user-role based rule matches, the rule is applied and the IP address-based rules are not consulted.</p> <p>Matching based on user role depends on integration with a Juniper Networks IC Series UAC device.</p>
Destination	Requires one of the specified destination IP addresses to match the session for the rule to be applied. You can add address objects for hosts, groups, or network address ranges.
Service	<p>Requires a match of one of the specified services.</p> <p>A single rule can match a service object definition or an application list, but not both. We recommend you create rules that match an application list whenever possible. Matching based on application uses the application identification feature, which can identify the application regardless of port. We support rules that match service object definitions for cases where there is not a suitable application object.</p> <p>If your rule includes application or extended application objects, specify <b>Default</b> for the service parameter.</p> <p>If you do not want to match on service or application list, specify <b>Any</b> for all three (service, application, and extended application).</p> <p>If there are no suitable application objects, create a rule that uses the service object and set the application and extended application columns to <b>Any</b>.</p> <p>If the service uses standard ports, you can select from predefined services. If the service uses nonstandard ports, you can create a custom service object. The IDP engine can inspect services that use TCP, UDP, RPC, and ICMP transport layer protocols.</p>

Table 26: APE Rulebase Match Condition Guidelines (*continued*)

Setting	Guideline
Application	<p>Requires one of the specified applications to match the session for the rule to be applied.</p> <p>You use the Application and Extended Application columns to build a list of applications to match the rule. You can specify individual applications or application groups. When you add a group, you are in effect adding its members to the list. The group object itself is not evaluated. The list is evaluated as a Boolean OR, so if one of the application or extended application objects specified in the rule is identified, the “service or application” component of the tuple matches. If any application or member of a group matches, the rule matches.</p> <p>The predefined list of applications is populated by the application signatures included in J-Security Center signature updates. The application identification feature uses both heuristic methods and signature pattern matching to identify the application regardless of port. Port-independent application identification simplifies rule configuration and ensures that you do not miss applications that are running on nonstandard ports. For this reason, we recommend that you use the application parameter instead of the service parameter whenever possible.</p> <p>Specify <b>Any</b> in the Application column when creating a service-based rule or when creating an application-based rule where the application list consists only of extended application objects.</p> <p><b>NOTE:</b> Extended application matching is more granular than application matching. Do not select HTTP in the application column if you also plan to specify extended application objects in the same rule. If you specify HTTP and HTTP:Facebook, for example, the rule matches HTTP or HTTP:Facebook. The result is indistinguishable from a rule matching only HTTP.</p>
Extended Application	<p>Requires one of the specified <i>extended applications</i> to match the session for the rule to be applied. Extended applications are also called <i>nested applications</i>. The Juniper Networks Security Center (J-Security Center) provides predefined application signatures for many Web 2.0 applications running over HTTP. Matching on these signatures depends on the application identification feature, which is enabled by default.</p> <p>You use the Application and Extended Application columns to build a list of applications to match the rule. The list is evaluated as a Boolean OR, so if one of the application or extended application objects specified in the rule is identified, the “service or application” component of the tuple matches.</p> <p>Specify <b>Any</b> in the Extended Application column when you are creating a service-based rule or when you are creating an application-based rule where the application list consists only of application objects.</p>
VLAN	<p>Requires one of the specified VLAN IDs to match the session for the rule to be applied.</p> <p>Specifying <b>Any</b> effectively removes VLAN ID from the tuple.</p>



**TIP:** You can use Profiler to identify the destination servers and services that are included in your network. In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM online Help.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Rule-Matching Algorithm on page 45](#)

- [Understanding the APE Rulebase on page 69](#)
- [Using Application Identification on page 43](#)
- [Using Application Objects on page 73](#)
- [User-Role-Based Policy Feature Overview on page 58](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)

## Using Application Objects

---

You specify application objects in APE rules as a key element in the matching tuple. This topic provides an overview of application objects and includes the following sections:

- [Application Objects Overview on page 73](#)
- [Understanding Predefined Application Objects on page 73](#)
- [Using Application Groups on page 78](#)
- [Using Custom Application Objects on page 79](#)

### Application Objects Overview

Application objects are also called *application signatures*. The signature comprises the Layer 7 protocol, protocol contexts, and a DFA pattern found in client-to-server and server-to-client traffic flows. An application object adds program logic to signatures, such as the capability of chaining signatures to create an ordered or unordered compound expression, a maximum number of transactions wherein the signature must occur to be a match, an order value that sets match precedence in cases where multiple signatures are identified, and a unique ID that the system uses both for logical processing and reporting. Extended application objects, also called nested applications, identify Web 2.0 applications running over HTTP.

Application objects are stored in the NSM database application signature table (also referred to as the appsig table) and extended application signature table (also referred to as the extappsig table). Juniper Networks Security Center (J-Security Center) makes predefined application objects and predefined extended application objects available for download to NSM during signature database updates. You use the NSM Object Manager to manage application objects. You specify application objects in APE rules as a key element in the matching tuple. You push the application signatures from NSM to your devices when you push policy updates.

### Understanding Predefined Application Objects

J-Security Center makes predefined application objects and predefined extended application objects available for download to NSM during signature database updates. A complete list of application objects is maintained on the J-Security Center [website](#). We recommend that you become familiar with these objects and leverage them in your APE rules as much as possible.

Figure 12 on page 74 shows the NSM Object Manager Predefined Application Objects tab. This view displays the following properties for predefined application objects:

- Name—A unique, descriptive name.
- Application category—A classification used for sorting the list. Not unique.
- Port range—A range of ports on which the application might run. The application is identified only if the server port is within the specified range.
- Application type—A unique identifier used by the application identification feature.
- Port binding—The standard ports known to be used by the application.
- Match order—In case traffic matches protocol, port, and pattern for two or more applications, the match order determines which object is considered the match (the object with the lower match order number is considered the match).

Figure 12: NSM Object Manager: Predefined Application Objects

Application Objects					
Predefined Application Objects	Custom Application Objects	Predefined Extended Application Objects	Application Group Objects		
Name	Application Category	Port Ranges	Application Type	Port Binding	Match Order
HPOVTRACE	ENTERPRISE-INFRASTRUCTURE	TCP:5051-5053	HPOVTRACE	TCP:5051-5053	146
HSS-SSL-TCP	MISC	TCP:0-65535	HSS-SSL-TCP	...	32
HSS-SSL-UDP	MISC	UDP:0-65535	HSS-SSL-UDP	...	150
HTTP	WEB	TCP:0-65535	HTTP	TCP:80,3128,80...	99
ICA-TCP	REMOTE-ACCESS	TCP:0-65535	ICA-TCP	TCP:1494	5
ICA-UDP	REMOTE-ACCESS	UDP:0-65535	ICA	UDP:1604	51
ICQP	SCADA	TCP:102	ICQP	TCP:102	147
ICQ	PEER-TO-PEER-CHAT	TCP:0-65535	ICQ	...	22
IDENT	MISC	TCP:113	IDENT	TCP:113	68
IEC104	SCADA	TCP:2404	IEC104	TCP:2404	155
IMAP	MESSAGING	TCP:0-65535	IMAP	TCP:143	115
IPSEC-IKE-MAIN-AGGR...	ENCRYPTION	UDP:0-65535	IKE	UDP:500	25
IRC	PEER-TO-PEER-CHAT	TCP:0-65535	IRC	TCP:6666,6667...	46
JABBER	PEER-TO-PEER-CHAT	TCP:0-65535	JABBER	TCP:5222	114
JAVA-RMI	REMOTE-COMMAND	TCP:1099	JAVA-RMI	TCP:1099	188
JONDO-PROXY	WEB	TCP:0-65535	JONDO-PROXY	...	140
KADEMLIA-KAD	PEER-TO-PEER-FILE-SHARING	UDP:0-65535	KADEMLIA-KAD	...	83
KADEMLIA-OVERNET	PEER-TO-PEER-FILE-SHARING	TCP:0-65535 UDP:0-65535	KADEMLIA-OVERNET	...	79
KAZAA	PEER-TO-PEER-FILE-SHARING	TCP:0-65535 UDP:0-65535	KAZAA	...	119
KRB4	ENCRYPTION	UDP:0-65535	KRB4	...	113
KRB5	ENCRYPTION	TCP:0-65535 UDP:0-65535	KRB5	TCP:543 UDP:58	27
KUOOO	PEER-TO-PEER-FILE-SHARING	UDP:0-65535	KUOOO	UDP:7000	78
LDAP	ENTERPRISE-INFRASTRUCTURE	TCP:0-65535	LDAP	TCP:389	111
LOTUSNOTES	MESSAGING	TCP:0-65535	LOTUS-NOTES	TCP:1352	134

You can double-click the table entry to view additional details, including the signature pattern regular expression to match in client-to-server and server-to-client directions. Figure 13 on page 75 shows the general properties of the predefined application object for HTTP.

Figure 13: NSM Object Manager: Predefined Application: General Tab

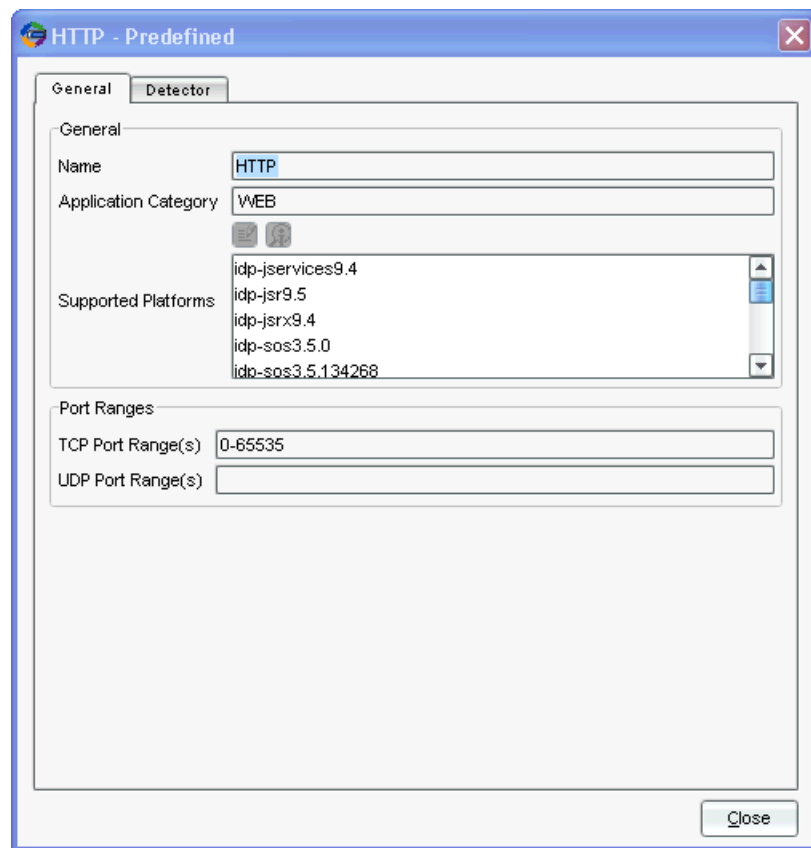


Figure 14 on page 76 shows the signature properties of the predefined application object for HTTP.

Figure 14: NSM Object Manager: Predefined Application: Detector Tab

**HTTP - Predefined**

**General** **Detector**

**Port Binding**

Application Type: HTTP

TCP Port Binding: 80,3128,8000,8080

UDP Port Binding:

**Signature**

**Client-to-server**

DFA Pattern: (\\OPTIONS|HEAD|GET|POST|PUT|B?DELETE|TRACE|SEARCH|B?PROPFIND|PROPPATCH|MKCOL|B?COPY|B?MOVE|LOCK|UNLOCK|CHECKOUT|

PCRE Pattern:

**Server-to-client**

DFA Pattern: (. \*HTTP/1\\.([01])s\\.?.?w<\\DOCTYPE\\w\\.?.?w<\\HTML\\w\\.?.?w<\\?xml\\w\\[Content-type\\: ].\*)

PCRE Pattern:

Minimum data length: 20

Signature Match Order: 122

Close

You can use extended application objects in APE rules if you want to treat various Web 2.0 applications running over HTTP differently. [Figure 15 on page 77](#) shows the NSM Object Manager Predefined Extended Application Objects tab. This view displays the following properties for predefined extended application objects:

- Name—A unique, descriptive name.
- Application category—A classification used for sorting the list. Not unique.
- Extended application ID—A unique identifier. The system uses the unique ID for both logical processing and reporting.
- Application type—A unique identifier used by the application identification feature.
- L7 protocol—Only HTTP is supported.
- Chain order—Indicates whether or not the member signatures are ordered.



Figure 15: NSM Object Manager: Predefined Extended Application Objects

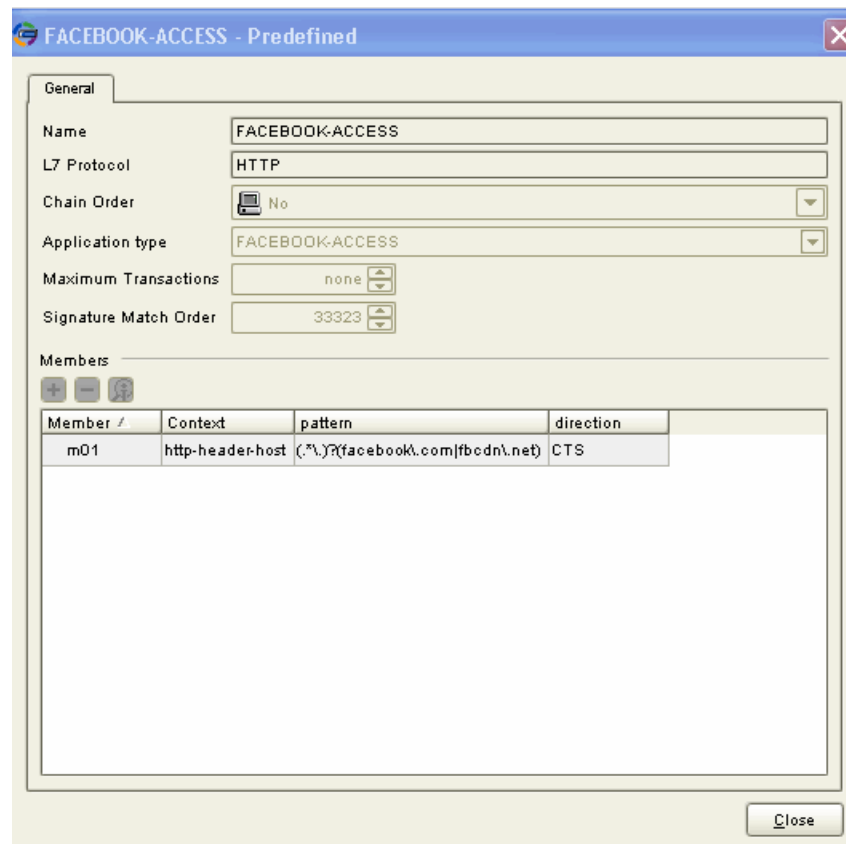
**Application Objects**

Predefined Application Objects   Custom Application Objects   **Predefined Extended Application Objects**   Application Group Objects

Name	Application Category	Ext ID	Application Type	L7 Protocol	Chain Order
MYSPACE	SOCIAL-NETWORKING	316	MYSPACE	HTTP	No
TWITTER	SOCIAL-NETWORKING	317	TWITTER	HTTP	No
BEBO	SOCIAL-NETWORKING	321	BEBO	HTTP	No
CLASSMATES	SOCIAL-NETWORKING	322	CLASSMATES	HTTP	No
Hi5	SOCIAL-NETWORKING	329	Hi5	HTTP	No
DOOF	SOCIAL-NETWORKING	290	DOOF	HTTP	No
BLOGGER-POST	SOCIAL-NETWORKING	343	BLOGGER-POST	HTTP	No
MYSPACE-MAIL	SOCIAL-NETWORKING	351	MYSPACE-MAIL	HTTP	No
MYSPACE-CHAT	SOCIAL-NETWORKING	352	MYSPACE-CHAT	HTTP	No
MYSPACE-VIDEO	SOCIAL-NETWORKING	360	MYSPACE-VIDEO	HTTP	No
VKONTAKTE	SOCIAL-NETWORKING	501	VKONTAKTE	HTTP	No
MIKI	SOCIAL-NETWORKING	444	MIKI	HTTP	No
TIANYA	SOCIAL-NETWORKING	445	TIANYA	HTTP	No
KAIXIN001	SOCIAL-NETWORKING	447	KAIXIN001	HTTP	No
ODNOKLASSNIKI	SOCIAL-NETWORKING	448	ODNOKLASSNIKI	HTTP	No
RENREN	SOCIAL-NETWORKING	449	RENREN	HTTP	No
ADULTFRIENDFINDER	SOCIAL-NETWORKING	480	ADULTFRIENDFINDER	HTTP	No
TARINGA	SOCIAL-NETWORKING	481	TARINGA	HTTP	No
BADOO	SOCIAL-NETWORKING	483	BADOO	HTTP	No
NING	SOCIAL-NETWORKING	484	NING	HTTP	No
NETLOG	SOCIAL-NETWORKING	492	NETLOG	HTTP	No
HYVESDOTNL	SOCIAL-NETWORKING	506	HYVESDOTNL	HTTP	No
PLENTYOFFISH	SOCIAL-NETWORKING	508	PLENTYOFFISH	HTTP	No
NATEON	SOCIAL-NETWORKING	403	NATEON	HTTP	No
BLOGSPOT-POST	SOCIAL-NETWORKING	413	BLOGSPOT-POST	HTTP	No
PING-FM	SOCIAL-NETWORKING	509	PING-FM	HTTP	No

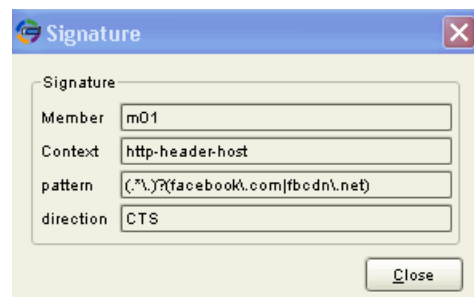
You can double-click the table entry to view additional details, including the matching HTTP context, signature pattern, and client-to-server or server-to-client direction. [Figure 16 on page 78](#) shows the properties of the HTTP:Facebook-Access application object.

Figure 16: NSM Object Manager: Extended Application Details



An application signature can include one member or more members in a compound signature. Double-click the table row entry for the member to display its details. [Figure 17 on page 78](#) shows the properties of the HTTP:Facebook-Access application object.

Figure 17: NSM Object Manager: Extended Application Member Details

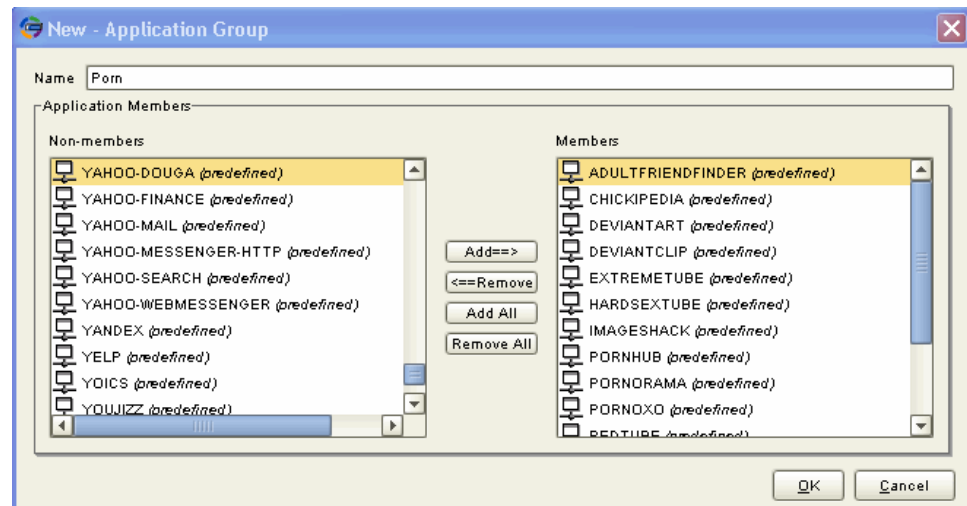


## Using Application Groups

Application groups are administrative objects you can use to simplify rule configuration. A group comprises application objects that you want to treat the same—that is, you want to apply the same action to matching traffic. [Figure 18 on page 79](#) shows the Application Group dialog box. In the Non-Members box, applications are listed first, followed by extended applications. You can nest a group within another group. In the Application

Group dialog box, the icons next to group object names and application object names differ so you can distinguish the two when you browse the lists.

Figure 18: NSM Object Manager: Application Group Dialog Box



## Using Custom Application Objects

You can create rules for most business cases with the predefined application objects provided by J-Security Center. In some cases, you might need to manage traffic for applications not yet supported by J-Security Center. First, check with your Juniper Networks representative to see if a predefined application object is forthcoming. If support for your application object is not forthcoming, you can use the NSM Object Manager to define a custom application object. You can then specify that object as a match for APE rules.

[Figure 19 on page 80](#) shows the Custom Application dialog box.

Figure 19: NSM Object Manager: Custom Application Dialog Box

The screenshot shows a dialog box titled "Aspera FASP - Custom". It has two tabs: "General" and "Detector". The "General" tab is selected. Inside the "General" tab, there are three main sections: "Name" with the value "Aspera FASP", "Application Category" with the value "File-Server", and "Supported Platforms" with the value "idp5.1.0". Below these, there are two more sections: "TCP Port Range(s)" with the value "22,33001" and "UDP Port Range(s)" with the value "0-65535". At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [J-Security Center Updates Overview on page 19](#)
- [Understanding the APE Rulebase on page 69](#)
- [APE Rulebase Example: Using Extended Application Objects on page 165](#)
- [APE Rulebase Example: Matching Custom Application Objects on page 171](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

## Understanding APE Rulebase Actions

Actions are responses to sessions that match the source/destination/service or source/destination/application condition.

[Table 27 on page 81](#) describes the actions you can specify for application policy enforcement (APE) rulebase rules.

Table 27: IDP Rulebase Actions

Action	Description
None	Does not perform rate limiting. Logs generated for traffic that match this rule display <b>Accepted</b> .
Drop Connection	<p>Drops the connection without sending an RST packet to the sender, thereby preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.</p> <p>Logs generated for traffic that match this rule display <b>Drop Connection</b>.</p> <p><b>NOTE:</b> In sniffer mode, this action has no effect because the IDP Series device is not in the path of network traffic.</p>
Close Client	<p>Closes the connection to the client but not to the server.</p> <p>Logs generated for traffic that match this rule display <b>Close Client</b>.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and the server, but it does not close the connection.</p>
Close Server	<p>Closes the connection to the server but not to the client.</p> <p>Logs generated for traffic that match this rule display <b>Close Server</b>.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and the server, but it does not close the connection.</p>
Close Client and Server	<p>Closes the connection and sends an RST packet to both the client and the server.</p> <p>Logs generated for traffic that match this rule display <b>Close</b>.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and the server, but it does not close the connection.</p>
DiffServ Marking	<p>Assigns the DiffServ value you specify to the packet. This action is useful when your network has a class of service (CoS) design, and you want to use the IDP Series device to rewrite the CoS code point based on APE rules processing. The CoS rules you have implemented for the next devices in the network path ultimately determine the effect on the transmission rate.</p> <p>Logs generated for traffic that match this rule display <b>DiffServ</b>.</p> <p><b>NOTE:</b> In sniffer mode, this action has no effect because the IDP Series device is not in the path of network traffic.</p>

Table 27: IDP Rulebase Actions (*continued*)

Action	Description
Rate Limit	<p>Rate limits set an aggregate limit for all matching sessions. If a session matches an APE rule in which a rate limit has been set, the IDP engine performs a rate-limit check. If the limit is not reached, the IDP Series device forwards the packets. If the limit is reached, the IDP Series device behaves as if no bandwidth is available: it drops packets until the aggregate bandwidth falls below the limit. When the IDP Series device drops packets, the TCP or UDP endpoints identify the packet loss and slow the transmission rate.</p> <p>The rate limits that are best suited for your business case depend on the bandwidth for your links. If you have a 1-Gbps link and want no more than 10% available to peer-to-peer traffic, the sum of the rate limits you specify for all peer-to-peer rules must be less than 102.4 Mbps (in each direction).</p> <p>If you implement user-role-based rules, you can apply rate limiting to all users who belong to the specified role or to individual users who belong to the specified role. By default, rate limiting is applied to all users who belong to the specified role. In this case, you would configure a larger limit. You can change this setting with the command-line interface. If you change the default to enable rate limiting per user, configure a smaller limit.</p> <p>You configure separate rate limits for client-to-server and server-to-client directions. For peer-to-peer traffic, we recommend that you set the same rate for each direction.</p> <p><b>NOTE:</b> For TFTP traffic, all traffic is considered client-to-server traffic. A TFTP server responds to get requests by establishing an ephemeral port from which to send the reply. In this case, both directions appear to the IDP Series device as client-to-server flows. We recommend you set the same rate for each direction.</p> <p>Logs generated for traffic that match this rule display <b>Rate Limit</b> and traffic direction (c2s or s2c).</p> <p><b>NOTE:</b> In sniffer mode, this action has no effect because the IDP Series device is not in the path of network traffic.</p>
DiffServ Marking & Rate Limiting	Takes both actions described above.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the APE Rulebase on page 69](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Enabling Per-User Rate Limiting for User-Role-Based Rules on page 321](#)

## Understanding APE Rulebase Notification Options

Notification options determine whether the IDP Series device generates logs and alerts when a session matches a rule. When enabled, the IDP Series device generates a log that the client-to-server or server-to-client rate limit was reached. Logging is enabled by default. [Table 28 on page 83](#) describes the notification options.

Table 28: APE Rulebase Notification Options

Option	Description
Event logs and alerts	<p>You can enable the following delivery and handling options for logs:</p> <ul style="list-style-type: none"><li>• Send to NSM log viewer.</li><li>• Send to NSM log viewer and flag as an alert.</li><li>• Send to an e-mail address list.</li><li>• Send to syslog.</li><li>• Send to SNMP trap.</li><li>• Save in XML format.</li><li>• Save in CVS format.</li><li>• Process with a script.</li></ul> <p>You also have the option to disable logging.</p>
<b>Related Documentation</b>	<p>The following related topic is included in the <i>IDP Series Concepts and Examples Guide</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Understanding the APE Rulebase on page 69</a></li></ul> <p>The following related topic is included in the <i>IDP Series Administration Guide</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Configuring the APE Rulebase (NSM Procedure) on page 228</a></li></ul>





## CHAPTER 8

# The Backdoor Rulebase

- [Understanding the Backdoor Rulebase on page 85](#)
- [Understanding Backdoor Rulebase Match Settings on page 87](#)
- [Understanding the Backdoor Rulebase Operation Setting on page 88](#)
- [Understanding Backdoor Rulebase Actions on page 88](#)
- [Understanding Backdoor Rulebase Notification Options on page 89](#)

### Understanding the Backdoor Rulebase

---

The Backdoor rulebase detects the kind of interactive traffic produced during backdoor attacks.

A *backdoor* is a mechanism installed on a host computer that facilitates unauthorized access to the system. Attackers who have already compromised a system can install a backdoor to make future attacks easier. When attackers type commands to control a backdoor, they generate interactive traffic.

Unlike antivirus software, which scans for known backdoor files or executable files on the host system, the IDP engine detects the interactive traffic that is produced when backdoors are used. Interactive programs often transmit several short IP packets containing individual keystrokes and their echoes, reflecting the real-time actions of a user (or an attacker).

When detection is enabled, the IDP engine detects traffic that exceeds the interactive traffic thresholds you set as runtime parameters. [Figure 20 on page 86](#) shows the backdoor detection settings in the NSM Device Manager configuration editor.

Figure 20: NSM Device Manager: Sensor Settings &gt; Run-Time Parameters

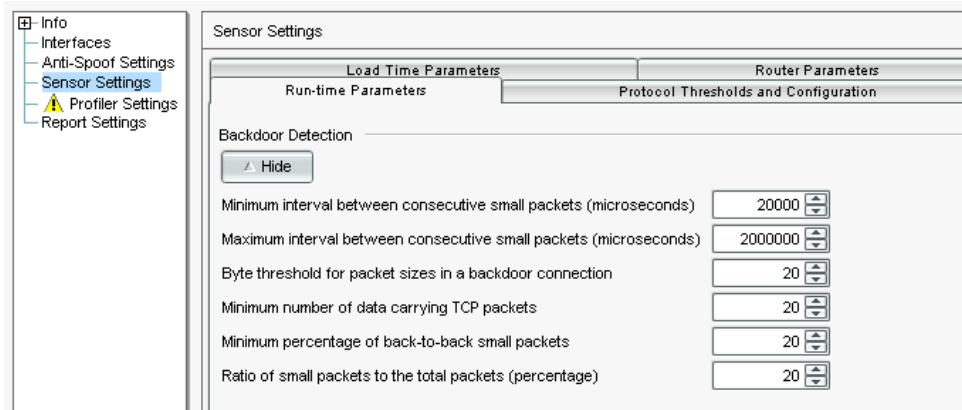


Table 29 on page 86 shows the defaults for backdoor detection runtime parameters. You can tune these parameters if safe traffic in your network triggers false positives.

Table 29: Backdoor Detection Runtime Parameters

Parameter	Default
Minimum interval between consecutive small packets (microseconds)	20,000
Maximum interval between consecutive small packets (microseconds)	2,000,000
Byte threshold for packet sizes in a backdoor connection (bytes)	20
Minimum number of data carrying TCP packets (number)	20
Minimum percentage of back-to-back small packets (percentage)	20
Ratio of small packets to the total packets (percentage)	20

Detecting the signs of interactive traffic ensures that the IDP Series device can detect all backdoors, both known and unknown. If the IDP Series device detects interactive traffic, it can perform actions against the connection to prevent the attacker from further compromising your network.

When you create rules for the Backdoor rulebase, you specify:

- A source/destination/service match condition
- Operation
- Action
- Notification options

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)

- [Understanding Backdoor Rulebase Match Settings on page 87](#)
- [Understanding the Backdoor Rulebase Operation Setting on page 88](#)
- [Understanding Backdoor Rulebase Actions on page 88](#)
- [Understanding Backdoor Rulebase Notification Options on page 89](#)
- [Backdoor Rulebase Example: netcat on page 176](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)
- [Modifying the IDP Series Device Configuration on page 351](#)

## Understanding Backdoor Rulebase Match Settings

Backdoor rulebase rules are triggered when source, destination, and service for the traffic match the rule.

To detect incoming interactive traffic, set the source to **Any** and the destination to the IP address of network device you want to protect.

To detect outgoing interactive traffic, set the source to the IP address of the network device you want to protect and the destination to **Any**.

Specify not only the services on the network device you want to protect but also interactive services that can be installed and used by attackers.



.....  
**NOTE:** Including Telnet, SSH, RSH, NetMeeting, or VNC as services can result in false positives because these services are used to legitimately control a remote system. We recommend that you not include these services in your service list.  
.....



.....  
**TIP:** You can use two rules to protect a large number of servers. Configure rule 1 to match services you do not want to detect and set Operation to Ignore. Configure rule 2 to match any traffic and set Operation to Detect.  
.....



.....  
**TIP:** In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM documentation.  
.....



**NOTE:** The Backdoor rulebase is a terminal rulebase—that is, Backdoor rules are inherently terminal rules. If a Backdoor rule matches, the IDP engine does not process subsequent rules.

**Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Understanding the Backdoor Rulebase on page 85](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)

---

## Understanding the Backdoor Rulebase Operation Setting

The Backdoor rulebase operation setting is a toggle between Ignore and Detect.

Use **Ignore** to whitelist services for accepted forms of interactive traffic, such as Telnet, SSH, RSH, NetMeeting, or VNC.

Use **Detect** for all other interactive traffic.

List the ignore rule first, followed by the detect rule.



**TIP:** You can use two rules to protect a large number of servers. Configure rule 1 to match services you do not want to detect and set Operation to Ignore. Configure rule 2 to match any traffic and set Operation to Detect.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Backdoor Rulebase on page 85](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)

---

## Understanding Backdoor Rulebase Actions

By default, Backdoor rulebase rules accept and log traffic that matches the rule. If you choose, you can set a different action. [Table 30 on page 89](#) describes the actions you can set for Backdoor rulebase rules.

Table 30: Backdoor Rulebase Actions

Action	Description
Accept	Accepts the interactive traffic.
Drop Connection	Drops the interactive connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
Close Client and Server	<p>Closes the connection and sends an RST packet to both the client and the server.</p> <p>Logs generated for traffic that match this rule display <b>Close</b>.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the IDP Series device can send an RST packet to both the client and server but does not close the connection.</p>
Close Client	Closes the interactive connection to the client but not to the server.
Close Server	Closes the interactive connection to the server but not to the client.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Backdoor Rulebase on page 85](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)

## Understanding Backdoor Rulebase Notification Options

By default, logging is enabled for Backdoor rulebase rules. [Table 31 on page 89](#) describes the notification options you can configure. You also have the option to disable logging.

Table 31: Backdoor Rulebase Notification Options

Option	Description
Event logs and alerts	<p>You can enable the following delivery and handling options for logs:</p> <ul style="list-style-type: none"> <li>• Send to NSM log viewer.</li> <li>• Send to NSM log viewer and flag as an alert.</li> <li>• Send to an e-mail address list.</li> <li>• Send to syslog.</li> <li>• Send to SNMP trap.</li> <li>• Save in XML format.</li> <li>• Save in CVS format.</li> <li>• Process with a script.</li> </ul>

Table 31: Backdoor Rulebase Notification Options (*continued*)

Option	Description
Packet captures	<p>Viewing the packets used in an attack on your network can help you determine the extent of the attempted attack and its purpose, whether or not the attack was successful, and any possible damage to your network.</p> <p>If multiple rules with packet capture enabled match the same attack, the IDP Series device captures the maximum specified number of packets. For example, you configure rule 1 to capture 10 packets before and after the attack, and you configure rule 2 to capture 5 packets before and after the attack. If both rules match the same attack, the IDP Series device attempts to capture 10 packets before and after the attack.</p> <p>You can capture up to 256 packets before the event and 256 packets after the event.</p> <p><b>NOTE:</b> If necessary, you can improve performance by logging only the packets received after the attack.</p>



**NOTE:** Backdoor rulebase notification options are the same as IDP rulebase options.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Backdoor Rulebase on page 85](#)
- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)

## CHAPTER 9

# The SYN Protector Rulebase

- [Understanding the SYN Protector Rulebase on page 91](#)
- [Understanding SYN Protector Rulebase Match Settings on page 93](#)
- [Understanding SYN Protector Rulebase Modes on page 94](#)
- [Understanding SYN Protector Rulebase Notification Options on page 95](#)

### Understanding the SYN Protector Rulebase

---

The SYN Protector rulebase protects your network from malicious SYN flood attacks.

A *SYN flood attack* is a type of denial-of-service (DoS) attack, where the attacker attempts to flood your server with TCP requests to overwhelm your resources.

Attackers send a SYN message from a host with a spoofed, unreachable IP address to foil the TCP three-way handshake:

- A client host sends a SYN packet to a specific port on the server.
- Next, the server sends a SYN/ACK packet to the client host. The potential connection is now in a SYN\_RECV state.
- Because the system is unreachable, the server never receives an ACK or RST packet back from the client host. The potential connection remains in the SYN\_RECV state, and is placed into a connection queue while it waits for an ACK or RST packet. This “half-opened” connection remains in the queue until the connection-establishment timer expires (when it will be deleted).

To exploit this vulnerability, attackers use attack programs that generate thousands of bogus SYN messages, resulting in denial of service.

When the SYN Protector rulebase is enabled, the IDP engine detects traffic that exceeds the traffic thresholds you set as runtime parameters. [Figure 21 on page 92](#) shows the SYN protector detection settings in the NSM Device Manager configuration editor.

Figure 21: NSM Device Manager: Sensor Settings &gt; Run-Time Parameters

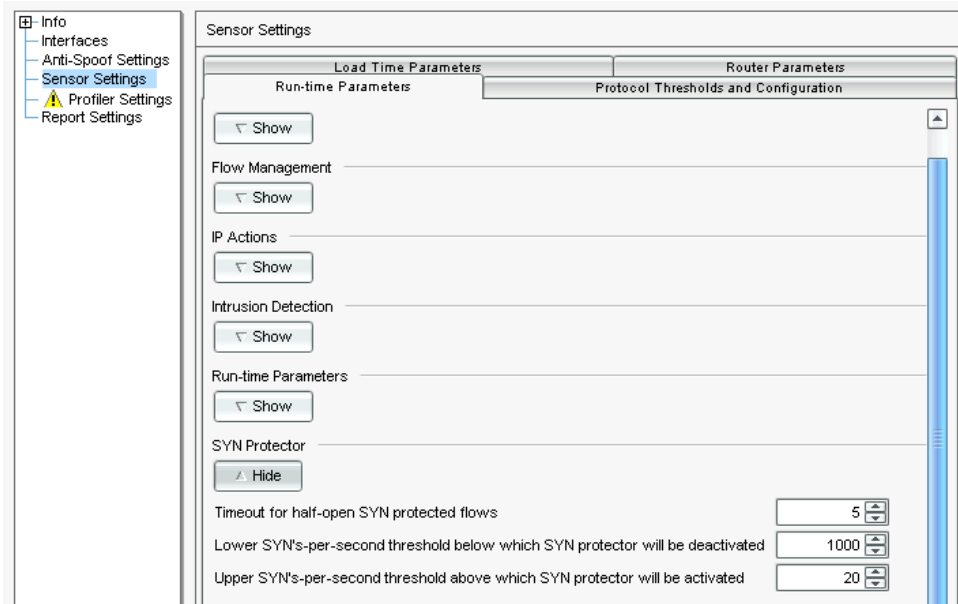


Table 32 on page 92 describes the SYN Protector thresholds.

Table 32: SYN Protector Thresholds

Setting	Description
Timeout for half-open SYN protected flows	<p>Used when SYN Protector is configured in passive mode.</p> <p>A half-open SYN flow occurs during the TCP three-way handshake, after the client has sent a SYN/ACK packet to the server. The half-open connection is now in the SYN_RECV state, and is placed into a connection queue while it waits for an ACK or RST packet. The connection remains in the queue until the connection-establishment timeout expires and the half-open connection is deleted.</p> <p>In passive mode, the IDP Series device monitors session startup. If the client does not send an ACK within the specified timeout, the IDP Series device sends a TCP reset. This setting controls the connection establishment timer, which determines the number of seconds that the IDP engine maintains a half-open SYN protected flow. The default is 5 seconds.</p>
Lower SYN's-per-second threshold below which SYN Protector will be deactivated	<p>Used to activate the SYN Protector in passive or relay mode.</p> <p>In passive mode, the SYN Protector rulebase is activated when the number of SYN packets per second is greater than the sum of the lower SYN's-per-second threshold and the upper SYN's-per-second threshold. The defaults are 1000 and 20. Using the defaults, the SYN Protector is activated when SYN's-per-second reach 1020. The SYN Protector rulebase is deactivated when the number of SYN packets per second falls below the lower SYN's-per-second threshold.</p>
Upper SYN's-per-second threshold above which SYN Protector will be activated	<p>In relay mode, the SYN Protector rulebase is activated when the number of SYN packets per second exceeds the lower SYN's-per-second threshold. The upper threshold is not used.</p>

When you create rules for the SYN Protector rulebase, you specify:

- A source/destination/service match condition



- A response mode: passive or relay
- Notification options

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding SYN Protector Rulebase Match Settings on page 93](#)
- [Understanding SYN Protector Rulebase Modes on page 94](#)
- [Understanding SYN Protector Rulebase Notification Options on page 95](#)
- [Understanding the Components of an IDP Security Policy on page 41](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring SYN Protector Rulebase Rules \(NSM Procedure\) on page 235](#)

## Understanding SYN Protector Rulebase Match Settings

The SYN Protector rulebase becomes active when IDP detects traffic that exceeds the thresholds you set as runtime parameters. [Table 33 on page 93](#) shows the defaults for SYN Protector rulebase detection runtime parameters. You can tune these parameters if safe traffic in your network triggers false positives.

**Table 33: SYN Flood Detection Runtime Parameters**

Parameter	Default
Timeout for half-open SYN protected flows	5
Lower SYN-per-second threshold below which SYN Protector will be deactivated	1000
Upper SYN-per-second threshold above which SYN Protector will be activated.	20

In other words, using the defaults, the SYN Protector rulebase is activated when the IDP Series device counts 1020 SYN packets per second and deactivates when it falls below 1000 SYN packets per second.

When the SYN Protector rulebase is active, the IDP process engine evaluates its rules, beginning with source, destination, and service matching.

Because all TCP-IP is susceptible to a SYN flood attack, we recommend the following settings:

- Source—Any
- Destination—Servers you want to protect
- Service—TCP Any



**TIP:** You can use two rules to protect a large number of servers. Configure rule 1 to match servers you do not need to protect, and set Mode to None. Configure rule 2 to match any traffic and set Mode to Passive or Relay, as you prefer.



**TIP:** In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM documentation.



**NOTE:** The SYN Protector rulebase is a terminal rulebase—that is, SYN Protector rules are inherently terminal rules. If a SYN Protector rule matches, IDP does not process subsequent rules.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Understanding the SYN Protector Rulebase on page 91](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring SYN Protector Rulebase Rules \(NSM Procedure\) on page 235](#)
- [Modifying the IDP Series Device Configuration on page 351](#)

## Understanding SYN Protector Rulebase Modes

Table 34 on page 94 summarizes SYN Protector rulebase modes.

**Table 34: SYN Protector Rulebase Modes**

Mode	Description
None	The IDP Series device takes no action and does not participate in the three-way handshake.
Passive	In passive mode, the IDP Series device monitors session startup. If the client does not send an ACK within a timeout period, the IDP Series device sends a TCP reset.

Table 34: SYN Protector Rulebase Modes (*continued*)

Mode	Description
Relay	<p>In relay mode, the IDP Series device acts as a relay for the connection establishment, performing the three-way handshake with the client on behalf of the server. When the IDP Series device receives the initial SYN packet, it returns a SYN/ACK packet with a SYN cookie. A SYN cookie is a 32-bit number that is put into the TCP sequence number field of a packet. If the client replies with an ACK packet with the appropriate cookie, the IDP Series device completes the three-way handshake and allows the session to become established. If the IDP Series device does not receive an appropriate ACK packet from the client, as is the case in a SYN flood attack, the IDP Series device does not establish the connection. Relay mode guarantees that the server allocates resources only to connections that are already in an established state. The relay is transparent to both the client and server.</p> <p>Relay mode has the following limitations:</p> <ul style="list-style-type: none"> <li>• When the ACK packet from the client is lost, it can potentially lead to an unsynchronized state between client and server.</li> <li>• Because the IDP Series device does not save TCP options found in SYN packets, TCP extensions used for efficient transaction-oriented service (T/TCP) and Selective Acknowledgment (SACK), or protocols such as BGP, have a problem when SYN flooding is detected and the IDP Series device initiates the proxy TCP handshake.</li> <li>• Relay mode can be susceptible to ACK flooding because the IDP Series device must check for the validity of a cookie in the ACK messages.</li> </ul> <p><b>NOTE:</b> Relay mode might not work as expected for MPLS traffic. When the IDP engine processes MPLS traffic, it stores the MPLS label information for traffic in each direction. In the case of traffic that matches SYN Protector rules in relay mode, the IDP Series device is programmed to send a SYN-ACK before the traffic has reached the server. In these cases, the IDP engine does not have server-to-client MPLS label information. Therefore, the SYN-ACK packet does not include an MPLS label. Some MPLS routers can add packets without a label to an existing MPLS tunnel; others drop such packets.</p>



**TIP:** You can use two rules to protect a large number of servers. Configure rule 1 to match servers you do not need to protect, and set Mode to None. Configure rule 2 to match any traffic and set Mode to Passive or Relay, as you prefer.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the SYN Protector Rulebase on page 91](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring SYN Protector Rulebase Rules \(NSM Procedure\) on page 235](#)

## Understanding SYN Protector Rulebase Notification Options

By default, notification is not enabled for SYN Protector rulebase rules. You have the option to enable notification options. [Table 35 on page 96](#) describes these options.

Table 35: SYN Protector Rulebase Notification Options

Option	Description
Event logs and alerts	<p>You can enable the following delivery and handling options for logs:</p> <ul style="list-style-type: none"> <li>• Send to NSM log viewer.</li> <li>• Send to NSM log viewer and flag as an alert.</li> <li>• Send to an e-mail address list.</li> <li>• Send to syslog.</li> <li>• Send to SNMP trap.</li> <li>• Save in XML format.</li> <li>• Save in CVS format.</li> <li>• Process with a script.</li> </ul>
Packet captures	Packet capture is not available for SYN Protector rulebase rules.



**NOTE:** SYN Protector rulebase notification options are the same as IDP rulebase options, except that packet capture is not applicable.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the SYN Protector Rulebase on page 91](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring SYN Protector Rulebase Rules \(NSM Procedure\) on page 235](#)

## CHAPTER 10

# The Traffic Anomalies Rulebase

- [Understanding the Traffic Anomalies Rulebase on page 97](#)
- [Understanding Traffic Anomalies Rulebase Match Conditions on page 99](#)
- [Understanding Traffic Anomalies Rulebase Detection Settings on page 100](#)
- [Understanding Traffic Anomalies Rulebase IP Actions on page 100](#)
- [Understanding Traffic Anomalies Rulebase Notification Options on page 101](#)

### Understanding the Traffic Anomalies Rulebase

---

The Traffic Anomalies rulebase employs a traffic flow analysis method to detect attacks that occur over multiple connections and sessions (such as scans).

A *traffic anomaly* is a pattern that indicates abnormal network activity. Traffic generated by automated port scanning tools trigger Traffic Anomalies rulebase rules. Attackers use automated port scanning tools to perform reconnaissance on your network. Typically, an automated port scanning tool attempts to connect to every port on a single machine (port scanning) or to connect to multiple IP addresses on a network (network scanning). Attackers do this to determine which services are allowed and responding on your network, so they can focus attacks on any vulnerabilities.

Traffic Anomalies rulebase rules count the number of ports scanned in a specified time period and use this traffic flow analysis to identify scans, as well as other attacks that occur over multiple connections and sessions. If the rule detects an attack, you can drop the connection and block the IP address that originated the attack. The IDP engine takes action against traffic that exceeds the thresholds you set.

[Table 36 on page 97](#) summarizes Traffic Anomalies rulebase detection settings. You can tune these parameters if safe traffic in your network triggers false positives.

**Table 36: Traffic Anomalies Rulebase Detection Settings**

Group	Description
TCP scans, UDP Port Scans	<p>Set a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default port count is 20. The default time threshold is 120 seconds. The rule is matched if the same source scans 20 TCP ports on your internal network within 120 seconds or if the same source scans 20 UDP ports on your internal network within 120 seconds.</p>

Table 36: Traffic Anomalies Rulebase Detection Settings (*continued*)

Group	Description
Distributed Port Scan	<p>A distributed port scan is an attack that uses multiple source IP addresses to scan ports.</p> <p>Set a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default IP count is 50. The default time threshold is 120 seconds. The rule is matched if 50 IP addresses attempt to scan ports on your internal network within 120 seconds.</p>
ICMP Sweep	<p>An ICMP sweep is an attack where a single source IP pings multiple IP addresses.</p> <p>Set a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default IP count is 50. The default time threshold is 120 seconds. The rule is matched if the same source IP attempts to ping 50 IP addresses within 120 seconds.</p>
Network Scan	<p>A network scan is an attack where a single source IP scans multiple IP addresses.</p> <p>Set a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default IP count is 50. The default time threshold is 120 seconds. The rule is matched if the same source IP attempts to scan 50 IP addresses within 120 seconds.</p>
Session Limit	<p>Set a threshold number of sessions allowed from a single host within a second. The default is 100 sessions.</p> <p>For example, assume your internal network typically has low volume traffic. To detect a sudden increase in traffic from a specific host (which might indicate a worm), configure a rule that matches traffic over your internal network and configure a limit of 200. To block traffic that exceeds the session limit, set an IP action of <b>IDP Block</b> and select <b>Source, Protocol</b> from the Blocking Options menu.</p>

In addition, you can tune runtime parameters for Traffic Signatures. [Table 37 on page 98](#) describes the runtime parameters associated with the Traffic Anomalies rulebase.

Table 37: Traffic Signature Runtime Settings

Setting	Description
Byte threshold for suspicious flows	Scans typically use small packets to access targets. You can exclude flows that contain large packets to reduce false positives. The default is to exclude flows where packet size exceeds 20 bytes.
Reporting frequency while scan in progress (seconds)	Specifies how frequently log messages are generated. Default is 30 seconds.
The number of IP addresses we track for session rate	Specifies the maximum number of source IP addresses for which session rate is calculated. Default is 32,767.

When you create rules for the Traffic Anomalies rulebase, you specify:

- A source/destination/service match condition
- Detection settings
- Response options
- Notification options

For complete procedures on configuring Traffic Anomalies rulebase rules, see the *IDP Series Administration Guide*.

**Related  
Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding Traffic Anomalies Rulebase Match Conditions on page 99](#)
- [Understanding Traffic Anomalies Rulebase Detection Settings on page 100](#)
- [Understanding Traffic Anomalies Rulebase IP Actions on page 100](#)
- [Understanding Traffic Anomalies Rulebase Notification Options on page 101](#)
- [Understanding the Components of an IDP Security Policy on page 41](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Traffic Anomalies Rulebase Rules \(NSM Procedure\) on page 237](#)

---

## Understanding Traffic Anomalies Rulebase Match Conditions

We recommend the following settings for Traffic Anomalies rulebase match conditions:

- Source — Any
- Destination — IP addresses for servers you want to protect
- Service — Any (or specify specific services if you are creating an ignore list)



**TIP:** You can use two rules to protect a large number of servers. Configure rule 1 to match services you do not want to detect and set the detection option to Ignore. Configure rule 2 to match any traffic and set the detection operation to Detect.



**TIP:** In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM documentation.



**NOTE:** The Traffic Anomalies rulebase is a terminal rulebase—that is, Traffic Anomalies rules are inherently terminal rules. If a Traffic Anomalies rule matches, IDP does not process subsequent rules.

**Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Understanding the Traffic Anomalies Rulebase on page 97](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Traffic Anomalies Rulebase Rules \(NSM Procedure\) on page 237](#)

---

## Understanding Traffic Anomalies Rulebase Detection Settings

The Traffic Anomalies rulebase detection setting is a toggle detection off and on.

Specify **Ignore** to turn off traffic anomaly detection for traffic that matches the rule.

Specify **Detect** to turn on detection for traffic that matches the rule and to configure detection settings.



**TIP:** You can use two rules to protect a large number of servers. Configure rule 1 to match services you do not want to detect and set the detection option to **Ignore**. Configure rule 2 to match any traffic and set the detection option to **Detect**.



**TIP:** In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM documentation.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Traffic Anomalies Rulebase on page 97](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Traffic Anomalies Rulebase Rules \(NSM Procedure\) on page 237](#)

---

## Understanding Traffic Anomalies Rulebase IP Actions

If traffic matches a traffic anomalies rule, the IDP Series device can take action against the current connection and against subsequent network traffic from the same IP address.



Such actions are called *IP actions*. By default, the specified IP action is permanent (timeout = 0). If you prefer, you can set a timeout.

[Table 38 on page 101](#) describes Traffic Anomalies rulebase IP actions.

**Table 38: Traffic Anomalies Rulebase IP Actions**

IP Action	Description
IP Block	<p>IDP blocks the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP address</li> <li>• Source subnet</li> <li>• Protocol</li> <li>• Destination IP address</li> <li>• Destination subnet</li> <li>• Destination port</li> <li>• From zone</li> </ul>
IP Close	<p>IDP closes the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP address</li> <li>• Source subnet</li> <li>• Protocol</li> <li>• Destination IP address</li> <li>• Destination subnet</li> <li>• Destination port</li> <li>• From zone</li> </ul>
IP Notify	IDP does not take any action against future traffic but logs the event or sends an alert.



**NOTE:** Traffic Anomalies rulebase IP actions are the same IP actions available for IDP rulebase rules.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Traffic Anomalies Rulebase on page 97](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Traffic Anomalies Rulebase Rules \(NSM Procedure\) on page 237](#)

## Understanding Traffic Anomalies Rulebase Notification Options

By default, logging is enabled for Traffic Anomalies rulebase rules. [Table 39 on page 102](#) describes notification options. You also have the option of disabling logging.

Table 39: Traffic Anomalies Rulebase Notification Options

Option	Description
Event logs and alerts	<p>You can enable the following delivery and handling options for logs:</p> <ul style="list-style-type: none"> <li>• Send to NSM log viewer.</li> <li>• Send to NSM log viewer and flag as an alert.</li> <li>• Send to an e-mail address list.</li> <li>• Send to syslog.</li> <li>• Send to SNMP trap.</li> <li>• Save in XML format.</li> <li>• Save in CVS format.</li> <li>• Process with a script.</li> </ul>
Packet captures	Packet capture is not available for Traffic Anomalies rulebase rules.



**NOTE:** Traffic Anomalies rulebase notification options are the same as IDP rulebase options, except that packet capture is not applicable.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Traffic Anomalies Rulebase on page 97](#)
- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Traffic Anomalies Rulebase Rules \(NSM Procedure\) on page 237](#)

## CHAPTER 11

# The Network Honeypot Rulebase

- [Understanding the Network Honeypot Rulebase on page 103](#)
- [Understanding Network Honeypot Rulebase Match Settings on page 104](#)
- [Understanding Network Honeypot Operation Setting on page 104](#)
- [Understanding Network Honeypot Rulebase IP Actions on page 105](#)
- [Understanding Network Honeypot Rulebase Notification Options on page 106](#)

## Understanding the Network Honeypot Rulebase

---

The Network Honeypot rulebase is a method to detect reconnaissance activities.

A *network honeypot* is an apparently vulnerable system that draws the attention and action of attackers. In an IDP network honeypot, the IDP Series device impersonates ports on protected servers.

When you create rules for the Network Honeypot rulebase, you specify:

- A destination/service match condition
- Operation mode
- Response options
- Notification options



**NOTE:** The IDP Series device drops MPLS traffic that matches a Network Honeypot rule. When the IDP engine processes MPLS traffic, it stores the MPLS label information. It stores separate labels for client-to-server and server-to-client communication. In the case of traffic that matches Network Honeypot rules, there is no genuine server-to-client communication, so the IDP engine does not have server-to-client MPLS label information. Therefore, the impersonation operation cannot be supported for MPLS traffic.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding Network Honeypot Rulebase Match Settings on page 104](#)
- [Understanding Network Honeypot Operation Setting on page 104](#)

- [Understanding Network Honeypot Rulebase IP Actions on page 105](#)
- [Understanding Network Honeypot Rulebase Notification Options on page 106](#)
- [Understanding the Components of an IDP Security Policy on page 41](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Network Honeypot Rulebase Rules \(NSM Procedure\) on page 240](#)

---

## Understanding Network Honeypot Rulebase Match Settings

Network Honeypot rulebase rules are triggered when a source IP address makes a connection request to the destination IP address and service specified in the rule.

We recommend you set source to **Any**; set destination and service to the server and service you want to appear to be available.



**TIP:** In NSM, you can create address objects and service objects to facilitate configuration. One benefit of using objects is that you can configure them once and then use them in multiple rules. For details, see the NSM documentation.



**NOTE:** The Network Honeypot rulebase is a terminal rulebase—that is, Network Honeypot rules are inherently terminal rules. If a Network Honeypot rule matches, IDP does not process subsequent rules.

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Network Honeypot Rulebase on page 103](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Network Honeypot Rulebase Rules \(NSM Procedure\) on page 240](#)

---

## Understanding Network Honeypot Operation Setting

The Network Honeypot rulebase operation setting is used to toggle the network honeypot on and off.

Specify **Impersonate** to turn it on; specify **Ignore** to turn it off.

If the Network Honeypot rulebase is turned on, an attacker attempts to connect to an impersonated port, and the rule matches, the IDP Series device responds with a TCP SYN/ACK.

**Related Documentation** The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Network Honeypot Rulebase on page 103](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Network Honeypot Rulebase Rules \(NSM Procedure\) on page 240](#)

## Understanding Network Honeypot Rulebase IP Actions

If traffic matches a Network Honeypot rule, the IDP Series device can take action against the current connection and against subsequent network traffic from the same IP address. Such actions are called *IP actions*. By default, the specified IP action is permanent (timeout = 0). If you prefer, you can set a timeout.

[Table 40 on page 105](#) describes Network Honeypot rulebase IP actions.

**Table 40: Network Honeypot Rulebase IP Actions**

IP Action	Description
IP Block	<p>IDP blocks the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP address</li> <li>• Source subnet</li> <li>• Protocol</li> <li>• Destination IP address</li> <li>• Destination subnet</li> <li>• Destination port</li> <li>• From zone</li> </ul>
IP Close	<p>IDP closes the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP address</li> <li>• Source subnet</li> <li>• Protocol</li> <li>• Destination IP address</li> <li>• Destination subnet</li> <li>• Destination port</li> <li>• From zone</li> </ul> <p><b>NOTE:</b> The IP Close action might not work as expected for MPLS traffic. When the IDP engine processes MPLS traffic, it stores the MPLS label information for traffic in each direction. In the case of an IP action, the IDP engine is programmed to take a server-to-client action before the traffic has reached the server. In these cases, the IDP engine does not have server-to-client MPLS label information. Therefore, the TCP reset packet does not include an MPLS label. Some MPLS routers can add packets without a label to an existing MPLS tunnel; others drop such packets.</p>
IP Notify	IDP does not take any action against future traffic but logs the event or sends an alert.



**NOTE:** Network Honeypot rulebase IP actions are the same IP actions available for IDP rulebase rules.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Network Honeypot Rulebase on page 103](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Network Honeypot Rulebase Rules \(NSM Procedure\) on page 240](#)

## Understanding Network Honeypot Rulebase Notification Options

By default, logging is not enabled for Network Honeypot rulebase rules. You have the option to enable notification options. [Table 41 on page 106](#) describes these options.

**Table 41: Network Honeypot Rulebase Notification Options**

Option	Description
Event logs and alerts	<p>You can enable the following delivery and handling options for logs:</p> <ul style="list-style-type: none"> <li>• Send to NSM log viewer.</li> <li>• Send to NSM log viewer and flag as an alert.</li> <li>• Send to an e-mail address list.</li> <li>• Send to syslog.</li> <li>• Send to SNMP.</li> <li>• Save in XML format.</li> <li>• Save in CVS format.</li> <li>• Process with a script.</li> </ul>
Packet captures	<p>Viewing the packets used in an attack on your network can help you determine the extent of the attempted attack and its purpose, whether or not the attack was successful, and any possible damage to your network.</p> <p>If multiple rules with packet capture enabled match the same attack, IDP captures the maximum specified number of packets. For example, you configure rule 1 to capture 10 packets before and after the attack, and you configure rule 2 to capture 5 packets before and after the attack. If both rules match the same attack, IDP attempts to capture 10 packets before and after the attack.</p> <p>You can capture up to 256 packets before the event and 256 packets after the event.</p> <p><b>NOTE:</b> If necessary, you can improve performance by logging only the packets received after the attack.</p>



**NOTE:** Network Honeypot rulebase notification options are the same as IDP rulebase options.

**Related  
Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Network Honeypot Rulebase on page 103](#)
- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Network Honeypot Rulebase Rules \(NSM Procedure\) on page 240](#)





# Additional Security Features

- [IP Spoof Attack Prevention Overview on page 109](#)

## IP Spoof Attack Prevention Overview

---

Every IP packet includes the destination address (where the packet is going) and the source address (where the packet came from). The routers that provide Internet communication between distant computers determine the best route for the IP packet using only the destination address and typically ignore the source address.

Attackers, who typically do not want you to know where an attack is coming from, can fake the source address of a malicious IP packet (by modifying the packet headers) so that the packet appears to come from a trusted system. The use of a fake IP address is called *IP spoofing*. You can configure the IDP system to detect these irregularities.

To detect attacks that attempt to spoof the addresses of hosts in your protected network, you can associate IDP traffic interfaces with the addresses of hosts in your protected network. IDP then detects an IP spoof attack if:

- An incoming packet uses an IP address that belongs to a network object on your internal network.
- An outgoing packet uses an IP address that does not belong to a network object on your internal network.

You can configure whether IDP drops or logs the session with a spoofed IP address.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying the IDP Series Device Configuration on page 351](#)



## CHAPTER 13

# Inspection of Encapsulated and Encrypted Traffic

- [Inspection of GRE Traffic Overview on page 111](#)
- [Inspection of GTP Traffic Overview on page 111](#)
- [Inspection of IPsec VPN Traffic Overview on page 112](#)
- [Inspection of MPLS Traffic Overview on page 112](#)
- [Inspection of SSL Traffic Overview on page 113](#)

## Inspection of GRE Traffic Overview

---

Generic Routing Encapsulation (GRE) is a tunneling protocol designed to encapsulate a wide variety of network layer packets inside IP tunneling packets. The original packet is the payload for the final packet. The protocol is used on the Internet to secure virtual private networks.

To inspect the payload of an encapsulated packet, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for IP-in-GRE and PPP-in-GRE. You can configure decapsulation for one or two layers.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Enabling Inspection of GRE Traffic on page 313](#)

## Inspection of GTP Traffic Overview

---

GPRS Tunneling Protocol (or GTP) is an IP-based protocol used within Global System for Mobile communication (GSM) and Universal Mobile Telecommunications System (UMTS) networks.

To inspect the payload of an encapsulated traffic, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for UDP GTPv0 and GTPv1. You can configure decapsulation for one or two layers.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Enabling Inspection of GTP Traffic on page 315](#)

## Inspection of IPsec VPN Traffic Overview

---

Internet Protocol Security (IPsec) virtual private networks use the Encapsulated Security Payload (ESP) protocol and the NULL encryption algorithm to ensure the authenticity, integrity, and confidentiality of IP packets.

To inspect the payload of an encapsulated packet, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for IPsec ESP NULL traffic. You can configure decapsulation for one or two layers.

- Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:
- [Enabling Inspection of IPsec VPN Traffic on page 317](#)

## Inspection of MPLS Traffic Overview

---

Multiprotocol Label Switching (MPLS) is an IP label switching technology that enables predetermined paths to specific destinations, called Label Switched Paths (LSPs), to be established through an inherently connectionless IP network. In MPLS networks, packets contain short labels that describe how to forward them through the network.

With MPLS decapsulation enabled, the IDP engine can inspect the IPv4 payload and pass through non-IPv4 payload. Note the following requirements and limitations:

- The IDP engine cannot decapsulate other encapsulated protocols within an MPLS frame. For example, the IDP engine cannot decapsulate the MPLS frame, find a GRE frame, decapsulate the GRE, and inspect the payload. Instead, the IDP engine passes through such traffic.
- If your traffic uses Ethernet frames larger than 1750 bytes, you must ensure the IDP default maximum frame size is sufficient (the default maximum frame size is 9014 bytes). In addition, we recommend you set the maximum transmission unit (MTU) on the switch or router connected to the IDP Series device to 1750 bytes or lower.

The IDP Series device does not participate in Label Distribution Protocol (LDP). When the IDP Series device receives the traffic, the IDP engine stores the MPLS label stacks of client-to-server or server-to-client directions. After processing the flow, the IDP Series device forwards the IP frames with the label stack it had stored when it created the flow, relying on the label switch router (LSR) to add the correct MPLS labels to the packet.

In some cases, the IDP engine is programmed to act in the server-to-client direction before it has seen and stored a server-to-client MPLS label. In effect, these connections are dropped. You might observe dropped MPLS traffic if the following rule elements apply:

- IDP rulebase – Action: Close Client (limitation applies to VLAN tagged traffic only)
- IDP rulebase – IP action: IP Close

- SYN Protector rulebase – Relay mode
- Network Honeypot rulebase – Impersonate mode

MPLS support is not enabled by default. You can use the CLI to enable MPLS support.

**Related  
Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Enabling Inspection of MPLS Traffic on page 318](#)

---

## Inspection of SSL Traffic Overview

Secure Sockets Layer (SSL) is a cryptographic protocol that adds security to TCP/IP communication. Several versions of the SSL and Transport Layer Security (TLS) protocols are in widespread use in applications like Web browsing, electronic mail, Internet faxing, instant messaging, and voice over IP (VoIP). SSL and TLS encrypt the Transport Layer protocol datagrams that carry the payload of these communications. While encryption is an excellent way to keep private data from prying eyes, without inspection by the IDP Series device, it also unwittingly opens a network to dangerous viruses, trojans, or network attacks.

To inspect the HTTP payload of HTTPS traffic, the IDP Series device must decrypt the HTTPS session. Your security policy can examine both the SSL session and the decrypted HTTP payload.

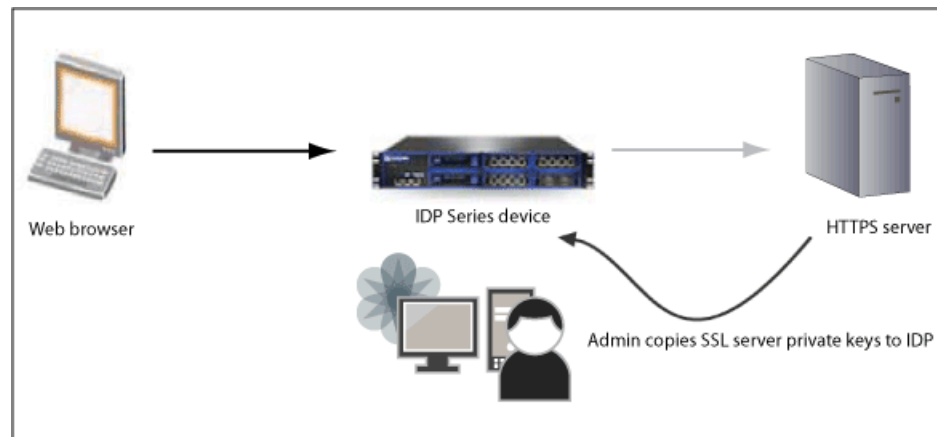
The following sections describe alternative methods you can use to enable SSL inspection:

- [Using the SSL Server Private Keys on page 113](#)
- [Using a Root Certificate Authority in SSL Forward Proxy Operations on page 114](#)
- [Supported SSL Specifications on page 115](#)

### Using the SSL Server Private Keys

Beginning with IDP OS Release 3.2r1, we support inspection of client-to-server traffic to internal SSL servers. As shown in [Figure 22 on page 114](#), this method depends on administrative access to the SSL server private keys.

Figure 22: SSL Inspection Using SSL Server Private Keys

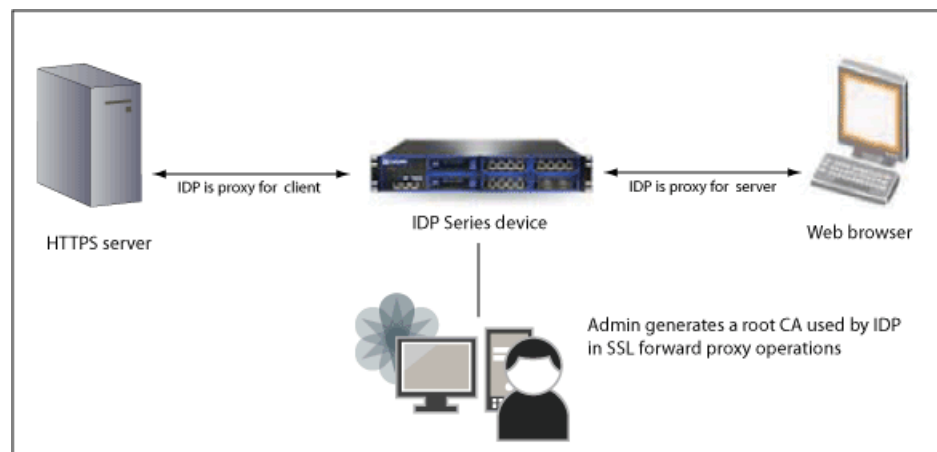


You must be able to copy the SSL server private key to the IDP SSL keystore. The IDP Series device uses the key to decrypt the inbound traffic so that it can inspect the payload. The private key must be in Privacy-Enhanced Mail (PEM) format. We have verified support for the following RSA private key lengths: 1024 bits, 2048 bits, 3072 bits, and 4096 bits.

### Using a Root Certificate Authority in SSL Forward Proxy Operations

Beginning with IDP OS Release 5.0r2, we support inspection of traffic to HTTPS servers where you do not have access to the SSL private key, such as outbound traffic to the WWW. As shown in [Figure 23 on page 114](#), this method uses a root certificate authority (CA) to proxy the SSL key negotiation. The IDP Series device inserts itself into the SSL key negotiation phase so that it can decrypt the HTTPS session and inspect the session and payload according to your security policy.

Figure 23: SSL Inspection Using a Root CA



When the special root CA is present, the IDP Series device intercepts the HTTPS connection and makes a request to the server as if it were the client; it presents to the client a CA (derived from the special root CA) as if it were the server. The IDP Series device then negotiates the key exchange, decrypts the session, inspects the payload, and re-encrypts the session as necessary before forwarding.

To ensure employee or customer privacy, you can configure a whitelist to exclude matching sessions from being processed by the SSL forward proxy feature. Traffic to destination servers on your whitelist is not intercepted and is passed through uninspected.



**NOTE:** When both the CA and server private keys are present, the IDP Series device uses the SSL forward proxy method to inspect HTTPS traffic.

Supported SSL Specifications

The IDP Series device supports decryption of HTTPS traffic that uses SSLv3 and TLSv1. The IDP Series device can inspect an SSLv2 header for anomalies, but it cannot decrypt and examine the HTTP payload in such sessions. In addition, the IDP Series device does not support inspection of compressed TLS traffic.

Table 42 on page 115 lists the SSL cipher suites supported by the two IDP SSL inspection methods.

Table 42: Supported SSL Cipher Suites

Cipher Suite	Decryption Using Private Keys	Decryption Using Forward Proxy
Name: TLS_RSA_WITH_NULL_MD5 Authorization: RSA Key Exchange: RSA Encryption: NULL Digest: MD5	Yes	Yes
Name: TLS_RSA_WITH_NULL_SHA Authorization: RSA Key Exchange: RSA Encryption: NULL Digest: SHA	Yes	Yes
Name: TLS_RSA_WITH_RC4_128_MD5 Authorization: RSA Key Exchange: RSA Encryption: RC4_128 Digest: MD5	Yes	Yes

Table 42: Supported SSL Cipher Suites (*continued*)

Cipher Suite	Decryption Using Private Keys	Decryption Using Forward Proxy
Name: TLS_RSA_WITH_RC4_128_SHA Authorization: RSA Key Exchange: RSA Encryption: RC4_128 Digest: SHA	Yes	Yes
Name: TLS_RSA_WITH_DES_CBC_SHA Authorization: RSA Key Exchange: RSA Encryption: DES_CBC Digest: SHA	Yes	Yes
Name: TLS_RSA_WITH_3DES_EDE_CBC_SHA Authorization: RSA Key Exchange: RSA Encryption: 3DES_EDE_CBC Digest: SHA	Yes	Yes
Name: TLS_RSA_WITH_AES_128_CBC_SHA Authorization: RSA Key Exchange: RSA Encryption: AES_128_CBC Digest: SHA	Yes	Yes
Name: TLS_RSA_WITH_AES_256_CBC_SHA Authorization: RSA Key Exchange: RSA Encryption: AES_256_CBC Digest: SHA	Yes	Yes

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Example: Implementing Inspection of Outbound SSL Traffic on page 179](#)
- [Example: Exempting Outbound SSL Traffic from Inspection on page 181](#)



The following related topics are included in the *IDP Series Administration Guide*:

- [Using the SSL Private Server Key to Enable Inspection of SSL Traffic on page 308](#)
- [Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic on page 311](#)
- [Exempting HTTPS Traffic from Inspection on page 312](#)



## PART 2

# Examples

- [Simulation Mode on page 121](#)
- [Using Profiler and Application Volume Tracking on page 123](#)
- [Logging on page 139](#)
- [IDP Rulebase Examples on page 153](#)
- [APE Rulebase Examples on page 165](#)
- [Exempt and Backdoor Rulebase Examples on page 175](#)
- [Inspection of HTTPS Traffic on page 179](#)



# Simulation Mode

- [Example: Getting Started with Simulation Mode on page 121](#)

## Example: Getting Started with Simulation Mode

---

The primary use case for simulation mode is when you are evaluating the effectiveness of the IDP Series device as the intrusion prevention system for your network.

Follow these basic steps to get started:

1. Read the release notes for your release. The release notes contain important release-related information about release-specific features, unsupported features, changed features, fixed issues, and known issues. The information in the release notes is more current than the information in this guide.
2. Install the IDP Series appliance, connect the management interface to your network, configure network settings, and configure virtual routers in transparent mode (in-path) or sniffer mode (out-of-path). For details, see the installation guide for your IDP Series device.
3. Upgrade IDP Series software to the latest version (if applicable).
4. Add the IDP Series device to the NSM Device Manager.
5. Update the IDP detector engine and NSM attack object database.
6. Become familiar with the default security policy (named Recommended).
7. Use the command-line interface to enable simulation mode.
8. Connect transit interfaces to the firewall and/or switch. See the installation guide for your IDP Series device.
9. Use the documentation to become familiar with the product features and user interface:
  - Use the *IDP Series Concepts and Examples Guide* to become familiar with IDP Series features.
  - Use this guide, the *IDP Series Administration Guide*, to learn the steps to implement IDP Series features and monitor security events.
  - Use the Appliance Configuration Manager (ACM) online Help for information about using ACM.

- Use CLI man pages for syntax and parameter hints for CLI commands.
  - Use the NSM online Help for information about using the NSM user interface.
10. Run Profiler to discover the network hosts you want to protect.
  11. Review logs to verify the initial deployment.
  12. Fine-tune your security policy.

**Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Simulation Mode Overview on page 33](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Updating IDP OS Software \(NSM Procedure\) on page 390](#)
- [Adding IDP Series Devices to NSM Device Manager on page 344](#)
- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)
- [Developing Security Policies Task Summary on page 195](#)
- [Enabling Simulation Mode on page 201](#)
- [Profiler Task Summary on page 203](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## CHAPTER 15

# Using Profiler and Application Volume Tracking

- [Example: Using Profiler to Set a Baseline on page 124](#)
- [Example: Using Profiler to Alert You to New Hosts and Port Activity on page 129](#)
- [Example: Identifying Services That Use Nonstandard Ports on page 129](#)
- [Example: Responding to Vulnerability Announcements with Due Diligence on page 130](#)
- [Example: Using Profiler to Investigate Unanticipated Attacks on page 131](#)
- [Example: Using Profiler to Mitigate Risks from Laptops on page 132](#)
- [Example: Using NSM to Enable and View Application Volume Tracking on page 133](#)

## Example: Using Profiler to Set a Baseline

A baseline is a place to start. Baseline data gives you the building blocks for your network security policy. The first time you use Profiler, the Profiler report will provide you with detailed views of the devices and applications that communicate in your network.

You use the baseline to:

- Determine which hosts to protect with security policies.
- Determine the applications that communicate over the network and therefore which services require protection in general.
- Determine specific operating systems and software versions in use and therefore which security policy attack objects are relevant and which may be exempted.
- Determine which security policy rulebases are relevant.
- Determine session contexts that can be safely ignored and those that should be monitored.

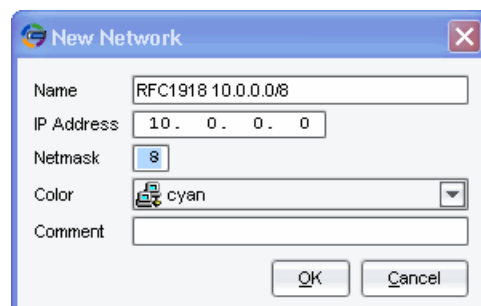
This example assumes a network that uses the private address space defined in RFC 1918: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.

To discover hosts and applications in your private network:

1. Use NSM to create network address objects for each of the three private address spaces.

Figure 24 on page 124 shows the NSM network address object editor.

**Figure 24: NSM Network Address Object Editor**



2. Create address objects for any additional networks or hosts that you are aware of.

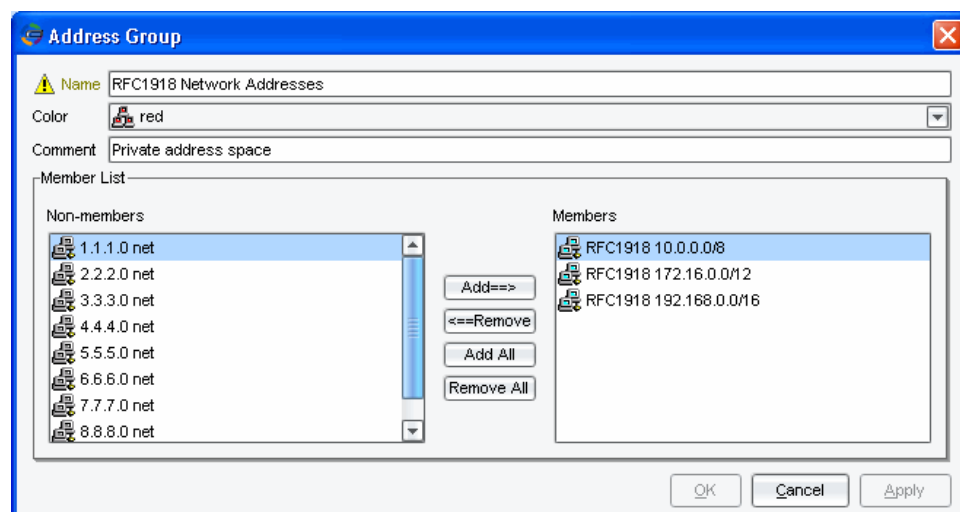
The Profiler detects host and application information for all traffic that traverses it. If it cannot match the traffic to your network address objects, it assumes you are not tracking the host and populates the source or destination fields as **Non-tracked IP**.

3. Optionally, create a group object to contain the private address space networks.

Figure 25 on page 125 shows the NSM group object editor.



Figure 25: NSM Group Object Editor



4. In the NSM Device Manager, right-click the IDP Series device and select **IDP Profiler > Start Profiler**.

Figure 26 on page 125 shows how to navigate in NSM Device Manager to start Profiler.

Figure 26: Starting Profiler from NSM Device Manager

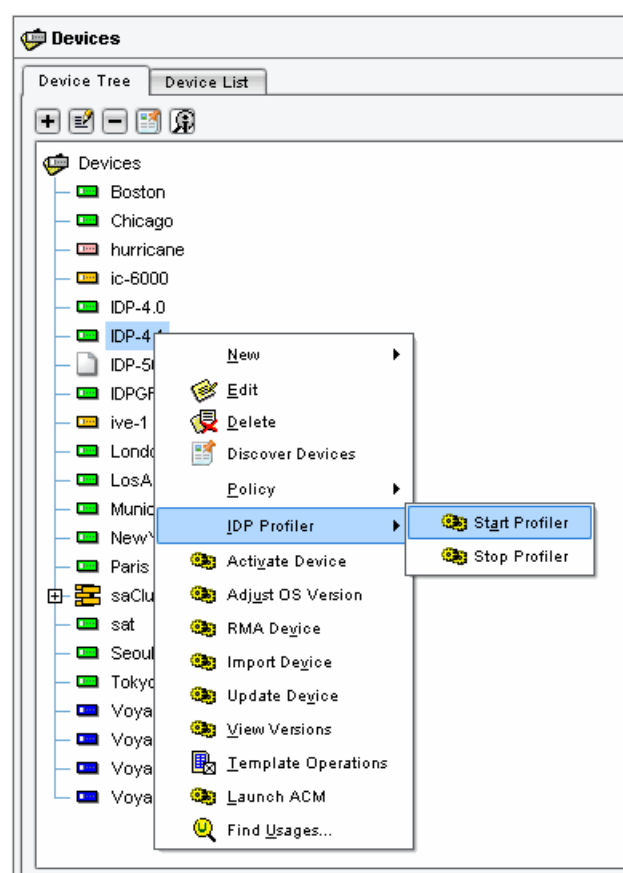
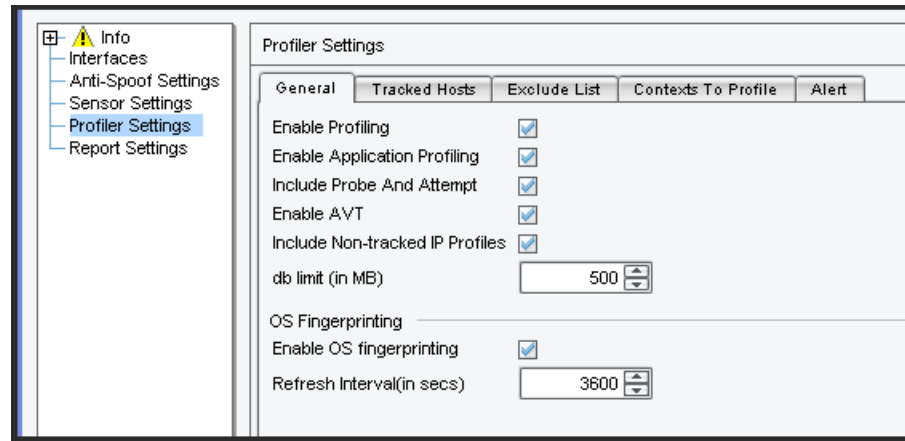


Figure 27 on page 126 shows the Profiler configuration tabs.

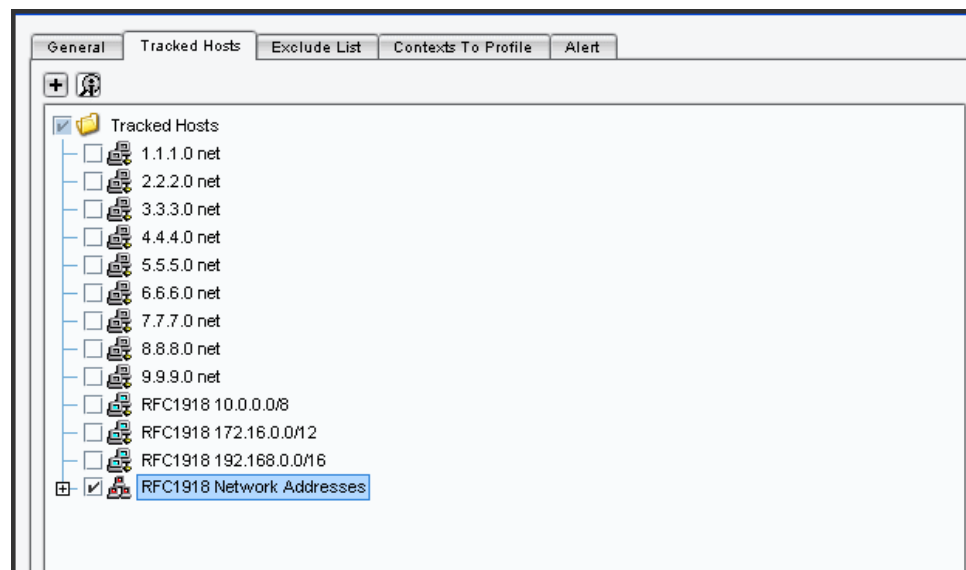
Figure 27: NSM Profiler Configuration Tabs



5. In the General tab, check the boxes to enable profiling, application profiling, OS fingerprinting, and non-tracked IP addresses.
6. Click the **Tracked Hosts** tab and add the address objects you created in Step 1.

Figure 28 on page 126 shows the Tracked Hosts tab.

Figure 28: NSM Profiler Tracked Hosts Tab



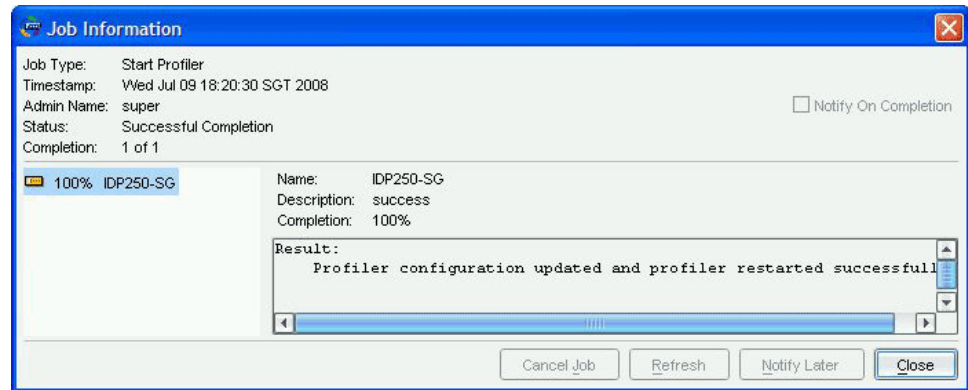
7. Click the **Contexts to Profile** tab and select all contexts.
8. Click the **Alert** tab and clear all alerts. You can use alerts after you have established your baseline but you do not need them in this initial procedure.
9. Click **Apply** to update the Profiler configuration and start the Profiler update job.

The Profiler detects network traffic that traverses the path of the IDP Series device. Consequently, it takes time to build the Profiler database. In most networks, critical

services are used frequently and you might see data in five or ten minutes. For best results, let the Profiler run for a full business day to ensure that it has had enough time to monitor all pertinent network traffic.

Figure 29 on page 127 shows the Job Information window that appears when the Profiler update job is completed.

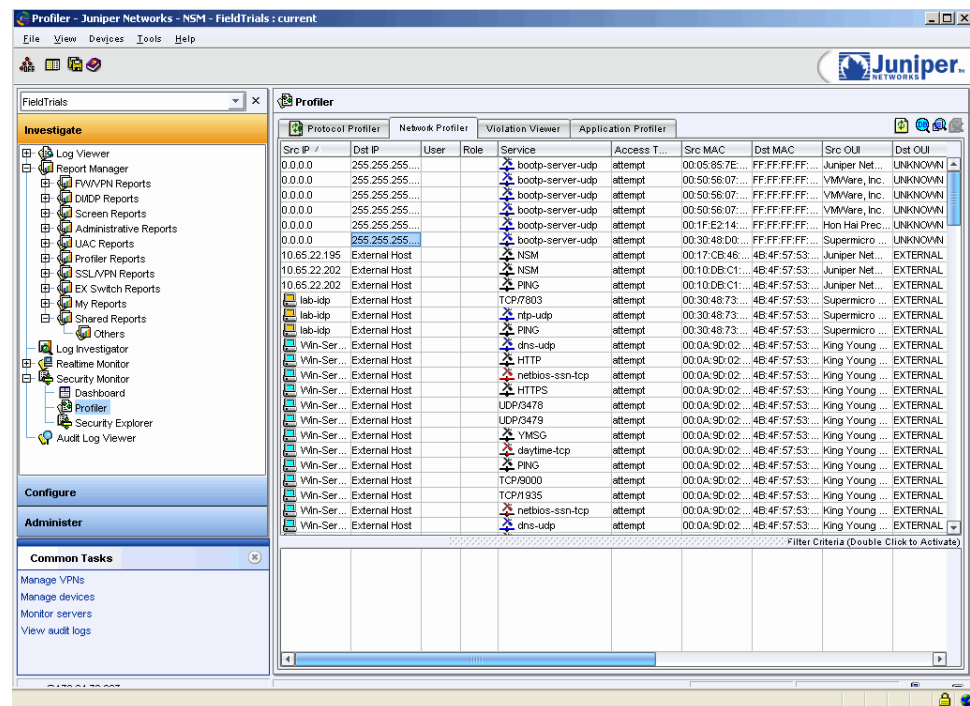
Figure 29: NSM Profiler Update Job Information Window



10. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler** to display the Profiler viewer.
11. Click the **Network Profiler** tab and examine the data gathered about hosts in your network.

Figure 30 on page 128 shows the Network Profiler tab.

Figure 30: Profiler: Network Profiler Tab



12. Optionally, use the Profiler data to create address objects and groups that you can later use when you create security policy rules.

For example, to create groups for SMTP servers, DNS servers, Windows AD servers, and, HTTP servers:

- a. Create group objects, such as SMTP, DNS, Windows AD, and HTTP.
- b. Use NSM UI features to filter and sort Profiler table rows by service.
- c. Double-click a destination entry to display the host editor, populated with data for the row you clicked.
- d. If you have created a group for the server type, such as SNMP, assign the host to the group.
- e. Click **Save**.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

## Example: Using Profiler to Alert You to New Hosts and Port Activity

After you have created a baseline and installed an appropriate security policy, you can use Profiler to alert you when new hosts or applications appear in your network. You can analyze the alerts to decide whether to update your security policy.

To set alerts when Profiler detects new hosts or applications:

1. In the NSM Device Manager, right-click the IDP Series device and select **IDP Profiler > Start Profiler**.
2. Retain your baseline settings for general features, tracked hosts, and contexts.
3. Click the **Alert** tab and select options to generate alerts when Profiler detects new hosts, new protocols, or new ports.
4. Click **Apply**.

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

## Example: Identifying Services That Use Nonstandard Ports

Suppose you want to identify traffic that uses nonstandard ports so that you can take the appropriate security measures, such as physically removing the unauthorized network components, accounting for nonstandard ports in your existing corporate security policy, or creating rules in your security policy to restrict the traffic to specific network components.

To display a view of traffic that uses nonstandard ports:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler** to display the Profiler viewer.
2. Click the **Violation Viewer** tab.
3. Click the + icon that appears on the top of the right-hand window to display the New Permitted Object window.
4. For this example, name the new permitted object **Non-Standard-Ports**.
5. Right-click the Service column and select **Add Service**.
6. Select all predefined services.
7. Click **OK**.

After you have created and saved the permitted object, the object automatically becomes available in the Profiler.

8. Select the new permitted object **Non-Standard-Ports**.

The Profiler uses the object to filter the data collected from the devices. Traffic that matches the object (uses a standard service port) is filtered out, leaving only the traffic that does not match (uses a nonstandard service port).

9. Review the data for all traffic on your network that uses nonstandard service ports and take appropriate action.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

---

## Example: Responding to Vulnerability Announcements with Due Diligence

---

New network attacks and exploits are discovered every day. When new security patches are issued, use the Profiler to quickly identify which systems are running the affected software version, then patch them appropriately.

For large networks, it is difficult to patch everything immediately. Plan your patching process by prioritizing based on the importance of the resources. Critical, high-risk, and heavily used resources should be patched first, while less important, minimally used resources might be able to wait.

For example, suppose Microsoft announces a vulnerability in version 6.0 of the Microsoft Internet Information Services (IIS).

To quickly identify all network components running the vulnerable version:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler** to display the Profiler viewer.
2. Click the **Protocol Profiler** tab and review the Profiler logs keyed to protocols running on the network.
3. In the Context column, right-click a value and select **Edit Filters** to display the Context Filters dialog box.
4. Set a filter for HTTP Header Servers, for example.

The filtered view highlights the Web servers in your network. Suppose the table lists the following Web servers:

- Apache (two versions)
- Microsoft IIS, version 6.0

5. Select the Microsoft IIS 6.0 value to display your Microsoft IIS 6.0 destination server IP addresses.
6. Patch the vulnerable IIS server by using the information supplied with the Microsoft Security Bulletin.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

## Example: Using Profiler to Investigate Unanticipated Attacks

Suppose your corporate security policy does not permit SQL servers on the internal network. However, during a regular Microsoft update, SQL applications are installed on a network server, without your knowledge. Because you are not aware that an SQL server is running on your network, you have not configured security policy rules to block SQL attacks.

Suppose you receive a call informing you that the SQL Slammer worm attacks and infects your network.

To investigate:

1. Create a custom TCP service object to represent Microsoft SQL (default port: TCP/1433).
2. Restart the Profiler.
3. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler** to display the Profiler viewer.
4. Click the **Network Profiler** tab and review the source, destination, and service data.
5. Use a filter to display only records matching the SQL service object you created in Step 1.

The filtered view highlights the SQL servers in your network.

6. Take appropriate measures to secure the network, such as:
  - Applying patches.
  - Removing the components from your network.
  - Removing SQL server from all components.
  - Creating a rule in your security policy that drops all SQL connections between your internal network objects.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

---

## Example: Using Profiler to Mitigate Risks from Laptops

---

Suppose your corporate firewall denies RPC file sharing traffic to protect sensitive corporate files from Internet users, but enables RPC file sharing on a local network for convenience.

Suppose a laptop user has a good reason to use a partner's wireless network to access the Internet. Because the laptop is configured to allow RPC, it contracts a Blaster worm from an infected user on that network. When the user returns to the office and connects the laptop to the corporate network, the worm immediately begins scanning the internal network and infecting all components that have RPC enabled.

The Profiler records all unique activity on the network, so it identifies the ICMP packet scans as a new event. If you have configured the Profiler to send alerts for new hosts, you receive an alert that a new host has joined the network. In response to the alert, you check the Profiler viewer for details on the new host, and you learn that a host in your network is scanning the entire network using ICMP, a possible sign of the Blaster worm.

To investigate:

1. Restart the Profiler.
2. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler** to display the Profiler viewer.
3. Click the **Network Profiler** tab and review the source, destination, and service data.
4. Apply a filter to the Service column values so that only records matching **ICMP** are displayed.
5. Apply a second filter to the Access Type column so that only records matching **ICMP** and **probe** are displayed.
6. Apply a third filter to the Last Time column so that only records from the last two hours are displayed.

The Network Profiler displays all network components that used ICMP to probe the network in the last two hours.

Assuming the filters have cleared nonmatching records, you can now see the only IP address or IP addresses currently probing your network using ICMP. However, because



you use DHCP to dynamically assign IP addresses, you need to identify which user laptop is currently using that IP address.

7. Right-click the table cell for source properties to display the MAC address. If your enterprise maintains records matching MAC address to laptops, you can track down the specific host that is infected.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

## Example: Using NSM to Enable and View Application Volume Tracking

You can use NSM to enable application volume tracking (AVT) and to view AVT logs and reports.

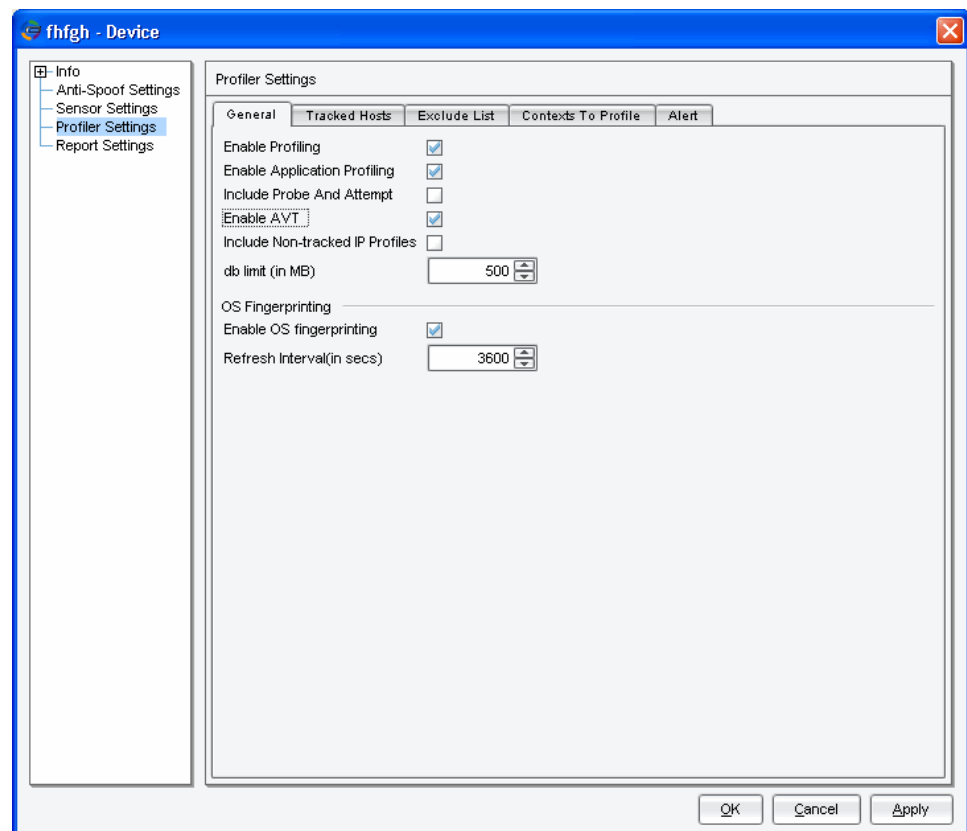
To enable AVT:

1. From NSM Device Manager, double-click a device and then click **Profiler Settings**.
2. Click the **General** tab.
3. Ensure **Enable AVT** is selected. This setting is enabled by default and shown in [Figure 31 on page 134](#).
4. If you have changed settings, click **Apply**.
5. Start the Profiler:
  - a. From the NSM main menu, select **Devices > IDP Profiler > Start Profiler**.
  - b. Select the devices on which you want to start the Profiler.
  - c. Click **OK**.



**NOTE:** If you change Profiler settings, you must push a configuration update to the device before the new settings take effect. From the Device Manager, right-click the device, select **Update Device**, select the **Restart IDP Profiler After Device Update** check box, and click **OK**.

Figure 31: Profiler Settings: Enable AVT



To view AVT logs:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Application Profiler** tab.

The Application Profiler tab displays application data. [Figure 32 on page 134](#) shows the Application Profiler tab.

Figure 32: Profiler Viewer: Application Profiler Tab

The screenshot shows the 'Profiler' window with the 'Application Profiler' tab selected. The table displays application data. The left sidebar shows a tree view of application categories, with 'Http' selected under 'Web'. The table has columns for Application Category, Bytes, Packets, Src IP, Dst IP, Dst Port, VLAN ID, Application, Byte Count, Packet Count, User, Role, First Time, Last Time, Dom, and Device.

Application Category	Bytes	Packets	Src IP	Dst IP	Dst Port	VLAN ID	Application	Byte Count	Packet Count	User	Role	First Time	Last Time	Dom	Device
Application	10.83 Mb	22,896	HTTP (1.03 Mb, 2,153)												
Unknown-application-category	9.51 Mb	19,887	Window...	128.241.220...	80	0	HTTP	222.34 Kb	329			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Unknown	9.51 Mb	19,887	Window...	204.160.122...	80	0	HTTP	188.69 Kb	342			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Web	1.03 Mb	2,153	Window...	204.160.122...	80	0	HTTP	150.56 Kb	295			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Http	1.03 Mb	2,153	Window...	207.46.26.20	80	0	HTTP	122.94 Kb	312			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Remote-access	50.75 kb	584	Window...	128.241.220...	80	0	HTTP	72.01 Kb	111			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Misc	14.61 kb	109	Window...	6.12.204.126	80	0	HTTP	55.79 Kb	97			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Node	8.83 kb	39	Window...	4.59.128.37	80	0	HTTP	52.42 Kb	76			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Nbname	5.79 kb	70	Window...	204.160.122...	80	0	HTTP	50.16 Kb	62			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Enterprise	12.75 kb	86	Window...	by2meg1204...	80	0	HTTP	30.49 Kb	111			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Infrastructure	12.75 kb	86	Window...	ct-in-1103 go...	80	0	HTTP	12.41 Kb	31			Mon Apr 06 ...	Mon Apr 06 ...	global	DP-NS-2...
Dns	6.98 kb	68													
Dhcp	5.77 kb	18													
Encryption	2.31 kb	24													
Ssl	2.31 kb	24													
Messaging	880	22													
Setup	880	22													
Peer-to-peer	678	11													
File-sharing	378	6													
Gnutella-udp	378	6													
Chat	300	5													

Extended applications, also called nested applications, are reported separately from HTTP results. [Figure 33 on page 135](#) shows the Application Profiler tab where results for the HTTP Google Earth application are distinguished from HTTP results.

**Figure 33: Profiler Viewer: Application Profiler Tab: Nested Applications**

Application Cate...	Bytes	Packets	Src IP	Dst IP	Dst Port	VLAN ID	Application	Byte Count	Packet Co...	User	Role
Application	99.22 Mb	76,036	<div>HTTP ( 62.14 Mb , 47,809 )</div>								
Web	99.22 Mb	76,036	<div> <div>vict1 att1 80 0</div> <div>HTTP 62.14 Mb 47,809</div> </div>								
Http	62.14 Mb	47,809	<div> <div>GOOGLE-EARTH ( 37.08 Mb , 28,227 )</div> </div>								
Google-earth	37.08 Mb	28,227	<div> <div>vict1 att1 80 0</div> <div>GOOGLE-EA... 37.08 Mb 28,227</div> </div>								

The Application Profiler view is divided into two sections:

- In the left pane, the Application Profiler tab displays a hierarchical tree of application categories. Applications are grouped by common functionality. For example, Peer-to-Peer applications include Chat and File Sharing applications. Under Chat, you can display Yahoo messenger, MSN, and AIM; under File Sharing, you can display Kazaa, Bittorrent, and Gnutella.

The left pane also displays aggregate statistics for volume (bytes) and packet count for the application category, application group, or application you select in the tree.

- In the right pane, the Application Profiler tab displays tables of session logs related to the application category or application you select in the left pane.

[Table 43 on page 135](#) describes the Application Profiler session table.

**Table 43: Application Profiler Session Table**

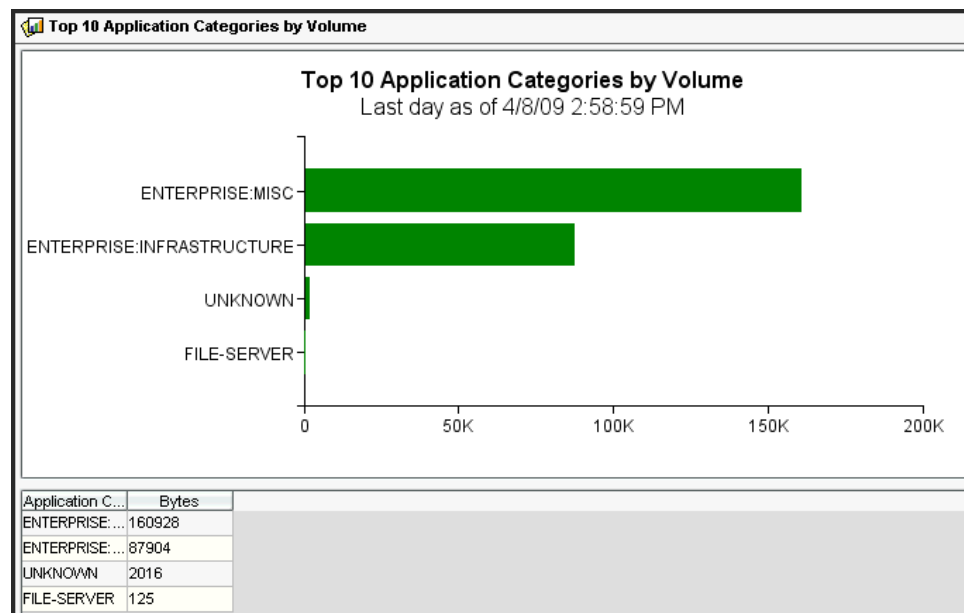
Column	Description
Src IP	Source IP address of the session.
Dst IP	Destination IP address.
VLAN ID	VLAN ID (if any).
Application ID	Application. The application identified by the application identification feature. Extended applications (also called nested applications) are reported separately from HTTP results. A 0 indicates the application was not identified.
Byte count	Byte count.
Packet count	Packet count.
User	The user associated with the session.
Role	The role to which the user belongs.
First Time	Timestamp for the first time the device logged the event (within the specified time interval).

Table 43: Application Profiler Session Table (*continued*)

Column	Description
Last Time	Timestamp for the last time the device logged the event (within the specified time interval).
Domain	NSM domain.
Device	Device through which the session was forwarded.

The Application Profiler tab displays application data. [Figure 34 on page 136](#) is an example of an NSM AVT report.

Figure 34: NSM AVT Report



**NOTE:** AVT reports are not real-time reports. On the local IDP Series device, the AVT processor writes an AVT log file at 15 minute intervals. NSM collects the interval data during its routine device log collection activity. As a result, there might be up to a 15 or 16 minute lag from the time a session is received by the IDP Series device and the display of the data in the NSM report.

To view AVT reports:

1. In the NSM navigation tree, select **Investigate > Report Manager > AVT Reports**.
2. Click the name of a predefined report to display it. [Table 44 on page 137](#) describes the predefined AVT reports.

**Table 44: NSM: Application Volume Tracking Reports**

Report	Description
Top 10 Applications by Volume	Applications with the highest volume in bytes in the past 24 hours.
Top 10 Application Categories by Volume	Application categories with the highest volume in bytes in the past 24 hours.
Top 5 Applications by Volume over Time (last hour)	Applications with the highest volume in bytes in the past hour.
Top 5 Application Categories by Volume (last hour)	Application categories with the highest volume in bytes in the past hour.
Top 5 Source by Volume over Time (last hour)	Source IP addresses with the highest volume in bytes in the past hour.
Top 5 Destination by Volume over Time (last hour)	Destination IP addresses with the highest volume in bytes in the past hour.

**Related Documentation**

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Application Volume Tracking Overview on page 22](#)
- [NSM Reports Overview on page 29](#)
- [IDP Reporter Overview on page 31](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)
- IDP Reporter Task Summary



## CHAPTER 16

# Logging

- [Example: Using NSM Log Viewer Features on page 139](#)
- [Example: Packet Logging Workflow on page 145](#)
- [Example: Querying the IDP Series Device MIB on page 151](#)

### Example: Using NSM Log Viewer Features

---

The Network and Security Manager (NSM) Log Viewer includes many display features to help you sort and correlate logs so you can analyze security events. For complete information on NSM Log Viewer features, see Chapter 18 of the [NSM Administration Guide](#). The following sections are provided here to give you ideas of how to take advantage of NSM features as you develop your approach to log monitoring:

- [Using Predefined Views on page 139](#)
- [Showing and Hiding Columns on page 140](#)
- [Using Filters on page 141](#)
- [Using Log Viewer Detail Panes on page 142](#)
- [Using Flags and Comments on page 143](#)
- [Using Custom Views on page 144](#)

### Using Predefined Views

Out of the box, the NSM Log Viewer includes a predefined view for DI/IDP event logs. A predefined view is a filtered view of all logs collected on the NSM device server. The DI/IDP view is filtered for events that match a predefined or custom attack object (Category field = Predefined or Category = Custom). [Figure 35 on page 140](#) shows the DI/IDP view.

Figure 35: NSM Log Viewer: Predefined View

**Log Viewer [3-IDP000]**

Log ID	Time Received	Alert	User	Src Addr	Dst Addr	Action	Protocol	Dst Port	Rule #	Nat Src Addr	Nat Dst Addr	Details
20090813/770	8/12/09 7:52:54 PM			1.0.0.199	2.0.0.23	Conn Dropped	TCP	1433	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/771	8/12/09 7:52:54 PM			1.0.0.22	2.0.0.115	Conn Dropped	TCP	1433	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/772	8/12/09 7:52:58 PM			1.0.0.139	2.0.0.179	Conn Dropped	TCP	12174	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/773	8/12/09 7:53:01 PM			1.0.0.148	2.0.0.248	Conn Dropped	TCP	22	1	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/774	8/12/09 7:53:01 PM			1.0.0.138	2.0.0.119	Conn Dropped	TCP	22	1	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/775	8/12/09 7:53:04 PM			1.0.0.1	2.0.0.69	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/776	8/12/09 7:53:04 PM			1.0.0.166	2.0.0.45	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/777	8/12/09 7:53:04 PM			1.0.0.28	2.0.0.97	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/778	8/12/09 7:53:04 PM			1.0.0.155	2.0.0.87	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/779	8/12/09 7:53:04 PM			1.0.0.43	2.0.0.238	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/780	8/12/09 7:53:04 PM			1.0.0.172	2.0.0.8	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/781	8/12/09 7:53:04 PM			1.0.0.167	2.0.0.20	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/782	8/12/09 7:53:04 PM			1.0.0.113	2.0.0.94	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/783	8/12/09 7:53:04 PM			1.0.0.230	2.0.0.49	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/784	8/12/09 7:53:04 PM			1.0.0.73	2.0.0.119	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'
20090813/785	8/12/09 7:53:04 PM			1.0.0.89	2.0.0.232	Conn Dropped	UDP	7787	2	0.0.0.0	0.0.0.0	'interface=eth2'

No more logs found

Timeline: Aug 6, Aug 7, Aug 8, Aug 9, Aug 10, Aug 11, Aug 12, Aug 13, Aug 14, Aug 15, Aug 16, Aug 17, Aug 18, Aug 19

Out 8/12/09 7:53:04 PM Tailing Logs

Summary: All Fields, Whois Lookup, Quick Report

**Predefined :: APP: Unreal Gamespy Query Protocol Buffer Overflow**

**References**  
This signature detects attempts to exploit a known vulnerability against the GameSpy query protocol supported by Unreal game engine. Attackers can crash a game server running the Unreal game engine, or execute arbitrary code with permissions of the user running the server.

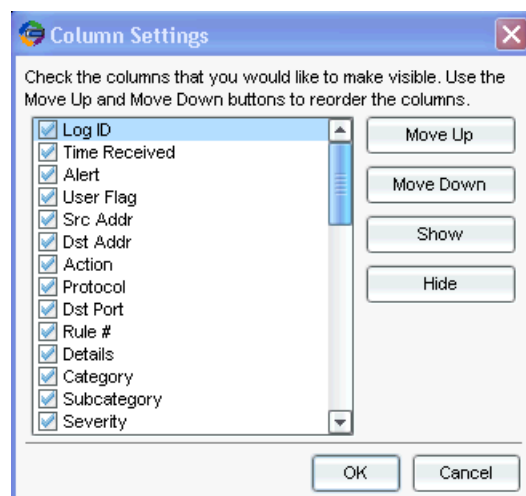
**Matching Data Snippet**

HEX ASCII

## Showing and Hiding Columns

The default columns shown in the predefined DI/IDP view might not include all of the data fields you are interested in. To select your preferred columns and the order in which they appear, select **View > Choose Columns** and use the dialog box to organize columns according to your preference.

Figure 36: NSM Log Viewer: Choose Columns

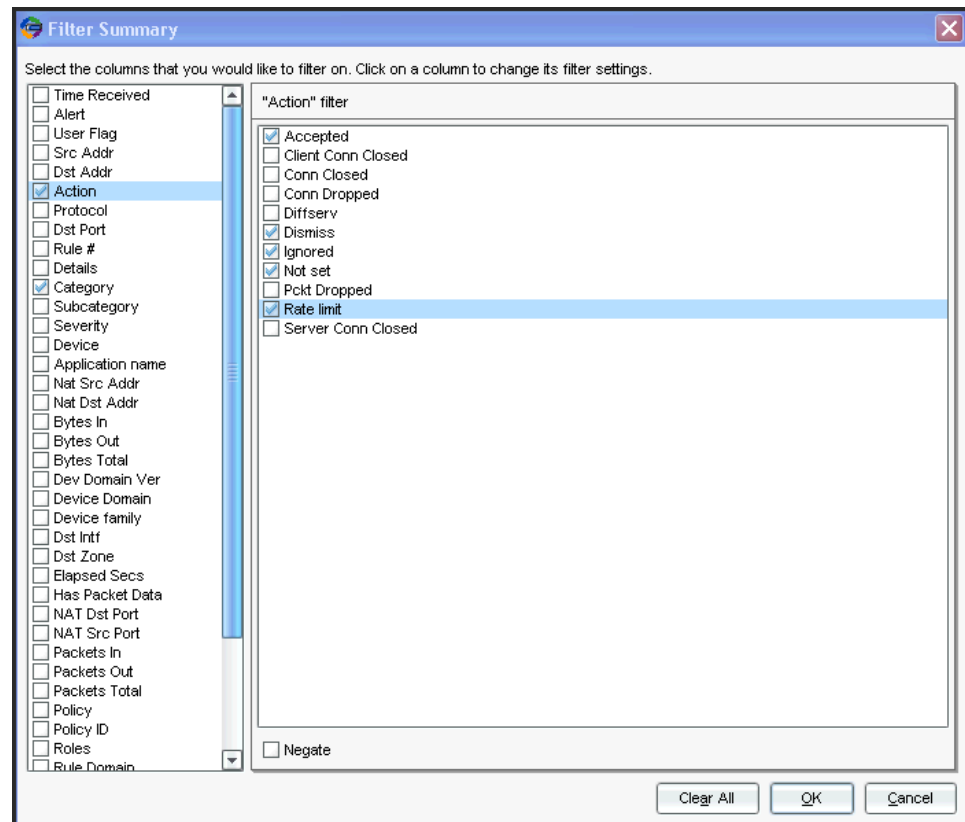




## Using Filters

The default DI/IDP view is filtered to display only logs where Category=Predefined or Category=Custom. To set additional filters, select **View > Filter Summary** and use the dialog box to set additional filters. In [Figure 37 on page 141](#), filters are selected to display logs for traffic where the rule action allowed the traffic continue to the destination server. When you approach the set of logs you examine each day, you might want to start with events of high severity, where traffic continued to the destination.

Figure 37: NSM Log Viewer: Filters



You can also filter on the fly. Suppose you find a log for an attack targeting HTTP traffic. In the row for the log, you can right-click the cell containing destination port 80 and select **Filter > Only This Value** to redisplay the table with only records where destination port = 80.

Figure 38: NSM Log Viewer: Filters

The screenshot shows the NSM Log Viewer interface with a log table and a 'Filter for Dst Port' dialog box. The log table has columns: Log ID, Time Received, Alert, User Flag, Comment, Src Addr, Dst Addr, Action, Protocol, Dst, Src, Inet Sr, Inet De, Details, Category, and Subcategory. The dialog box is titled 'Filter for Dst Port' and has fields for 'From' (80) and 'To' (80), with a 'Negate' checkbox and 'Clear', 'OK', and 'Cancel' buttons.

Log ID	Time Received	Alert	User Flag	Comment	Src Addr	Dst Addr	Action	Protocol	Dst	Sr	Inet Sr	Inet De	Details	Category	Subcategory
20090806473467	8/6/09 7:13:50 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473469	8/6/09 7:13:55 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473486	8/6/09 7:14:26 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473489	8/6/09 7:14:32 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473491	8/6/09 7:14:35 AM				1.0.0.99	2.0.0.99	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Predifined	HTTP: Missing HTTP Version
20090806473493	8/6/09 7:14:40 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473499	8/6/09 7:14:46 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473500	8/6/09 7:14:49 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473501	8/6/09 7:14:52 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473516	8/6/09 7:15:56 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473518	8/6/09 7:16:02 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473520	8/6/09 7:16:10 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473522	8/6/09 7:16:13 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473526	8/6/09 7:16:29 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473530	8/6/09 7:16:32 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473532	8/6/09 7:16:39 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473534	8/6/09 7:16:45 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473539	8/6/09 7:16:48 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473540	8/6/09 7:17:04 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473542	8/6/09 7:17:14 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473543	8/6/09 7:17:14 AM				1.0.0.99	2.0.0.99	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Predifined	HTTP: Missing HTTP Version
20090806473545	8/6/09 7:17:20 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473548	8/6/09 7:17:20 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473553	8/6/09 7:17:30 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806473557	8/6/09 7:17:36 AM				1.0.0.99	2.0.0.99	Rate limit	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept

## Using Log Viewer Detail Panes

The details pane below the log table provides summary and security reference information for the attack object that triggered the log. The details pane also includes a link to WHOIS information for the source IP.

Suppose your security policy rule includes the following attack object: Predifined :: HTTP: Windows Media Services NSISLog.DLL Buffer Overflow. It generates a log when it identifies the attack pattern in traffic through the IDP Series device. Use the reference information in the details pane below the log table to learn more about the attack. You can click the hypertext linked name of the attack object in the summary tab to display reference information for the attack, as shown in Figure 39 on page 142.

Figure 39: Using NSM Log Viewer Attack Reference Information

The screenshot shows the NSM Log Viewer interface with a log table and a 'Details' pane. The log table has columns: Log ID, Time Received, Alert, User Flag, Comment, Src Addr, Dst Addr, Action, Protocol, Dst, Src, Inet Sr, Inet De, Details, Category, and Subcategory. The 'Details' pane is titled 'HTTP: Windows Media Services NSISLog.DLL Buffer Overflow' and contains sections for 'References', 'Extended Description', 'Last Modified', 'Impact', 'Description', and 'Technical Information'.

Log ID	Time Received	Alert	User Flag	Comment	Src Addr	Dst Addr	Action	Protocol	Dst	Sr	Inet Sr	Inet De	Details	Category	Subcategory
20090806416941	8/5/09 11:13:33 PM				1.1.0.115	1.2.0.56	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Predifined	HTTP: IS cmd.exe Command Exec...
20090806416943	8/5/09 11:13:33 PM				1.1.0.192	1.2.0.102	Conn Dropped	TCP	80	3	0.0.0.0	0.0.0.0	Interface-eth2	Predifined	HTTP: ISS 0 WebDAV SEARCH Co...
20090806416945	8/5/09 11:13:33 PM				1.1.0.212	1.2.0.241	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416948	8/5/09 11:13:33 PM				1.1.0.248	1.2.0.132	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416949	8/5/09 11:13:36 PM				1.1.0.63	1.2.0.159	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416950	8/5/09 11:13:36 PM				1.1.0.88	1.2.0.56	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416951	8/5/09 11:13:36 PM				1.1.0.161	1.2.0.81	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416955	8/5/09 11:13:36 PM				1.1.0.231	1.2.0.243	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416957	8/5/09 11:13:36 PM				1.1.0.231	1.2.0.243	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416961	8/5/09 11:13:36 PM				1.1.0.241	1.2.0.121	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416966	8/5/09 11:13:42 PM				1.1.0.170	1.2.0.91	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416967	8/5/09 11:13:42 PM				1.1.0.88	1.2.0.56	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416971	8/5/09 11:13:45 PM				1.1.0.241	1.2.0.121	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806416972	8/5/09 11:13:45 PM				1.1.0.23	1.2.0.139	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417097	8/5/09 11:14:59 PM				1.1.0.188	1.2.0.231	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417098	8/5/09 11:17:37 PM				1.1.0.20	1.2.0.143	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417099	8/5/09 11:17:40 PM				1.1.0.199	1.2.0.227	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417100	8/5/09 11:17:40 PM				1.1.0.114	1.2.0.190	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417101	8/5/09 11:17:40 PM				1.1.0.234	1.2.0.123	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417102	8/5/09 11:17:40 PM				1.1.0.189	1.2.0.95	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417103	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417104	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417105	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417106	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417107	8/5/09 11:17:43 PM				1.1.0.205	1.2.0.103	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept
20090806417108	8/5/09 11:17:43 PM				1.1.0.109	1.2.0.55	Conn Dropped	TCP	80	2	0.0.0.0	0.0.0.0	Interface-eth2	Traffic	Accept

**Details Pane: HTTP: Windows Media Services NSISLog.DLL Buffer Overflow**

**References**

- <http://online.securityfocus.com/bid/3035/discussion/>
- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0349>
- <http://www.kb.cert.org/vuls/id/113716>
- <http://www.microsoft.com/technet/security/bulletin/MS03-022.mspx>
- <http://secunia.com/advisories/9115>

**Extended Description**

**Last Modified**  
2009-08-13

**Impact**  
Windows Media Services may expose IIS to remote arbitrary code execution if media logging is enabled.

**Description**  
Microsoft has reported a buffer overflow vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension handles incoming client requests. This could cause arbitrary code execution in IIS, which is exploitable through Media Services.

**Technical Information**  
Microsoft Media Services provides functionality for providing streaming media content to clients from IIS. It ships with a number of Microsoft Windows 2000 server releases and is also available for download for Windows NT. Microsoft has reported a buffer overflow vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension (nsislog.dll) handles incoming client requests. The logging facility may attempt to write excessive data to an undersized buffer when handling a malformed HTTP client request. This could trigger a denial of service or remote arbitrary code execution in IIS, which is exploitable through Media Services. The issue would occur in servers that are configured to provide logging of media requests. It is possible to exploit this issue by sending an overly long HTTP POST request to the

## Using Flags and Comments

As you work through logs, you can annotate them with flags and comments and then filter on your annotations. Figure 40 on page 143 shows a log marked as a false positive because the attack targets server versions not present in our network.

Figure 40: Using NSM Log Viewer Flag and Comment Features

The screenshot displays the NSM Log Viewer interface. At the top, there's a header bar with 'per Networks - NSM - global : current' and search/help options. Below is a 'Log Viewer [3-IDPODI]' section containing a table of log entries. The table has columns: Log ID, Time Received, Alert, User Flag, Comment, Src Addr, Dst Addr, Action, Protocol, Dst..., Nat Sr..., and Nat Ds... One log entry is highlighted in red, and a context menu is open over the 'Flag' column, showing options like High, Medium, Low, Closed, False Positive, Assigned, Investigate, Follow-Up, Pending, and Clear. Below the table is a timeline view showing dates from Jul 30 to Aug 7. At the bottom, there's a 'Summary' section with tabs for 'All Fields', 'Whois Lookup', and 'Quick Report'. The 'Quick Report' tab is selected, showing a 'Predefined :: HTTP: Windows Media Services NSISlog.DLL Buffer Overflow' signature and a 'Matching Data Snippet' section with a 'HEX' tab.

Log ID	Time Received	Alert	User Flag	Comment	Src Addr	Dst Addr	Action	Protocol	Dst...	Nat Sr...	Nat Ds...
20090806/416941	8/5/09 11:13:33 PM				1.1.0.115	1.2.0.58	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416943	8/5/09 11:13:33 PM				1.1.0.192	1.2.0.102	Conn Dropped	TCP	80	3	0.0.0.0
20090806/416945	8/5/09 11:13:33 PM				1.1.0.212	1.2.0.241	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416948	8/5/09 11:13:33 PM				1.1.0.248	1.2.0.132	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416949	8/5/09 11:13:36 PM				1.1.0.63	1.2.0.159	Conn Dropped	TCP	80	3	0.0.0.0
20090806/416950	8/5/09 11:13:36 PM				1.1.0.88	1.2.0.56	Conn Dropped	TCP	80	3	0.0.0.0
20090806/416951	8/5/09 11:13:36 PM				1.1.0.161	1.2.0.81	Conn Dropped	TCP	80	3	0.0.0.0
20090806/416956	8/5/09 11:13:36 PM				1.1.0.231	1.2.0.243	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416957	8/5/09 11:13:36 PM				1.1.0.231	1.2.0.243	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416961	8/5/09 11:13:36 PM				1.1.0.241	1.2.0.121	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416966	8/5/09 11:13:42 PM				1.1.0.170	1.2.0.91	Conn Dropped	TCP	80	4	0.0.0.0
20090806/416967	8/5/09 11:13:42 PM			Windows 2000 SP4 only?	1.1.0.88	1.2.0.56	Conn Dropped	TCP	80	3	0.0.0.0
20090806/416971	8/5/09 11:13:45 PM				1.1.0.241	1.2.0.121	Conn Dropped	TCP	80	3	0.0.0.0
20090806/416972	8/5/09 11:13:45 PM				1.1.0.23	1.2.0.139	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417097	8/5/09 11:14:59 PM				1.1.0.188	1.2.0.231	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417098	8/5/09 11:17:37 PM				1.1.0.20	1.2.0.143	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417099	8/5/09 11:17:40 PM				1.1.0.199	1.2.0.227	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417100	8/5/09 11:17:40 PM				1.1.0.114	1.2.0.190	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417101	8/5/09 11:17:40 PM				1.1.0.234	1.2.0.123	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417102	8/5/09 11:17:40 PM				1.1.0.189	1.2.0.95	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417103	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417104	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417105	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417106	8/5/09 11:17:40 PM				1.1.0.48	1.2.0.155	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417107	8/5/09 11:17:43 PM				1.1.0.205	1.2.0.103	Conn Dropped	TCP	80	4	0.0.0.0
20090806/417108	8/5/09 11:17:43 PM				1.1.0.109	1.2.0.55	Conn Dropped	TCP	80	4	0.0.0.0

To mark a log with a flag, right-click the cell in the Flag column and select one of the following flags:

- High (severity)
- Medium (severity)
- Low (severity)
- Closed
- False Positive

- Assigned
- Investigate
- Follow-Up
- Pending

## Using Custom Views

As you become familiar with NSM Log Viewer filters, you are likely to discover views of the data you typically want to use to monitor traffic. You can save custom views. Because the custom view is based on filters, incoming log entries that match the filter criteria are automatically displayed in the view. You do not need to reapply the view to new logs.

Figure 41 on page 144 shows a custom view of columns and filters focusing on events where the IDP Series device allowed HTTP traffic to proceed to its destination.

Figure 41: NSM Log Viewer: Custom View

[illegible]

You might want to create views to help manage the following example cases:

- **Workflow**—If your team distributes responsibilities based on IDP Series device, internal servers, application, severity, or type of attack, you can create views filtered on the appropriate columns. In the same manner, you can also use the Flag or Comments columns to prioritize or delegate investigation.
- **Attackers**—Once you learn the IP address of an attacker, you can create a view filtered on Source IP to watch what the attackers activities on your network.

- **Devices**—After you deploy a new device, you can create a view filtered on the Device column to observe and validate device effectiveness.

To create a new view, select the columns you want to display and apply filters. Select **File > New View** to display a dialog box to save the view in your preferred Log Viewer folder. We recommend saving custom views in the Custom folder.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)
- [NSM Reports Overview on page 29](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## Example: Packet Logging Workflow

This topic summarizes IDP Series packet logging basics. It includes the following sections:

- [Using Packet Captures on page 145](#)
- [Enabling Packet Capture in Security Policy Rules on page 145](#)
- [Forwarding Packet Capture Logs to NSM on page 146](#)
- [Viewing Packet Capture Logs on page 147](#)

### Using Packet Captures

The IDP solution supports packet capture logging triggered by security policy rules.

You can use packet captures for a number of response activities, including:

- Validation of the security policy rule and attack object. You may choose to enable packet logging to test a new attack object. Once verified, you may find packet logging for the rule unnecessary.
- Further analysis of traffic surrounding the matching event. The surrounding traffic might provide information that helps you determine whether you need to take further steps to protect the target or whether the attack should be considered a false positive.
- Reproducibility and documentation for Internet security groups, including the Juniper Networks Security Center.
- Legal evidence. Consult with your legal counsel for guidance on how local laws and rules of evidence apply if you want to use packet capture data as evidence in the prosecution of attackers.

### Enabling Packet Capture in Security Policy Rules

When traffic matches a rule where packet logging is configured, the IDP Series device captures the packet that matched the rule, as well as the preceding and trailing packets (according to your configured preference).

To enable packet logging within a security policy rule, use the Security Policy editor. Right-click a cell in the Notification column and select **Configure** to display the dialog box where you can set packet logging options.

Figure 42: Notification Options: Packet Logging



In the NSM Log Viewer, logs for events where packet captures have been generated are noted by an icon in the Has Packet Data column (the last column in Figure 43 on page 146).

Figure 43: NSM Log Viewer: Has Packet Data Column

Src Addr	Dst Addr	Action	Protocol	Port	Rule #	Net Src Addr	Net Dst Addr	Details	Category	Subcategory	Severity	Device	Comment	Has Packet Data
1.1.0.68	1.2.0.40	Conn Dropped	TCP	554	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	RTSP: Real Server Describe Overl...	Major	DP8202		
1.1.0.26	1.2.0.26	Conn Dropped	TCP	554	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	RTSP: Real Server Transport Over...	Major	DP8202		
1.1.0.206	1.2.0.111	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	HTTP: Cisco IOS HTTP Configuratio...	Major	DP8202		
1.1.0.56	1.2.0.34	Conn Dropped	TCP	25	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SMTP: Email Domain Name	Major	DP8202		
1.1.0.56	1.2.0.34	Conn Dropped	TCP	25	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SMTP: Email Address	Major	DP8202		
1.1.0.48	1.2.0.159	Conn Dropped	TCP	25	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SMTP: Email Address	Major	DP8202		
1.1.0.30	1.2.0.175	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	HTTP: Cisco IOS HTTP Configuratio...	Major	DP8202		
1.1.0.110	1.2.0.192	Conn Dropped	TCP	25	3	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SMTP: Microsoft Exchange Mailbox	Device_critical_log	DP8202		
1.1.0.118	1.2.0.198	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: VUL-FTPD (plogb) Input Valid...	Major	DP8202		
1.1.0.230	1.2.0.123	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Username Too Long	Major	DP8202		
1.1.0.62	1.2.0.168	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Username Too Long	Major	DP8202		
1.1.0.231	1.2.0.116	Conn Dropped	TCP	25	3	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SMTP: Exchange Multiple Long Mail...	Device_critical_log	DP8202		
1.1.0.4	1.2.0.133	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Username Too Long	Major	DP8202		
1.1.0.87	1.2.0.161	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SHELLCODE: X86 NCCOP (TCP)	Major	DP8202		
1.1.0.81	1.2.0.168	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Username Too Long	Major	DP8202		
1.1.0.150	1.2.0.208	Conn Dropped	TCP	25	3	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	SMTP: Sendmail Oversized Address...	Device_critical_log	DP8202		
1.1.0.231	1.2.0.116	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Username Too Long	Major	DP8202		
1.1.0.112	1.2.0.193	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Ipswitch WS_FTP Server FTP...	Major	DP8202		
1.1.0.224	1.2.0.122	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Pathname Too Long	Major	DP8202		
1.1.0.17	1.2.0.136	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Username Too Long	Major	DP8202		
1.1.0.224	1.2.0.122	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	FTP: Pathname Too Long	Major	DP8202		
1.1.0.223	1.2.0.239	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	HTTP: Missing HTTP Version	Major	DP8202		
1.1.0.98	1.2.0.184	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	HTTP: Microsoft VSM/Who Buffer Ov...	Major	DP8202		
1.1.0.98	1.2.0.184	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	HTTP: Missing HTTP Version	Major	DP8202		
1.1.0.56	1.2.0.163	Conn Dropped	TCP	119	3	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	NATP: XPAT Pattern Overflow	Device_critical_log	DP8202		
1.1.0.222	1.2.0.244	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2	Preddefined	HTTP: All-N WebAdmin USER Buff...	Major	DP8202		

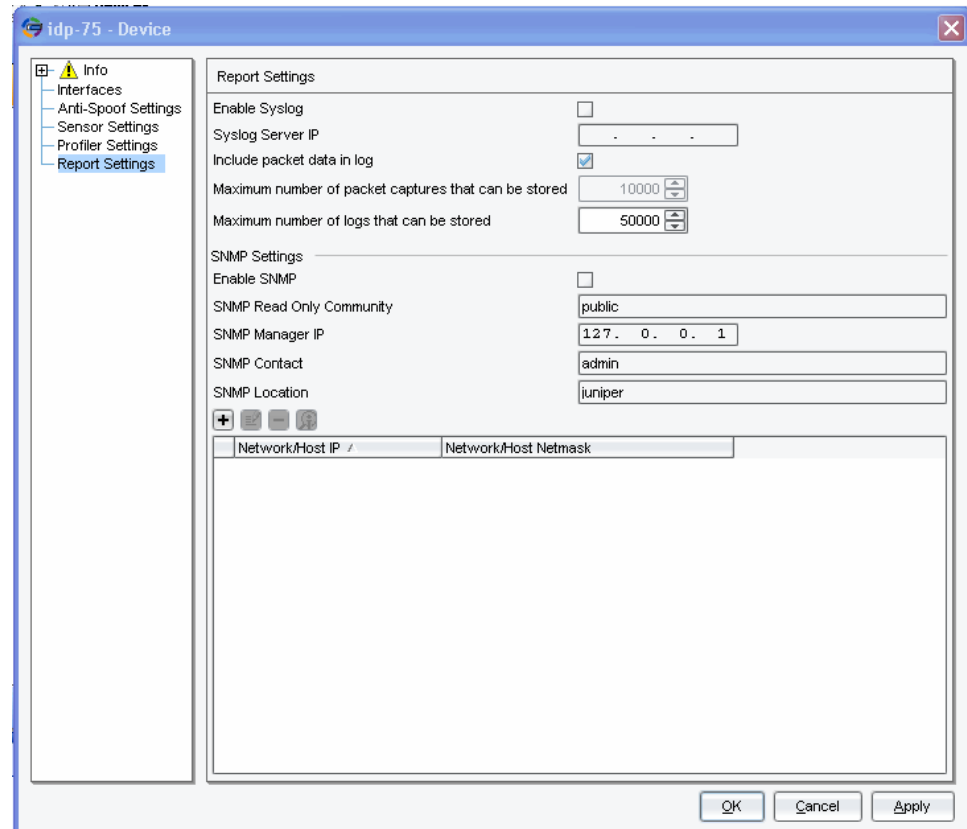
## Forwarding Packet Capture Logs to NSM

The IDP Series device writes packet captures locally to subdirectories of `/usr/idp/device/var/pktlogs/`. It forwards the packet data to NSM according to your NSM Report Settings:

- **Include packet data in log** selected. Forwards the packet capture to NSM automatically whenever it sends the corresponding event log.

- **Include packet data in log** not selected. Forwards a reference to the packet capture file to NSM automatically but forwards the packet data itself only on-demand (when an NSM user takes action to display the packet data).

Figure 44: NSM Device Configuration Editor: Report Settings



## Viewing Packet Capture Logs

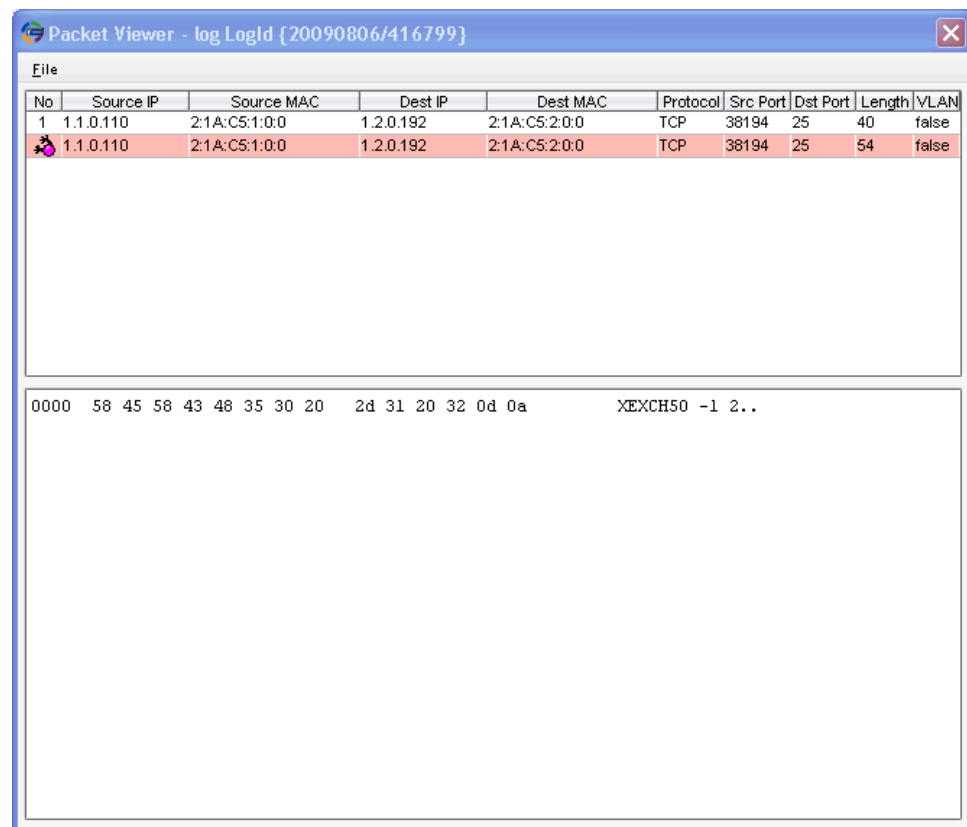
You have two options for viewing packet captures:

- [Using the NSM Packet Viewer on page 147](#)
- [Using an External Viewer to View Packet Data on page 148](#)

### Using the NSM Packet Viewer

The NSM packet viewer displays the offending attack payload that triggered the alert as well as preceding and trailing packets (according to your configuration). [Figure 45 on page 148](#) shows the NSM packet capture viewer.

Figure 45: NSM Packet Capture Viewer



To view a packet capture in the NSM packet viewer:

1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined > DI/IDP** to display the IDP table.
2. Select **View > Choose Columns** to display the dialog box you use to show and hide log table columns.
3. Select **Has Packet Data** to show this column.  
If a security event log has packet data, an icon appears in the table cell under this column.
4. Double-click the Has Packet data icon to display the packet data in the NSM packet viewer.

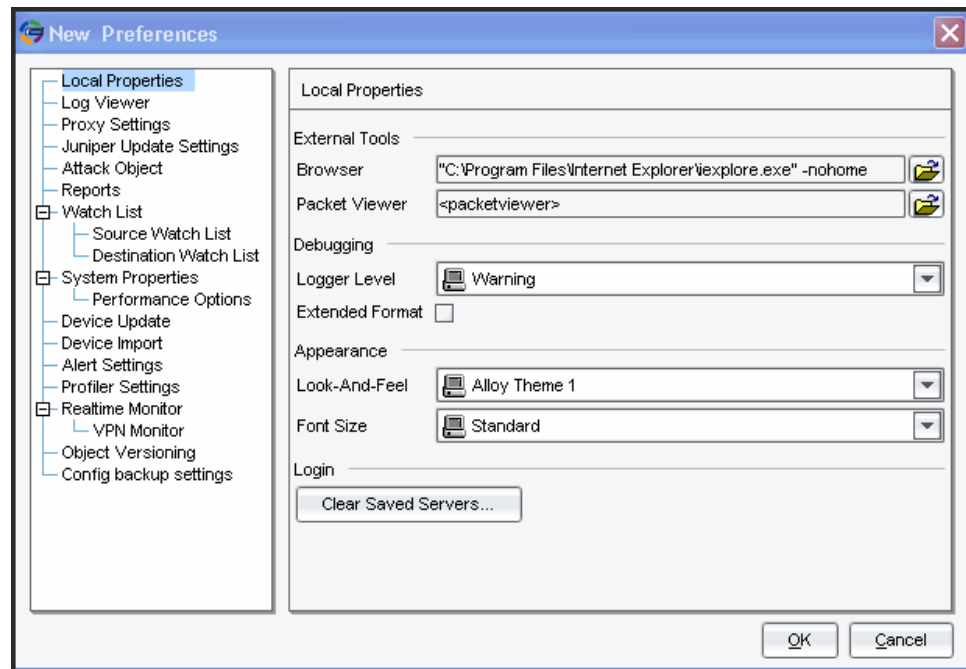
### Using an External Viewer to View Packet Data

You can configure NSM to launch an external viewer for packet captures.

Figure 46 on page 149 shows the NSM dialog box where you can specify the location of an external packet viewer.



Figure 46: Specifying an External Viewer

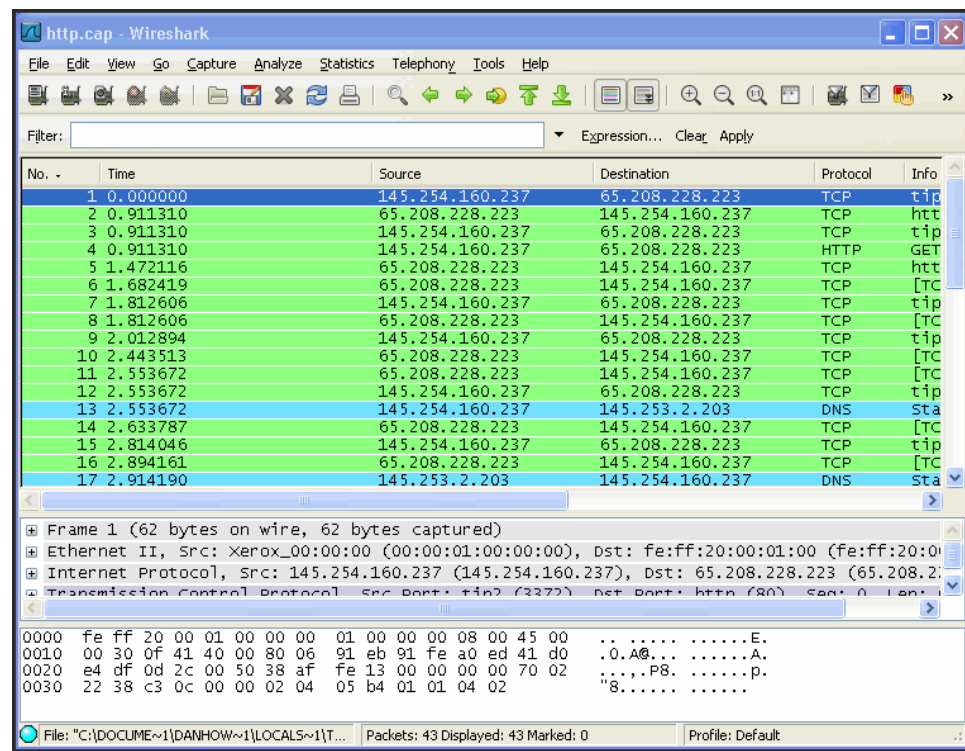


To set the location of the external viewer:

1. In NSM, select **Tools > Preferences**.
2. Select **Local Properties**.
3. Under External Tools > Packet Viewer, click the browse button and select the executable file for the external viewer (for example: **C:\Program Files\Wireshark\wireshark.exe**).
4. Click **OK** to close the New Preferences dialog box.

Figure 47 on page 150 shows packet data displayed in the Wireshark packet viewer.

Figure 47: Wireshark Packet Viewer



To view a packet capture in an external packet viewer:

1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined > DI/IDP** to display the IDP table.
2. Select **View > Choose Columns** to display the dialog box you use to show and hide log table columns.
3. Select **Has Packet Data** to show this column.  
If a security event log has packet data, an icon appears in the table cell under this column.
4. Right-click the Has Packet data icon and select **Show > Packet Data in External Viewer**.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Notification Options on page 65](#)
- [Using tcpdump to Capture Packets on page 481](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Specifying Rule Notification Options \(NSM Procedure\) on page 221](#)
- [Enabling Collection of Packet Data in NSM Logs \(NSM Procedure\) on page 303](#)

## Example: Querying the IDP Series Device MIB

You can use the reference tables in “IDP Series MIB Object ID Reference” on page 553 to help you understand reports retrieved by your MIB reader. You can also use these tables to look up the OID you want to query with an **snmpget** or **snmptable** utility.

The IDP OS image includes the most widely used, freely distributable SNMP command-line utilities available from <http://www.net-snmp.org/>. The following examples show **snmpget** and **snmptable** queries that specify OID by name and by number.

**Example: OID by Name** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorCpuUsageOneMin.0**  
JUNIPER-IDP-MIB::jnxIdpSensorCpuUsageOneMin.0 = Gauge32: 45

**Example: OID by Number** [host]# **snmpget -v2c -c public localhost 1.3.6.1.4.1.2636.3.9.1.10.0**  
SNMPv2-SMI::enterprises.2636.3.9.1.10.0 = Gauge32: 53

**Example: OID by Name** [host]# **snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropPerIntfcTable**  
SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropPerIntfcTable

jnxIdpSensorIFTable6Index	jnxIdpSensorTxdIntfcName	jnxIdpSensorTxPktsDropped
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth3"	0
7	"eth2"	0
8	"eth5"	0
9	"eth4"	0

**Example: OID by Number** [host]# **snmptable -v2c -c public localhost -m JUNIPER-IDP-MIB 1.3.6.1.4.1.2636.3.9.1.59**  
SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropPerIntfcTable

jnxIdpSensorIFTable6Index	jnxIdpSensorTxdIntfcName	jnxIdpSensorTxPktsDropped
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth3"	0
7	"eth2"	0
8	"eth5"	0
9	"eth4"	0



**NOTE:** You must use the **-m** option to reference the MIB file when you specify a numeric OID with **snmptable**.

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)
- [IDP Series MIB Object ID Reference on page 553](#)



## IDP Rulebase Examples

- [IDP Rulebase Example: Using Application Identification on page 153](#)
- [IDP Rulebase Example: Specifying the Default Service on page 154](#)
- [IDP Rulebase Example: Using Recommended Attack Objects on page 155](#)
- [IDP Rulebase Example: Using Recommended Actions on page 156](#)
- [IDP Rulebase Example: User-Role-Based Policies on page 157](#)
- [Example: Fine-Tuning a Security Policy on page 160](#)

### IDP Rulebase Example: Using Application Identification

This example demonstrates the usefulness of the application identification feature.

Suppose your corporate security policy changes, and you are charged with inspecting peer-to-peer traffic from applications such as Kazaa, Torrent, or eDonkey. To add new rules that inspect peer-to-peer traffic, you would take the following steps:

1. Analyze network traffic to identify peer-to-peer applications running in your network.
2. Research and identify the pattern for every peer-to-peer application.
3. Create signature and port definitions for every peer-to-peer application.
4. Verify the effectiveness of the signatures.
5. Repeat these steps several times for each peer-to-peer application.
6. Continually monitor the network for peer-to-peer traffic that uses nonstandard ports so you can update your signature set to inspect traffic over these ports.

Juniper Networks Security Center saves you much of this work. With predefined attack objects and application identification enabled, you can create a rule whose only elements are the predefined attack object and the service match set to **Default**.

[Figure 48 on page 154](#) is an example of a simple rule that detects any peer-to-peer application.

Figure 48: A Simplified Rule Enabled by the Application Identification Feature

IDP

APE

+

-

	No.	Match			Look For	Action	IP Action	Notification	Comments
		Source	Destination	Service	Attacks				
⚠	2	any	any	Default	[Recommended]P2P	Recommended	None	Logging	Client to Server P2P threats.

When the IDP engine identifies a source/destination/Default service match, it examines the session against the application signatures to determine the application, regardless of which port is used. The IDP system then decodes the traffic and inspects it for the attack objects related to that application.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Identification on page 43](#)
- [Understanding the IDP Rulebase on page 55](#)

## IDP Rulebase Example: Specifying the Default Service

This example demonstrates the usefulness of specifying the value **Default** for the service match parameter in IDP rulebase rules.

When you specify a service, you have the option to specify:

- A service object
- Any
- Default

If you specify the value **Default**, the rule gets the service parameter from the attack object. For example, if the attack object service binding specifies FTP, and you specify the value **Default** for service, the match value is FTP.

[Figure 49 on page 154](#) is an example of a rule where the default service resolves to FTP.

Figure 49: Default Service

IDP

No.	ID	Match				Look For	Action	IP Action	Notification
		Source	Destination	Service	Terminate Ma.	Attacks			
<div>⚠</div> <div>—</div> <div>📄</div> <div>1</div>	3	<div>🌐</div> <div>any</div>	<div>🌐</div> <div>any</div>	<div>🔌</div> <div>Default</div>	<div>☐</div>	<div>🏃</div> <div>[Recommended]FTP</div>	<div>🌿</div> <div>Recommended</div>	<div>🌿</div> <div>None</div>	<div>📄</div> <div>Logging</div>



**TIP:** With application Identification enabled, the IDP process engine identifies services even if they are running on nonstandard ports.

If you disable application identification and specify **Default**, the IDP process engine assumes that standard ports are used for the service.



**NOTE:** If you do not enable application identification and your service uses nonstandard ports, you must create a custom service object. For procedures, see the NSM documentation.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [IDP Rulebase Example: Using Application Identification on page 153](#)
- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [Understanding the IDP Rulebase on page 55](#)

## IDP Rulebase Example: Using Recommended Attack Objects

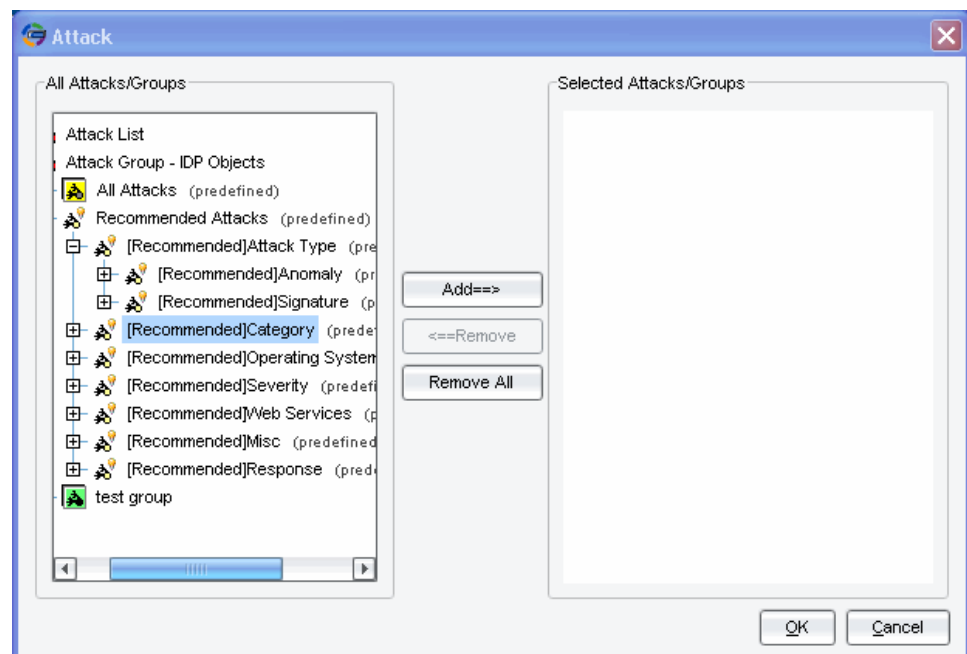
This example demonstrates the usefulness of Juniper Networks Security Center (J-Security Center) recommended attack objects.

When you add attack objects to an IDP rulebase rule, you have the option of adding:

- Predefined attack objects by group
- Recommended predefined attack objects by group
- Custom attacks

[Figure 50 on page 155](#) shows recommended attack objects in the dialog box for adding attack objects to the IDP rulebase.

**Figure 50: Recommended Attack Objects**



The groups marked **Recommended** have the following special features:

- Recommended attacks have been identified and coded for their recommended purpose by J-Security Center, a world class team of security experts.
- Recommended attack groups are dynamic groups, so members are added or deleted as appropriate during NSM attack database updates.

When you get started with an IDP Series deployment, you should use the recommended attack objects and enable notification for rule matches. Later, you can turn off logging (at your discretion). If you find you need to customize attack object properties, you can make a copy of the recommended attack object and modify it with your required properties. Then you can replace the recommended attack object with the custom attack object in your IDP rulebase rule.



**NOTE:** If you use a recommended attack object as the basis for a custom attack object, be sure to view the original attack object from time-to-time after attack database updates. If J-Security Center makes changes to the original, you must manually propagate changes to your custom attack object.

---



**BEST PRACTICE:** Each attack object specified in an IDP rulebase rule has a performance cost. We recommend that your rules include only the attack objects that are applicable to the rule destination server and only those of a severity that concerns you. We also recommend that you create more rules with a few attack objects in each rather than fewer rules with many attack objects.

---

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*.

- [Using Attack Objects on page 60](#)
- [J-Security Center Updates Overview on page 19](#)
- [Understanding the IDP Rulebase on page 55](#)

---

## IDP Rulebase Example: Using Recommended Actions

---

This example demonstrates the usefulness of Juniper Networks Security Center (J-Security Center) recommended actions.

When you specify a rule action, you have the option to specify:

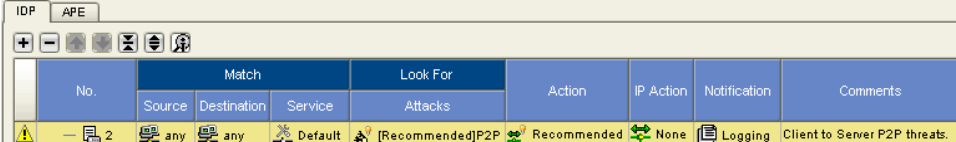
- No action
- A specific action
- The value **Recommended**



Recommended actions are coded in the predefined attack object by the J-Security Center team. The J-Security Center team codes a recommended action in all predefined attack objects, not just the recommended attack objects. When you use the recommended action, you leverage the experience and expertise of the J-Security Center team.

Figure 51 on page 157 shows an IDP rulebase rule with action set to **Recommended**.

Figure 51: Recommended Action



No.	Match			Look For	Action	IP Action	Notification	Comments
	Source	Destination	Service	Attacks				
2	any	any	Default	[Recommended]P2P	Recommended	None	Logging	Client to Server P2P threats.

When you update the NSM attack database, any changes to recommended actions are also automatically updated.

When you get started with an IDP Series deployment, you should use the recommended actions and enable notification for rule matches. If you find these settings meet your needs, you can turn off logging (at your discretion). If you find you prefer a different action, you can specify a different action.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*.

- [Understanding IDP Rulebase Actions on page 63](#)
- [Understanding the IDP Rulebase on page 55](#)

## IDP Rulebase Example: User-Role-Based Policies

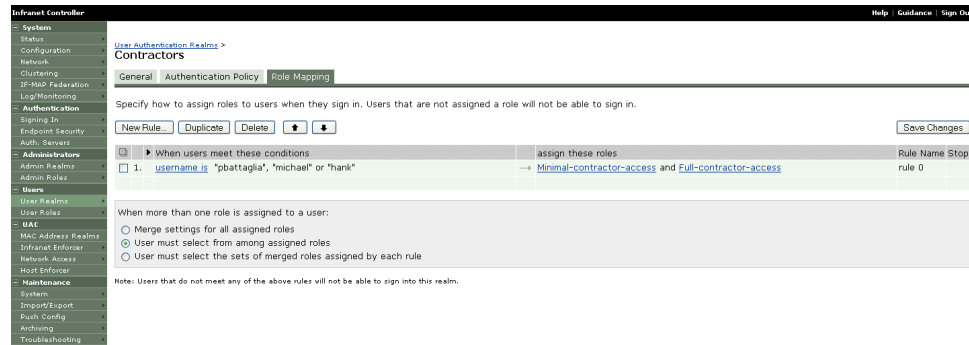
Suppose your enterprise uses Juniper Networks Unified Access Control (UAC) to authenticate access to the corporate network. When you initially rolled out the solution, Host Checker quarantined and denied network access to many users with noncompliant systems, and you received a lot of negative feedback about end user inconvenience and lost productivity. You can ameliorate these concerns when you deploy the IDP Series device with user session signaling from UAC. When the IDP Series device is protecting your network, users who were formerly flagged for quarantine because Host Checker identified vulnerabilities do not need to be denied access. With role-based IDP security policies, you can adopt a remediation plan that allows access, and even if the vulnerability has been exploited, your network will be protected by the IDP role-focused security policy.

To deploy this solution, follow these basic steps:

1. Read the release notes for the IDP Series device and the IC Series device to verify version compatibility requirements.
2. Deploy a UAC solution for user access to the network. For details, see the *Unified Access Control Administration Guide*.
3. Use UAC to create roles you want to use in your security policy. For security rules, you want to leverage results of the Host Checker to map users with vulnerable systems to roles that identify the vulnerabilities, such as "Laptop Users," "Unauthorized Instant

Messenger Installed,” or “Windows XP Patch Required.” [Figure 52 on page 158](#) shows the IC Series Admin Console Role Mapping page.

**Figure 52: IC Series Admin Console: Configuring User Roles**



For details on configuring roles and role mapping, see the *Unified Access Control Administration Guide* or UAC online Help.

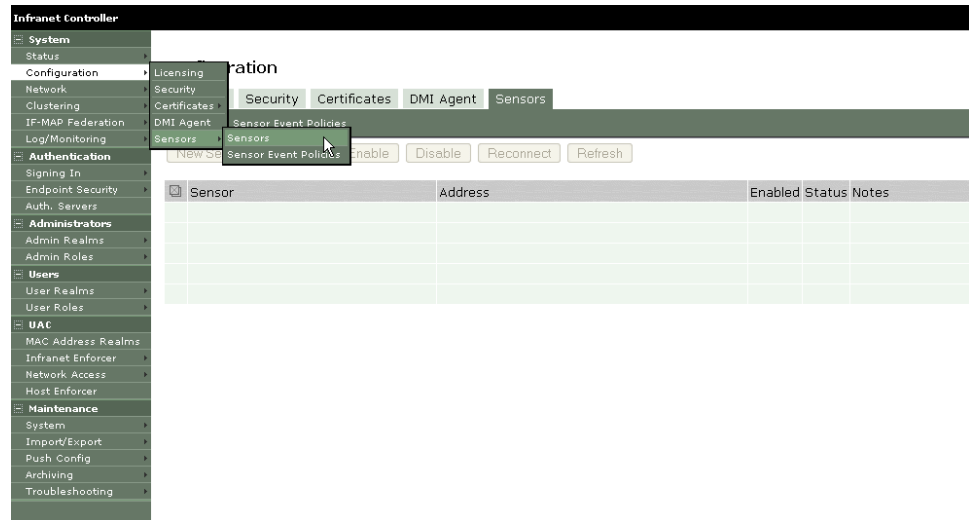
4. Configure communication between the IC Series device and the IDP Series device so you can use the IDP user-role-based policy feature:

- From the IDP Series side, you use the Appliance Configuration Manager (ACM) to generate a one-time password the IC Series device will use to connect to the IDP Series device. [Figure 53 on page 158](#) shows the ACM page used to generate a password for the IC Series connection.

**Figure 53: ACM: Generating a One-Time Password for the Connection from the IC Series Appliance**

- From the IC Series side, you configure the connection to the IDP Series device, specifying the IP address, port 7103, and the one-time password. [Figure 54 on page 159](#) shows the IC Series Admin Console Sensor Configuration page.

Figure 54: IC Series Admin Console: Configuring the Connection to the IDP Appliance



For details, see the UAC online Help.

5. In NSM, configure IDP rulebase rules that inspect traffic from users with vulnerable systems. Push the security policy to the IDP Series device.

Figure 55 on page 159 shows a rule where the IDP Series device inspects traffic from vulnerable hosts for the relevant Recommended attack objects.

Figure 55: IDP Rulebase: User-Role-Based Rules

No.	ID	Source	User Role	Destination	Service	Look For	Action	IP Action	Notification
4	1	any	Laptop Users	any	Default	[Recommended]SPYWARE	Recommended	IP Block	Logging
5	5	any	Unauthorized Instant Messenger Installed	any	Default	[Recommended]CHAT	Recommended	IP Notify	Logging
6	6	any	Windows XP Patch Required	any	Default	[Recommended]Windows	Recommended	IP Notify	Logging

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [User-Role-Based Policy Feature Overview on page 58](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying Integration with an IC Series Unified Access Control Appliance on page 493](#)
- [Configuring Advanced Settings for the User-Role-Based Policy Feature on page 322](#)

The following related topic is included in the *IDP Series Deployment Scenarios*:

- [Deploying IDP Series with an IC Series Device to Implement User-Role-Based Security Policies](#)

## Example: Fine-Tuning a Security Policy

---

This topic provides a suggested workflow for getting started and fine-tuning a security policy. It includes the following subtopics:

- [Fine-Tuning Security Policies Process Overview on page 160](#)
- [Getting Started with the Recommended Security Policy on page 160](#)
- [Refining Rule Matching Properties on page 161](#)
- [Reducing False Positives on page 162](#)
- [Adding Rulebases on page 164](#)

### Fine-Tuning Security Policies Process Overview

You want to tune a security policy so that it is:

- Comprehensive—Detects all possible attacks on specific hosts in your network.
- Optimized—Each attack object specified in an IDP rulebase rule has a performance cost. In general, you want more rules with a few attack objects in each rather than fewer rules with many attack objects. In addition, we recommend that a single rule includes only the attack objects that are applicable to the rule destination server and only those of a severity that concerns you.
- Precise—Generates few false positives.
- Maintainable—As you refine your rules, you want to leverage as much of the predefined logic as possible. In your IDP rulebase rules, for example, you want to use the application identification feature, dynamic attack object groups, recommended attack objects, and recommended actions as much as possible, knowing the Juniper Networks Security Center team updates these as needed (even daily).

Fine-tuning is an iterative process. The process involves the following steps:

1. [Getting Started with the Recommended Security Policy on page 49](#)
2. [Refining Rule Matching Properties on page 50](#)
3. [Reducing False Positives on page 50](#)
4. [Adding Rulebases on page 53](#)

### Getting Started with the Recommended Security Policy

When you add the IDP Series device to the NSM Device Manager, NSM automatically pushes the recommended policy to the IDP Series device. The recommended policy protects destination servers from the most frequent and severe attacks.

[Table 18 on page 49](#) summarizes the properties of the Recommended security policy.

Table 45: Recommended Security Policy Definition

Property	Value
Rulebase	IDP rulebase
Rules	Nine rules, distinguished by attack object
Source	Any
Destination	Any
Service	Default, meaning the matching property is based on the service bindings of the attack object specified by the rule
Attack objects	Recommended IP, Recommended TCP, Recommended ICMP, Recommended HTTP, Recommended SMTP, Recommended DNS, Recommended FTP, Recommended POP3, Recommended IMAP, Recommended Trojan, Recommended Virus, Recommended Worm  <b>NOTE:</b> All of the attack objects included in the predefined policies are client-to-server attacks.
Action	Recommended, meaning the action is specified by the attack object
Notification	Logging

## Refining Rule Matching Properties

The source and destination matching parameters for template rules are set to **Any**. These broad settings provide comprehensive protection, but they entail a performance cost and might result in more logs than are necessary. We recommend you customize these settings.

Run the Profiler for several days to gather information about the hosts and applications running in your network. After several days, you should have the data you need to complete the following tasks:

- Create NSM address objects that identify groups of internal servers. When you configure rules to examine client to server traffic, you specify the internal servers as the rule's destination servers.
- Create an address object that defines your internal network. When you configure rules to examine traffic from your network to hosts on the world wide web, you can specify the internal network address object as the rule's source.
- Create NSM service objects to identify services running on internal servers. In most cases, you can specify **Default** for service so the rule uses the service relevant to the attack object. However, there might be cases where you want to specify a service object or service group.

- Identify predefined attack groups related to services (or create your own attack group, if necessary).
- Refine the IDP rulebase rule set so that each rule is focused on a single destination server (client to server traffic) or service (server to client traffic).

## Reducing False Positives

A *false positive* is a log record that reflects an event on your network that you are not concerned about and no longer want to see in your logs. The IDP security policy found traffic that matched your rule, but over time you realize you do not need to track such events.

To determine whether a log is a false positive, you need to understand why the IDP Series device triggered the log and whether or not the traffic poses a real risk to the target server.

Suppose your security policy rule includes the following attack object: Predefined :: HTTP: Windows Media Services NSISlog.DLL Buffer Overflow. It generates a log when it identifies the attack pattern in traffic through the IDP Series device. Use the reference information in the details pane below the log table to learn more about the attack. You can click the hypertext linked name of the attack object in the summary tab to display reference information for the attack, as shown in [Figure 10 on page 51](#).

Figure 56: Using NSM Log Viewer Attack Reference Information

The screenshot displays the NSM Log Viewer interface. The main window shows a table of log entries with columns for Log ID, Time Received, Alert, User Flag, Comment, Src Addr, Dst Addr, Action, Protocol, Dst, Net Sr, Net Ds, Details, Category, and Subcategory. A specific log entry is highlighted, and a detailed pane on the right provides reference information for the attack object 'HTTP: Windows Media Services NSISlog.DLL Buffer Overflow'.

**Log Viewer [3-IDP@ID]**

Log ID	Time Received	Alert	User Flag	Comment	Src Addr	Dst Addr	Action	Protocol	Dst	Net Sr	Net Ds	Details	Category	Subcategory	
20090806/416941	8/5/09 11:13:33 PM	A			1.1.0.115	1.2.0.58	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface=eth2	Predefined	HTTP: IS cmd.exe Command Exec...
20090806/416943	8/5/09 11:13:33 PM	A			1.1.0.192	1.2.0.102	Conn Dropped	TCP	80	3	0.0.0.0	0.0.0.0	Interface=eth2	Predefined	HTTP: ISS 0 WebDAV SEARCH Co...
20090806/416945	8/5/09 11:13:33 PM	A			1.1.0.212	1.2.0.241	Conn Dropped	TCP							
20090806/416949	8/5/09 11:13:33 PM	A			1.1.0.240	1.2.0.132	Conn Dropped	TCP							
20090806/416949	8/5/09 11:13:36 PM	A			1.1.0.63	1.2.0.159	Conn Dropped	TCP							
20090806/416950	8/5/09 11:13:36 PM	A			1.1.0.88	1.2.0.56	Conn Dropped	TCP							
20090806/416951	8/5/09 11:13:36 PM	A			1.1.0.161	1.2.0.81	Conn Dropped	TCP							
20090806/416956	8/5/09 11:13:36 PM	A			1.1.0.231	1.2.0.243	Conn Dropped	TCP							
20090806/416957	8/5/09 11:13:36 PM	A			1.1.0.231	1.2.0.243	Conn Dropped	TCP							
20090806/416961	8/5/09 11:13:36 PM	A			1.1.0.241	1.2.0.121	Conn Dropped	TCP							
20090806/416966	8/5/09 11:13:42 PM	A			1.1.0.170	1.2.0.91	Conn Dropped	TCP							
20090806/416967	8/5/09 11:13:42 PM	A		Windows 2000 SP4 only?	1.1.0.88	1.2.0.56	Conn Dropped	TCP							
20090806/416971	8/5/09 11:13:45 PM	A			1.1.0.241	1.2.0.121	Conn Dropped	TCP							
20090806/416972	8/5/09 11:13:45 PM	A			1.1.0.23	1.2.0.139	Conn Dropped	TCP							
20090806/417097	8/5/09 11:14:59 PM	A			1.1.0.188	1.2.0.231	Conn Dropped	TCP							
20090806/417098	8/5/09 11:17:37 PM	A			1.1.0.20	1.2.0.143	Conn Dropped	TCP							
20090806/417099	8/5/09 11:17:40 PM	A			1.1.0.199	1.2.0.227	Conn Dropped	TCP							
20090806/417100	8/5/09 11:17:40 PM	A			1.1.0.114	1.2.0.190	Conn Dropped	TCP							
20090806/417101	8/5/09 11:17:40 PM	A			1.1.0.234	1.2.0.123	Conn Dropped	TCP							
20090806/417102	8/5/09 11:17:40 PM	A			1.1.0.189	1.2.0.95	Conn Dropped	TCP							
20090806/417103	8/5/09 11:17:40 PM	A			1.1.0.48	1.2.0.155	Conn Dropped	TCP							
20090806/417104	8/5/09 11:17:40 PM	A			1.1.0.48	1.2.0.155	Conn Dropped	TCP							
20090806/417105	8/5/09 11:17:40 PM	A			1.1.0.48	1.2.0.155	Conn Dropped	TCP							
20090806/417106	8/5/09 11:17:40 PM	A			1.1.0.48	1.2.0.155	Conn Dropped	TCP							
20090806/417107	8/5/09 11:17:43 PM	A			1.1.0.205	1.2.0.103	Conn Dropped	TCP							
20090806/417108	8/5/09 11:17:43 PM	A			1.1.0.109	1.2.0.55	Conn Dropped	TCP							

**HTTP: Windows Media Services NSISlog.DLL Buffer Overflow**

**References**

- <http://online.securityfocus.com/bid/3035/discussion/>
- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0349>
- <http://www.kb.cert.org/vuls/id/113716>
- <http://www.microsoft.com/technet/security/bulletin/MS03-022.mspx>
- <http://secunia.com/advisories/9115>

**Extended Description**

**Last Modified**  
2009-08-13

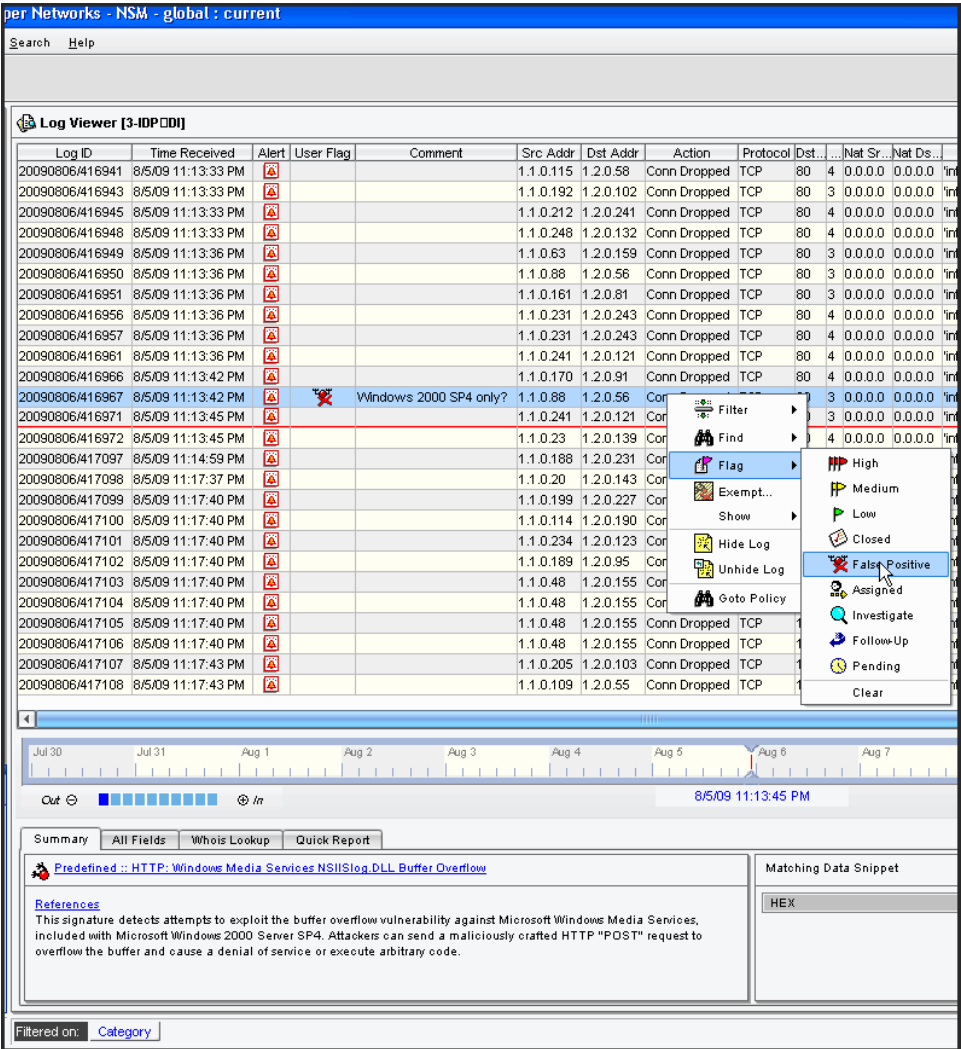
**Impact**  
Windows Media Services may expose IIS to remote arbitrary code execution if media logging is enabled.

**Description**  
Microsoft has reported a buffer overflow vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension handles incoming client requests. This could cause arbitrary code execution in IIS, which is exploitable through Media Services.

**Technical Information**  
Microsoft Media Services provides functionality for providing streaming media content to clients from IIS. It ships with a number of Microsoft Windows 2000 server releases and is also available for download for Windows NT. Microsoft has reported a buffer overflow vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension (nsislog.dll) handles incoming client requests. The logging facility may attempt to write excessive data to an undersized buffer when handling a malformed HTTP client request. This could trigger a denial of service or remote arbitrary code execution in IIS, which is exploitable through Media Services. The issue would occur in servers that are configured to provide logging of media requests. It is possible to exploit this issue by sending an overly long HTTP POST request to the

In this example, we learn that the threat detected applies only to Microsoft Windows 2000 Server SP4. It is a false positive because all of the Windows servers in our network are Windows Server 2008. You can use the NSM Log Viewer flag and comment features to mark logs as false positives. In [Figure 11 on page 52](#), we have marked the log ID 20090806/416967 as a false positive because the attack targets server versions not present in our network.

Figure 57: Using NSM Log Viewer Flag and Comment Features



There are a number of ways you can tune your security policy to reduce false positives. Table 19 on page 52 summarizes some basic tunings.

Table 46: Actions to Take To Reduce False Positives

Type of False Positive	Tuning Required
You trust the source.	Add an Exempt rulebase rule to “whitelist” the trusted source.

Table 46: Actions to Take To Reduce False Positives (*continued*)

Type of False Positive	Tuning Required
The attack applies to a hardware or software version that does not match your destination server.	<p>You have many options:</p> <ul style="list-style-type: none"> <li>• Delete the attack from the rule.</li> <li>• Modify an attack group to exclude the object.</li> <li>• Add an Exempt rulebase rule to whitelist the non-offending attack object.</li> <li>• Modify rule action so traffic is stopped or permitted differently from before.</li> <li>• Modify the rule severity so that you can filter these events differently from before.</li> </ul>
Your team has already patched the vulnerability detected by the attack.	Same as previous.
Upon examination, benign traffic crosses thresholds that trigger protocol anomaly events.	Use the NSM Device Manager to modify protocol anomaly thresholds.

## Adding Rulebases

The IDP rulebase is the primary rulebase in an IDP security policy. When you have sufficiently tuned your IDP rulebase rules so that the security policy generates the level of logs you want, you can add additional rulebases to enable additional detection methods.

Take the same approach to tuning these additional rulebases. Instead of refining the group of attack objects that are relevant, you tune the IDP runtime parameters that set thresholds for detection mechanisms.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)
- [Understanding the Rule-Matching Algorithm on page 45](#)
- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Understanding the Exempt Rulebase on page 67](#)
- [Using Attack Objects on page 60](#)
- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying the IDP Series Device Configuration on page 351](#)



## APE Rulebase Examples

- [APE Rulebase Example: Using Extended Application Objects on page 165](#)
- [APE Rulebase Example: Aggregate and Per-Subscriber Rate Limits on page 170](#)
- [APE Rulebase Example: Matching Custom Application Objects on page 171](#)

### APE Rulebase Example: Using Extended Application Objects

This example shows the usefulness of using extended application objects in APE rules when you want to treat different Web 2.0 applications differently. In IDP OS Release 5.1, the application identification feature is capable of identifying many of the most widely used Web 2.0 applications that run over HTTP. The Juniper Networks Security Center (J-Security Center) provides predefined application signatures for many Web 2.0 applications.

Let's assume you are a network administrator in the IT department of a large enterprise company. Your company executives have decided that it is in the company interest to allow employees basic access to Facebook because it will enable employees to maintain relationships and contacts that serve them professionally. However, the executives have deemed that employees should not use the corporate network to access Facebook games or multimedia content.

To create a policy that enforces these business objectives:

1. Go to the [J-Security Center website](#) to review the list of predefined application signatures, including a number of Facebook application signatures. Make note of the signatures you want to allow and the ones you want to block.
2. Use the NSM Object Manager Application Object viewer to browse the list of predefined extended application objects. [Figure 58 on page 166](#) shows the NSM Object Manager Application Object viewer.

Figure 58: NSM Object Manager: Predefined Extended Application Objects

**Application Objects**

Predefined Application Objects   Custom Application Objects   **Predefined Extended Application Objects**   Application Group Objects

Name	Application Category	Ext ID	Application Type	L7 Protocol	Chain Ord
MYSPACE	SOCIAL-NETWORKING	316	MYSPACE	HTTP	No
TWITTER	SOCIAL-NETWORKING	317	TWITTER	HTTP	No
BEBO	SOCIAL-NETWORKING	321	BEBO	HTTP	No
CLASSMATES	SOCIAL-NETWORKING	322	CLASSMATES	HTTP	No
Hi5	SOCIAL-NETWORKING	329	Hi5	HTTP	No
DOOF	SOCIAL-NETWORKING	290	DOOF	HTTP	No
BLOGGER-POST	SOCIAL-NETWORKING	343	BLOGGER-POST	HTTP	No
MYSPACE-MAIL	SOCIAL-NETWORKING	351	MYSPACE-MAIL	HTTP	No
MYSPACE-CHAT	SOCIAL-NETWORKING	352	MYSPACE-CHAT	HTTP	No
MYSPACE-VIDEO	SOCIAL-NETWORKING	360	MYSPACE-VIDEO	HTTP	No
VKONTAKTE	SOCIAL-NETWORKING	501	VKONTAKTE	HTTP	No
MIXI	SOCIAL-NETWORKING	444	MIXI	HTTP	No
TIANYA	SOCIAL-NETWORKING	445	TIANYA	HTTP	No
KAXIN001	SOCIAL-NETWORKING	447	KAXIN001	HTTP	No
ODNOKLASSNIKI	SOCIAL-NETWORKING	448	ODNOKLASSNIKI	HTTP	No
RENREN	SOCIAL-NETWORKING	449	RENREN	HTTP	No
ADULTFRIENDFINDER	SOCIAL-NETWORKING	480	ADULTFRIENDFINDER	HTTP	No
TARINGA	SOCIAL-NETWORKING	481	TARINGA	HTTP	No
BADOO	SOCIAL-NETWORKING	483	BADOO	HTTP	No
NING	SOCIAL-NETWORKING	484	NING	HTTP	No
NETLOG	SOCIAL-NETWORKING	492	NETLOG	HTTP	No
HYVESDOTNL	SOCIAL-NETWORKING	506	HYVESDOTNL	HTTP	No
PLENTYOFFISH	SOCIAL-NETWORKING	508	PLENTYOFFISH	HTTP	No
NATEON	SOCIAL-NETWORKING	403	NATEON	HTTP	No
BLOGSPOT-POST	SOCIAL-NETWORKING	413	BLOGSPOT-POST	HTTP	No
PING-FM	SOCIAL-NETWORKING	509	PING-FM	HTTP	No

- Double-click a table row to display the details of the application object. You want to learn about the HTTP contexts and patterns that define the application objects so you understand what traffic would be dropped or permitted if you create rules to drop or permit them. [Figure 59 on page 167](#) shows the properties of the HTTP:Facebook-Access application object. Note its signature is found in the HTTP header host details of an HTTP client request.

Figure 59: NSM Object Manager: Extended Application Details

FACEBOOK-ACCESS - Predefined

General

Name: FACEBOOK-ACCESS

L7 Protocol: HTTP

Chain Order: No

Application type: FACEBOOK-ACCESS

Maximum Transactions: none

Signature Match Order: 33323

Members

Member /	Context	pattern	direction
m01	http-header-host	(.*)?(facebook\.com fbcdn\.net)	CTS

Close

Figure 60 on page 168 shows the properties of the HTTP:Facebook-App application object. Note its signature is found in the parsed parameters of an HTTP client request.

Figure 60: NSM Object Manager: Extended Application Details

The screenshot shows the 'FACEBOOK-APP - Predefined' window in the NSM Object Manager. The 'General' tab is active, displaying the following configuration:

- Name: FACEBOOK-APP
- L7 Protocol: HTTP
- Chain Order: No
- Application type: FACEBOOK-APP
- Maximum Transactions: none
- Signature Match Order: 32976

Below the configuration fields is a 'Members' section with a table listing members:

Member	Context	pattern	direction
m02	http-url-parsed-param-parsed	(/ap\.php\?i=.*\.\?v=app_\d+)	CTS

At the bottom right of the window is a 'Close' button.

After you have familiarized yourself with the Facebook application signatures and understand how they will be identified by the application identification feature, you are ready to create APE rules that use them.

- Configure APE rules that use the extended application object in a way that meets your application usage policy objectives. [Figure 61 on page 168](#) shows a set of rules that distinguish between “acceptable” Facebook usage and “unacceptable” Facebook usage. Rule 1 allows access to the Facebook website and Facebook mail in order to allow employees to use the popular site to keep in touch with colleagues and professional acquaintances. Rule 2 drops all other Facebook traffic.

Figure 61: APE Rulebase: Using Extended Applications

The screenshot shows the 'APE' tab in the configuration interface. It displays a table of rules with the following columns: No., Source, User Role, Destination, Service, Application, Extended Application, Action, Notification, VLAN Tag, Severity, and Comments.

No.	Source	User Role	Destination	Service	Application	Extended Application	Action	Notification	VLAN Tag	Severity	Comments
1	any	any	any	Default	any	FACEBOOK:ACCESS FACEBOOK:MAIL	None	Logging	any	Default	Acceptable Facebook.
2	any	any	any	Default	any	FACEBOOK:CHAT FACEBOOK:APP FARMVILLE	Drop Connection	Logging	any	Default	Unacceptable Facebook.

- (Optional) [Figure 61 on page 168](#) depicts APE rules where applications were added one-at-a-time. Alternatively, you can create and maintain application groups based on classifications that you choose. [Figure 62 on page 169](#) and [Figure 63 on page 169](#)

show groups created to support the business classifications discussed in this example—acceptable and unacceptable Facebook.

Figure 62: NSM Object Manager: Creating Application Groups

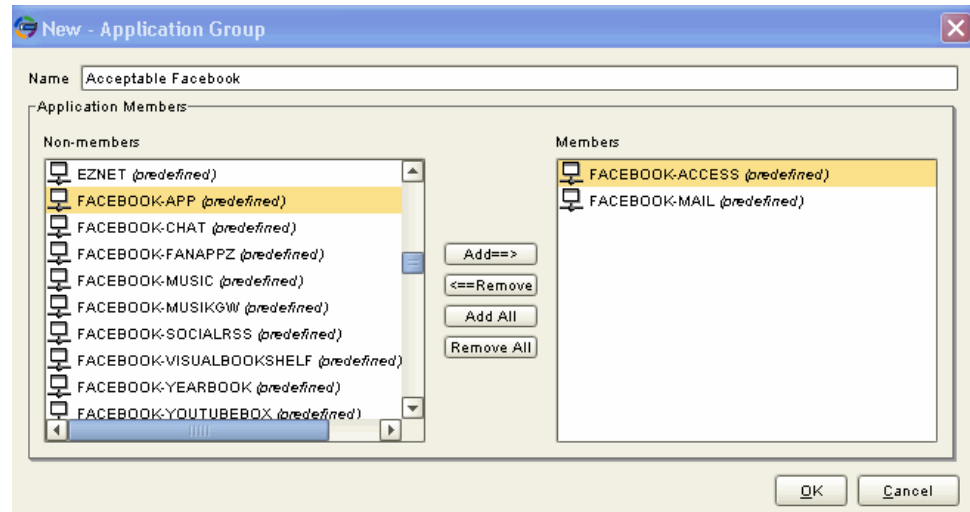


Figure 63: NSM Object Manager: Creating Application Groups

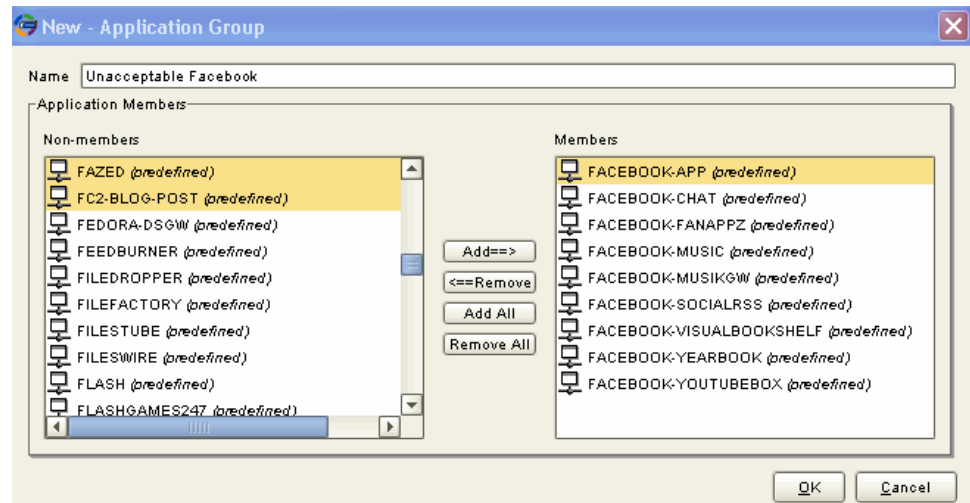


Figure 64 on page 169 shows an APE policy that uses application groups. Note that you use the Application column to add application groups to an APE rule.

Figure 64: APE Rulebase: Using Application Groups

Zone based Firewall IDP APE											
No.	Source	User Role	Destination	Service	Application	Extended Application	Action	Notification	VLAN Tag	Severity	Comments
1	any	any	any	Default	Acceptable Facebook	any	None	Logging	ANY Any	Default	Acceptable Facebook.
2	any	any	any	Default	Unacceptable Facebook	any	Drop Connection	Logging	ANY Any	Default	Unacceptable Facebook.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)

- [Understanding the APE Rulebase on page 69](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)
- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)

## APE Rulebase Example: Aggregate and Per-Subscriber Rate Limits

This example uses an Internet café business to show the usefulness of user-role-based rules. Let's assume that you own an Internet café, and you have deployed a Juniper Networks IC Series UAC solution with a firewall to create a captive portal. A captive portal redirects users to a Web page where they must enter credentials to access the WWW. As your business becomes more popular, you start hearing complaints about network performance during periods of peak use. You take great pride in your service and are particularly distressed when you hear your business customers tell you that the poor network performance is interfering with important work. You think it would be a good idea to have a separate “traffic lane” for users who need Internet access for important work.

You can deploy the IDP Series device in your network and use the application policy enforcement (APE) rulebase to create premium and economy “traffic lanes”. Premium customers pay a higher rate and receive unlimited bandwidth. Economy customers pay a lower rate and are subject to a bandwidth rate limit.

To deploy this solution:

1. Deploy a Juniper Networks IC Series UAC and firewall to create a captive portal and manage user access to the Internet.
2. Use the IC Series administration console to map users to roles, including:
  - Premium—Customers who pay extra for unlimited access.
  - Economy—Customers who pay for basic service.
3. Configure communication between the IC Series device and the IDP Series device so you can use the IDP Series user-role-based policy feature.
4. Use NSM to configure APE rules.

[Figure 65 on page 170](#) shows a set of rules that create traffic lanes for tiered access. Note that all matching parameters are set to Any except user role. This policy does not guarantee premium users a specific rate, but it conserves bandwidth for use by premium users by capping bandwidth for non-premium users.

**Figure 65: APE Rulebase: User-Role-Based Rules to Support Tiered Access**

No.	Match						Action	Notification	VLAN	Severity	Comments
	Source	User Role	Destin.	Service	App.	Extended App.					
1	any	Premium	any	Default	any	any	None	Log...	RVV...	Default	...
2	any	Economy	any	Default	any	any	Rate Limit (262144 kbps, 262144 kbps)	Log...	RVV...	Default	Aggregate rate limit for Economy users.

In the example above, users who belong to the Economy role have unlimited access until the aggregate of the bandwidth used by all Economy role users exceeds the rate limit. At that point, the IDP Series device begins dropping packets until the aggregate for the role falls below the rate limit. If you prefer, you can enable a system-wide option that enforces the rate limit on each user who belongs to the role. You might prefer to enforce rate limiting this way if you want to disclose to users who purchase the Economy package that their bandwidth is capped at a specific rate. You use the CLI to enable per-subscriber rate limiting, and you create an APE rule that sets the agreed upon rate. [Figure 66 on page 171](#) shows the APE rules with a rate limit for per-subscriber enforcement. If any Economy user exceeds 256 kbps client-to-server utilization, the IDP Series device drops packets from the user's session until the rate falls below the threshold.

**Figure 66: APE Rulebase: User-Role-Based Rules to Support Tiered Access**

No.	Match						Action	Notification	VLAN Tag	Severity	Comments
	Source	User Role	Destination	Service	Applica.	Extended App.					
1	any	Premium	any	Default	ANY	any	None	Log...	ANY	Default	...
2	any	Economy	any	Default	ANY	any	Rate Limit (256 kbps, 512 kbps)	Log...	ANY	Default	Per subscriber rate limit.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the APE Rulebase on page 69](#)
- [User-Role-Based Policy Feature Overview on page 58](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Enabling Per-User Rate Limiting for User-Role-Based Rules on page 321](#)

## APE Rulebase Example: Matching Custom Application Objects

This example shows the usefulness of support for custom application objects.

You can create rules for most business cases with the predefined application objects provided by J-Security Center. J-Security Center makes predefined application objects and predefined extended application objects available for download to NSM during signature database updates. For a list of predefined application objects, see the J-Security Center [website](#). In some cases, you might need to manage traffic for applications not yet supported by J-Security Center. First, check with your Juniper Networks representative to see if a predefined application object is forthcoming. If support for your application object is not forthcoming, you can use the NSM Object Manager to define a custom application object. You can then specify that object as a match for APE rules.

For example, suppose you use a final “catch-all” APE rule that drops any traffic not permitted by previous rules. You learn that your organization intends to use Aspera Software’s [FASP protocol](#) instead of FTP to transport large files, so you need to explicitly permit FASP. After checking with J-Security Center to see if a predefined application object is under development, you decide to create your own. Creating a custom application is not a trivial task. To do so, you need to discover the following information:

- Protocol and port—Usually well understood through vendor documentation or network security community websites.
- Signature pattern—Sometimes you can find packet capture files (pcaps) available through network security community websites, such as [Wireshark](#) or [pcap](#). If none can be found, you might have to use a packet capture utility to create your own pcaps and discover the signature pattern.
- Relationship to other application objects—When you create an application object, you specify a signature match order. If traffic matches multiple objects, an application object with the lower signature match-order number is considered the match. You should become aware of traffic that uses the same ports or possibly the same matching pattern.

Use the predefined application naming conventions as a guide to completing the name and category properties. [Figure 67 on page 172](#) shows basic information for FASP.

**Figure 67: NSM Object Manager: Custom Application Object**

The screenshot shows a window titled "Aspera FASP - Custom" with a close button in the top right corner. Inside the window, there are two tabs: "General" and "Detector". The "General" tab is selected and displays the following fields:

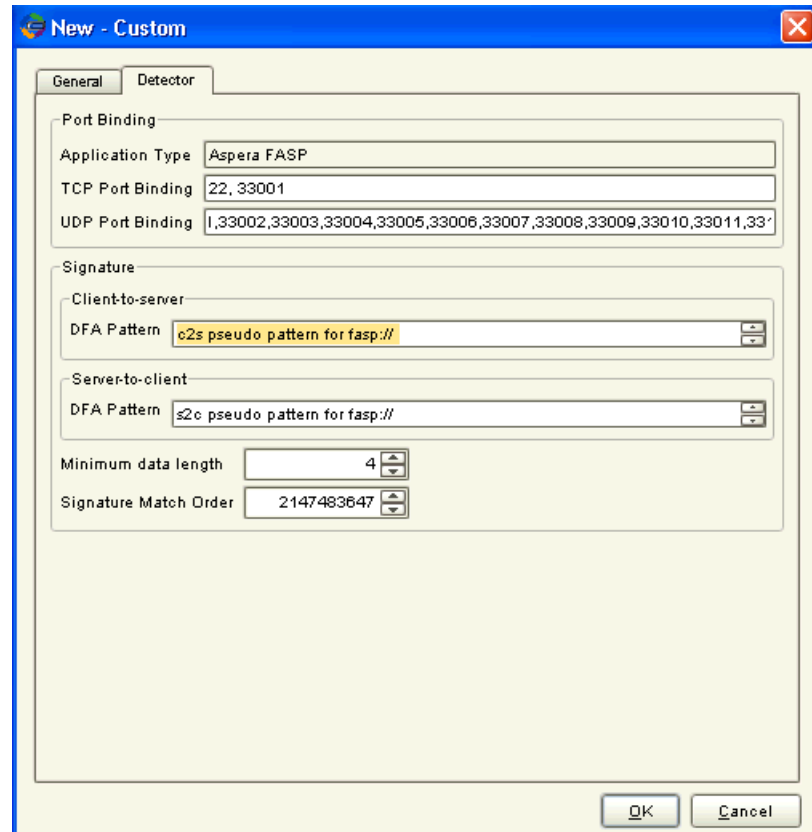
- Name:** A text box containing "Aspera FASP".
- Application Category:** A dropdown menu showing "File-Server".
- Supported Platforms:** A text box containing "idp5.1.0".
- Port Ranges:** A section containing two text boxes:
  - TCP Port Range(s):** A text box containing "22,33001".
  - UDP Port Range(s):** A text box containing "0-65535".

At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Apply".



Figure 68 on page 173 shows a pseudo DFA pattern signature for FASP.

Figure 68: NSM Object Manager: Custom Application Object



After you create the custom application object, it is available to be added to APE rules.

Figure 69 on page 173 shows the custom application in the APE rulebase application list.

Figure 69: APE Rulebase: Adding a Custom Application Object

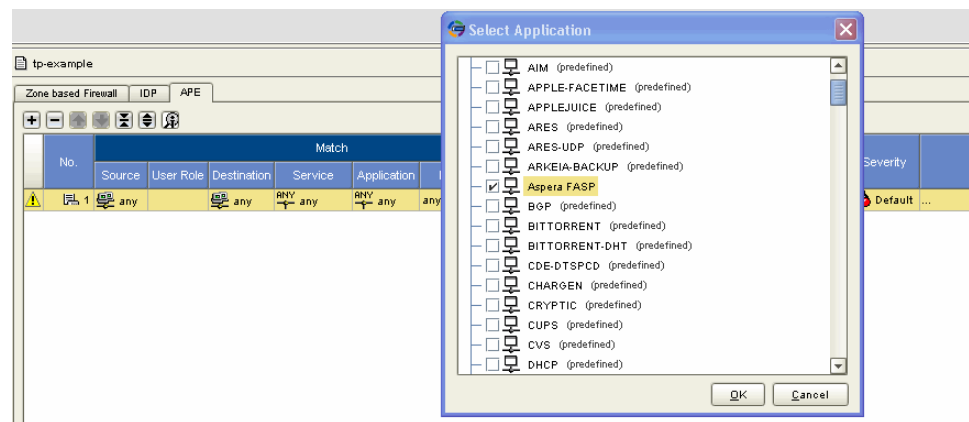


Figure 70 on page 174 shows a set of APE rules. Here, we have added the rule that permits FASP before the catch-all rule that drops all traffic not permitted by previous rules.

Figure 70: APE Rulebase: Rule Order

Zone based Firewall IDP APE										
No.	Match						Action	Notification	VLAN Tag	Severity
	Source	User Role	Destination	Service	Application	Extended Application				
1	any	any	any	Default	Aspera FASP	any	None	Logging	Any	Default
2	any	any	any	any	any	any	Drop Connection	Logging	Any	Default

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

# Exempt and Backdoor Rulebase Examples

- [Exempt Rulebase Example: Exempting a Source Destination Pair on page 175](#)
- [Exempt Rulebase Example: Exempting an Attack Object on page 176](#)
- [Backdoor Rulebase Example: netcat on page 176](#)

## Exempt Rulebase Example: Exempting a Source Destination Pair

Suppose in your security policy implementation there are schedule phases where your security team probes your internal network for vulnerabilities and you want the IDP Series device to generate logs, and phases where you have put your security policy in place and now want to exclude security team traffic from the generated logs. To support these alternative phases, you can create an Exempt rulebase rule and toggle it off and on.

To create an Exempt rulebase rule:

1. Create address objects that contain the security team IP addresses and the protected servers.
2. Add the Exempt rulebase to your security policy.
3. Add a rule that specifies the source/destination match condition to exempt.
4. Add the All group of attack objects.

[Figure 71 on page 175](#) shows an Exempt rulebase rule.

**Figure 71: Exempt Rulebase Rule**

No.	ID	Match		Attacks	Comments
		Source	Destination		
1	1	OurSecurity	Protected Servers	All	Security research might produce false positives from engineering servers.

To toggle the rule off, right-click it and select **Disable**.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*.

- [Understanding the Exempt Rulebase on page 67](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following related topic is included in the *IDP Series Administration Guide*.

- [Configuring Exempt Rulebase Rules \(NSM Procedure\) on page 227](#)

---

## Exempt Rulebase Example: Exempting an Attack Object

Suppose your security policy detects HTTP Buffer Overflow: Header attacks on your internal network, but you know this can safely be ignored. You can exempt this traffic from inspection to optimize IDP Series performance and eliminate unnecessary logs.

To exempt an attack object:

1. If you have not done so already, create an address object for your internal network.
2. Add the Exempt rulebase to your security policy.
3. Add a rule that specifies a source that is the internal network and destination that is anywhere.
4. Add the relevant attack object. In this example, add HTTP Buffer Overflow: Header.

### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*.

- [Understanding the Exempt Rulebase on page 67](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following related topic is included in the *IDP Series Administration Guide*.

- [Configuring Exempt Rulebase Rules \(NSM Procedure\) on page 227](#)

---

## Backdoor Rulebase Example: netcat

The **netcat** utility can open connections to any port and can offer services on any port. We know that attackers use **netcat** to create and exploit backdoors.

Suppose an attacker gains access to a host in your network and installs a **netcat** utility. The following example shows a **netcat** command an attacker can run:

```
nc 10.1.1.100 4444
```

This command opens a connection to the computer at IP address 10.1.1.100 over port 4444.

[Figure 72 on page 177](#) shows a recommended rule that would detect the interactive traffic generated by **netcat** in this case.

Figure 72: Backdoor Rulebase

No.	Match					Operation	Action
	From Zone	Source	To Zone	Destination	Service		
1	any	Web Server Group	any	any	ECHO FTP rtalk	Ignore	Accept
2	any	Web Server Group	any	any	any	Detect	Accept

Rule 1 ignores the interactive traffic you expect for interactive services in your network (Telnet, SSH, RSH, NetMeeting, and VNC). Rule 2 detects all other interactive traffic. Rule 2 detects interactive traffic that occurs over a port where there typically is not interactive traffic.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Backdoor Rulebase on page 85](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)



# Inspection of HTTPS Traffic

- [Example: Implementing Inspection of Outbound SSL Traffic on page 179](#)
- [Example: Exempting Outbound SSL Traffic from Inspection on page 181](#)

## Example: Implementing Inspection of Outbound SSL Traffic

---

When users in your protected network connect to HTTPS servers on the WWW, the application activity within the encrypted sessions cannot be inspected by most intrusion prevention systems. Once an encrypted session is established with the server, the user might download a seemingly harmless file or executable that contains a trojan. If this happens, an attacker could launch an attack from within the protected network.

To protect your network against this risk, you could create a firewall policy that blocks HTTPS connections from the protected zone to the unprotected zone. To protect your network *and* support legitimate access to the WWW, you want a solution that can inspect the HTTPS traffic.

The IDP solution supports SSL inspection in two ways:

- Using server private keys. Use this method when inspecting traffic to internal servers where you have access to the server private key.
- Using the SSL forward proxy feature. Use this method when the server private key method is not practical (for example, for traffic to servers on the WWW).



**NOTE:** If you enable both methods, the IDP Series device performs SSL inspection using the SSL forward proxy method and does not use the server private keys.

The following procedure provides the basic steps you take to implement the SSL forward proxy feature.

To implement the SSL forward proxy feature:

1. From the IDP Series device command-line interface:
  - a. Generate the root certificate authority (CA) that the IDP Series device uses to create and sign new certificates used in SSL proxy operations. The following example creates a root CA:

```
[root@default host admin]# scio ssl ca create US CA Sunnyvale 'Juniper Networks Inc.'
'SSL Inspection policy' 'Juniper IT Services' 'admin@juniper.net' 1024
```



**NOTE:** The system prompts the end user to install the CA you create in this step. Take care to configure an organization name that an end user is likely to accept, such as your company name.

- b. Verify the CA was added:

```
[root@defaulthost admin]# scio ssl ca show
serial=8E0012848A2D7CCD
subject= /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks Inc./OU=SSL Inspection
policy/CN=Juniper IT Services/emailAddress=admin@juniper.net
issuer= /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks Inc./OU=SSL Inspection
policy/CN=Juniper IT Services/emailAddress=admin@juniper.net
notBefore=Jun 25 22:13:23 2009 GMT
notAfter=Jun 23 22:13:23 2019 GMT
```

- c. (Optional) Distribute the CA to your users and have them install the CA in their Web browser.

The following example prints to the screen the CA in PEM format. You can copy this text to a file that your users can import into their browsers.

```
[root@default host admin]# scio ssl ca export
-----BEGIN CERTIFICATE-----
MIIC1TCCAj4CCQCQABKEi18zTANBgkqhkiG9w0BAQUFADCBrijELMAKGA1UEBhMC
VVMwXzAjbG9NVBAGTAkNBMRIEAYDVQQHEw1TdW5ueXZhbGUxHjAcBgNVBAoTFUp1
bm1wZXIwTmV0D29ya3MgSW5jLjJlEeMBG1uECMVU1NMIU1uc3B1Y3Rpb24gcG9s
aW5NSW9mG9kYVdVQQDEXNkdW5pcGvYyE1U1FN1c2pZVzV2MSAeHGYKJoZShvcnAQBg
FhFhZG1pbG8BqdW5pcGvYyLm51dDAeFw0OTA2MjUyMjEzEzNmNaFw0xOTA2MjMjMjEz
MjNmNmIuG9w0CQYDVQGEwJVUzELMAKGA1UECBMCQEQEExEjAQBGNVBACTCVN1bm55
dmFsZTEeMBwGA1UEChMVSnVuXzB1ciB0ZXR3b3JrcyBjbmMuMR4wHAYDVQLExVT
U0wgSW52czGvjdG1vb1Bwb2xpy3kxHDAaBgNVBAMTEOp1bm1wZXIwSVQgU2Vydmlj
ZXMXIDAeIDAEBgkqhkiG9w0BCQEWEFkbw1uQGp1bm1wZXIubmV0MIGfMA0GCSqGSIb3
DQEBQAAQANADCBiQKBgQDAsn2NFAxTRcPShf9sg+Ccn1rUYzPuVHTw1GUtnHHB
o/ofXENGETgGLZ/jck-L27103zPgD67yyHs08SXWvgC3MJukb14kqyTgyu3/E9
wkiIey8W4XzyBxRcfW2YEGmC0cFExdm+C6DrAailddTQdgelxZ7nfIj24iiBhYYM
GQIDAQABMA0GCSqGSIb3DQEBBQAA4GBAFTeRz9DHcbohdJfGqWpJ5+MDgsX9041
f/WzHXftak4ZHj0ryYvVaRlUyitehMX1KvMPQjYXf+TE2vF9yYqmoCj6710LiuzZJ
Tw4gwy9E9p58krqvZu4F2/kVM+yEAKsUIjBme1R1L6Az3kLauHvkyAbMcSFNZq2b0
7Z8WbQqn3o6s
-----END CERTIFICATE-----
```

2. In NSM:
  - a. Add IDP rulebase rules that target HTTPS traffic:
    - Match traffic flowing in the client-to-server direction.



- Be sure to account for HTTPS over nonstandard ports. The application identification feature is not applied when the IDP Series device proxies an SSL session.
- Include attack objects to inspect *both* the SSL session and the HTTP payload.



**TIP:** You must include at least one SSL attack object. We recommend you include the SSL: SERVER-CERT-FAILS-VALIDATION anomaly to detect invalid certificates (that is, certificates signed by an unknown CA or that cannot be validated against the issuer CA). An invalid certificate might indicate a phishing attack. You can drop or log matching sessions.

- Specify action and notification options.
  - a. Push the updated security policy to the IDP Series device.
  - b. Push the updated security policy to the IDP Series device.
  - c. Review logs to verify the feature operates as you expect.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of SSL Traffic Overview on page 113](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Using the SSL Private Server Key to Enable Inspection of SSL Traffic on page 308](#)
- [scio ssl on page 531](#)
- [scio const on page 505](#)

## Example: Exempting Outbound SSL Traffic from Inspection

The privacy policy for your business might include cases where sessions should remain encrypted throughout. For example, suppose you have an agreement with your users that your network security infrastructure will not interfere with SSL encrypted connections to banking sites. In these cases, you can create a whitelist of destination domain names, IP addresses, and subnets you want exempted from IDP policy inspection. If a server is included in the whitelist, the IDP system does not decrypt the traffic or inspect it. Instead, this traffic is passed through the IDP Series device uninspected.



**NOTE:** The whitelist applies only to traffic processing based on the SSL forward proxy feature. You would not use a whitelist to exclude inspection of traffic to internal destination servers. If desired, you can use a security policy rule to exempt such traffic from inspection.

The following example shows the format of a whitelist file:

```
10.0.0.1
1.0.0.0/8
70.34.21.82
trustedsite.com
landing.trustedsearch.com
```

Each line in the whitelist file specifies the IP address or domain name for a destination server. To whitelist multiple sites with one entry, you can use an IP prefix to match address blocks and a domain suffix to include all subdomains.

The domain name in your whitelist should match the common name (CN) entry in the certificate presented by the destination server. For example, suppose the certificate for the E-Trade HTTPS server contains the following subject:

```
C=US, ST=Georgia, L=Alpharetta, O=ETRADE FINANCIAL CORPORATION, OU=Global
Information Security, CN=us.etrade.com
```

You can whitelist this site by adding the string **us.etrade.com** or the string **etrade.com** to your whitelist file.

In most cases, the CN entry in the server certificate for a website matches the server name that appears in the browser address bar. In some cases, there are differences. You can use the features of your Web browser to find the CN entry in the server certificate for the website.

[Figure 73 on page 183](#) shows the location of the certificate details in Firefox.

Figure 73: Firefox: Displaying the Server Certificate for a Website

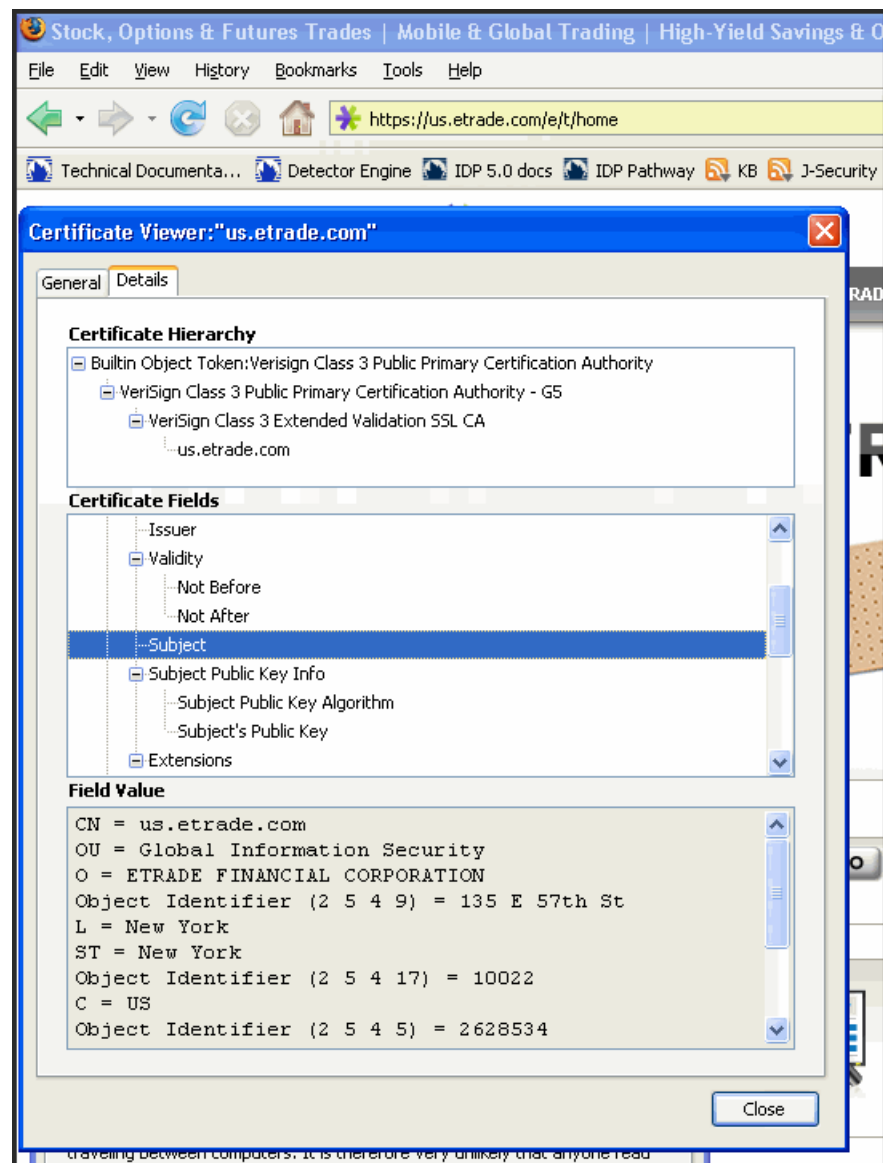
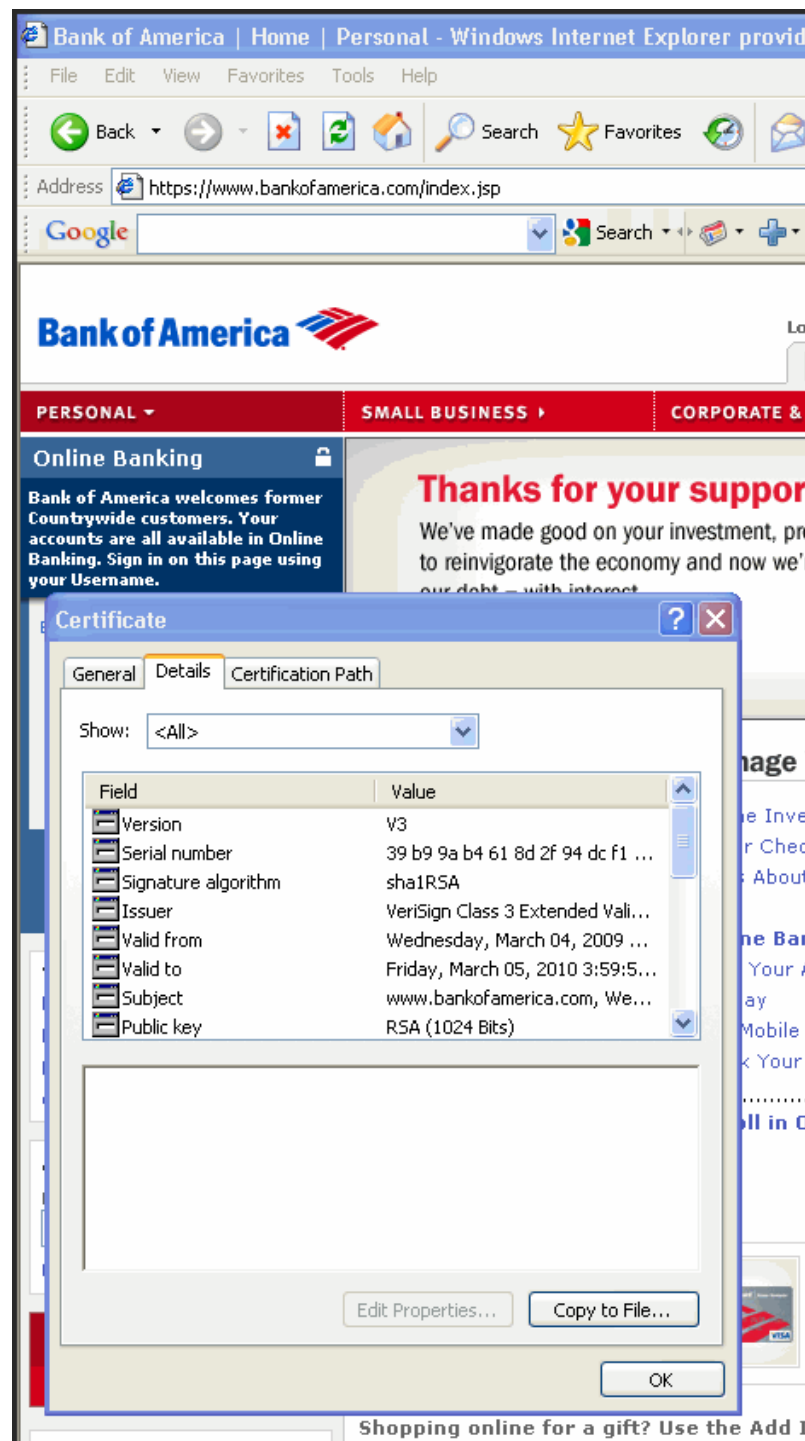


Figure 74 on page 184 shows the location of the certificate details in Internet Explorer.

Figure 74: Internet Explorer: Displaying the Server Certificate for a Website



To implement a whitelist:

1. Log into the CLI as **admin** and enter **su -** to switch to root.

2. Use an editor like vi to create a whitelist file. A whitelist file should contain the IP address prefixes and/or domain name suffixes you want to exempt from inspection. For example:

```
[root@default host admin]# vi /tmp/whitelist.txt
e-trade.com
bankofamerica.com
```

3. Run the following command to import the whitelist entries:

```
[root@default host admin]# scio ssl whitelist import /tmp/whitelist.txt
```



**NOTE:** To update the active whitelist, import an updated whitelist file. To clear the whitelist, import a file that contains only one empty line.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of SSL Traffic Overview on page 113](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [scio ssl on page 531](#)



## PART 3

# Configuration

- [Getting Started on page 189](#)
- [Simulation Mode on page 201](#)
- [Configuring Profiler on page 203](#)
- [Configuring the IDP Rulebase on page 213](#)
- [Configuring Additional Security Policy Rulebases on page 227](#)
- [Working with Attack Objects on page 243](#)
- [Working with Application Objects on page 279](#)
- [Configuring Logging Features on page 295](#)
- [Using the scio Command to Implement Advanced Features on page 307](#)





## CHAPTER 21

# Getting Started

- [Supported Tools for Management Tasks on page 189](#)
- [Connecting to ACM on page 191](#)
- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)
- [Configuring Virtual Routers \(ACM Procedure\) on page 192](#)
- [Getting Started with the Default Configuration on page 194](#)
- [Developing Security Policies Task Summary on page 195](#)
- [Using Predefined Security Policies on page 197](#)
- [Using the New Policy Wizard \(NSM Procedure\) on page 198](#)

## Supported Tools for Management Tasks

[Table 47 on page 189](#) identifies supported tools for IDP Series management tasks.

**Table 47: Management Tools by Task**

Task	ACM	CLI	NSM
Enable and configure application identification.	No	Yes	Recommended
Enable application volume tracking.	No	Yes	Recommended
Enable and run Profiler.	No	No	Yes
Configure and manage security policies and application resource enforcement policies.	No	Advanced users only	Recommended
Configure attack detection thresholds.	No	Yes	Recommended
Configure the antispoof feature.	No	No	Yes
Enable and configure the flow bypass feature.	No	Yes	No
Enable GRE decapsulation.	No	Yes	Recommended
Configure levels of GRE decapsulation.	No	Yes	No

Table 47: Management Tools by Task (*continued*)

Task	ACM	CLI	NSM
Enable GTP decapsulation.	No	Yes	Recommended
Configure levels of GTP decapsulation.	No	Yes	No
Enable and configure IPsec ESP NULL decapsulation.	No	Yes	No
Enable and configure MPLS decapsulation.	No	Yes	No
Enable SSL decryption.	No	Yes	Recommended
Configure SSL inspection.	No	Yes	No
Configure log settings (attack logs, application logs, device logs, syslog, basic SNMP server settings).	No	No	Recommended
Configure extensive SNMP reporting.	No	Yes	No
Configure interface aliasing for logging.	Yes	No	No
Configure virtual circuits (also called traffic interfaces).	Recommended	Advanced users only	No
Configure virtual routers, including deployment mode and bypass settings.	Recommended	Advanced users only	No
Configure the management interface	Recommended	Advanced users only	No
Enable Layer 2 bypass.	Yes	No	No
Enable peer port modulation.	Yes	No	No
Enable interface signaling.	No	Edit user_funcs	No
Configure the NSM agent.	Recommended	Yes	No
Configure credentials for interoperability with Network and Security Manager, IC Series Unified Access Control, and SA Series SSL VPN appliances.	Yes	No	No
Set host and network information, such as fully qualified domain name, default gateway, DNS server, and so forth.	Yes	No	No
Set access, such as passwords for root and admin, SSH requirements, and access to ACM.	Yes	No	No
Upgrade software.	No	Yes	Yes

Table 47: Management Tools by Task (*continued*)

Task	ACM	CLI	NSM
Update detector engine.	No	No	Yes
Update NSM attack database.	No	No	Yes
Update the device configuration.	File upload	<b>scio const</b> commands	Configuration push
Stop and start IDP processes.	No	Yes	Profiler
Reboot.	Yes	Yes	Yes
Shut down.	No	Yes	No

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [scio Configuration Commands Task Summary on page 307](#)
- [NSM Device Configuration Management Task Summary on page 343](#)
- [Updating IDP OS Software \(NSM Procedure\) on page 390](#)
- [Upgrading Software \(CLI Procedure\) on page 389](#)
- [Restarting the IDP Engine on page 383](#)
- [Rebooting and Shutting Down the IDP Series Appliance on page 384](#)

## Connecting to ACM

You use the Appliance Configuration Manager to configure IDP Series device network settings.

To connect to ACM:

1. In the Address or Location box of your Web browser, enter **https://IP**, where *IP* is the IP address you assigned to the management interface. For example, if you configured the IP address 10.100.200.1, enter **https://10.100.200.1**.



**NOTE:** ACM access uses SSL, so you must type **https://** and not **http://**.

2. Log in as **root**. If you do not know the **root** password, contact the administrator who ran the initial setup for the IDP Series device.

For information on using ACM to configure network communication, see the ACM online Help.



**TIP:** ACM logs are located in `/var/log/httpd/error_log`. You can use these logs to troubleshoot errors connecting to ACM.

---

## Connecting to the Command-Line Interface (CLI Procedure)

---

You use the command-line interface (CLI) to use CLI utilities, such as `bypassStatus`, `scio`, `sctop`, `idp.sh`; or Linux diagnostic commands, such as `ethtool`.

To connect to the command-line interface:

1. Use SSH to connect to the IP address or hostname for the management interface.
2. Log in as **admin** and enter `su -` to switch to **root**.
3. At the secure shell, define IDPDIR as follows:

```
IDPDIR=/usr/idp
export IDPDIR
```



**NOTE:** Bash is the default shell and bash commands are shown in the example. If you use a different shell, use the equivalent commands.



**NOTE:** If you automate management or monitoring tasks with cron jobs or daemons that run CLI commands, remember to define the shell environment at the top of your script file. For example, if you create a Bash script to run as a cron job, include the following line at the top of your script file:

```
export IDPDIR=/usr/idp
```

---

## Configuring Virtual Routers (ACM Procedure)

---

A virtual router is a logical pair of traffic interfaces that handles traffic into and out of the IDP Series device. You use the ACM Configure Virtual Routers page to configure the features of the virtual routers, including deployment mode and bypass options. For background information on these features, see the *IDP Series Concepts and Examples Guide*. [Figure 75 on page 193](#) shows the ACM Configure Virtual Routers page.

Figure 75: ACM Configure Virtual Routers Page

### Configure Virtual Routers

In this step, you must configure the interfaces that the IDP Sensor will use to handle traffic.

For each pair of interfaces, select the mode you want each pair to run in.

Active?	Interfaces	Virtual Router	Mode	NIC State (after system unavailability)	NIC State (after graceful shutdown)
<input checked="" type="checkbox"/>	eth2,eth3	vr0	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth4,eth5	vr1	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth6,eth7	vr2	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth8,eth9	vr3	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth10,eth11	vr4	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>

☒ Enable layer2 bypass  
☐ Enable Peer Port Modulator  
 Failover timeout value:

To configure virtual routers:

1. Connect to ACM.
2. From the main menu, click **Reconfigure Virtual Routers**.
3. On the Configure Virtual Routers page:
  - Select the box in the Active? column to enable the virtual routers you plan to connect to your network.
  - Specify a deployment mode: transparent or sniffer.
  - For transparent mode virtual routers, specify how you want to handle failure or shutdown: internal bypass, external bypass, or NICs off.
  - Specify whether you want to enable Layer 2 bypass.
  - Specify whether you want to enable peer port modulation.
  - Specify a failover timeout value.

For details, see the ACM online Help.
4. Click **Next** to advance the wizard until you reach the Brief Configuration Report page.
5. Review, save, and apply your configuration changes.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Connecting to ACM on page 191](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Network Interfaces Overview on page 11](#)

## Getting Started with the Default Configuration

---

The purpose of this topic is to provide you with a workflow to get started using an IDP Series appliance with the default configuration.

Follow these basic steps to get started:

1. Read the [release notes](#) for your release. The release notes contain important release-related information about release-specific features, unsupported features, changed features, fixed issues, and known issues. The information in the release notes is more current than the information in this guide.
2. Install the IDP Series appliance, configure network settings and virtual routers, and connect the device to your network.

For details, see the [installation guide](#) for your IDP Series appliance.

3. Add the device to the NSM Device Manager.
4. Upgrade IDP OS software to the latest version (if applicable).
5. Update the detector engine and NSM attack object database.
6. Become familiar with the default security policy (named Recommended).
7. Use the documentation to become familiar with the product features and user interface:
  - Use the [IDP Series Concepts and Examples Guide](#) to become familiar with the product features.
  - Use this guide, the [IDP Series Administration Guide](#), to learn the steps to implement the product features and monitor security events.
  - Use the Appliance Configuration Manager (ACM) online Help for information about using ACM.
  - Use CLI man pages for syntax and parameter hints for CLI commands.
  - Use the NSM online Help for information about using the NSM user interface.
8. Run Profiler to discover the network hosts you want to protect.
9. Review logs to verify the initial deployment.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)
- [Updating IDP OS Software \(NSM Procedure\) on page 390](#)
- [Adding IDP Series Devices to NSM Device Manager on page 344](#)
- [Developing Security Policies Task Summary on page 195](#)
- [Profiler Task Summary on page 203](#)

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

The following additional related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Example: Fine-Tuning a Security Policy on page 48](#)

## Developing Security Policies Task Summary

An IDP security policy allows you to use various attack detection and prevention techniques on traffic that traverses your network.

To create an effective security policy, follow these basic steps:

1. Run the New Policy wizard to create a security policy object. The new security policy can be based on a predefined template.
2. Use the Security Policy editor to add one or more rulebases. [Table 48 on page 195](#) describes the IDP security policy rulebases. A security policy can contain only one instance of any rulebase type.

A *rulebase* is an ordered set of rules that use a particular detection method to identify and prevent attacks.

3. Within rulebases, configure rules.

*Rules* are instructions that provide context to detection methods. Rules specify:

- A source/destination/service match condition that determines which traffic to inspect.
- Attack objects that determine what to look for (IDP rulebase and Exempt rulebase).
- Actions that determine what to do when an attack is detected or the application rate limit is reached.
- Notification options, including logs, alerts, and packet captures.

Each rulebase can contain up to 40,000 rules.

4. Fine-tune your security policy as you learn more about your network and security requirements and IDP Series capabilities.

**Table 48: IDP Security Policy Rulebases**

Rulebase	Description
IDP rulebase	<p>Protects your network from attacks by using attack objects to detect known and unknown attacks. Juniper Networks provides predefined attack objects that you can use in security policy rules. You can also configure your own custom attack objects.</p> <p>See <a href="#">“Modifying IDP Rulebase Rules (NSM Procedure)” on page 213</a>.</p>

Table 48: IDP Security Policy Rulebases (*continued*)

Rulebase	Description
Exempt rulebase	<p>Enables you to exclude known false positives or to exclude a specific source, destination, or source/destination pair from matching an IDP rule. If traffic matches a rule in the IDP rulebase, the IDP engine attempts to match the traffic against the Exempt rulebase before performing the action specified.</p> <p>See <a href="#">“Configuring Exempt Rulebase Rules (NSM Procedure)” on page 227.</a></p>
APE rulebase	<p>Enables you to set an action for traffic that matches an application signature. Actions include dropping the connection, closing client and/or server, applying a DiffServ marker, applying a rate limit condition, or applying both a DiffServ market and a rate limit condition.</p> <p>See <a href="#">“Configuring the APE Rulebase (NSM Procedure)” on page 228.</a></p>
Backdoor rulebase	<p>Protects your network from mechanisms installed on a host computer that facilitates unauthorized access to the system. Attackers who have already compromised a system typically install backdoors (such as Trojans) to make future attacks easier. When attackers send and retrieve information to and from the backdoor program (as when typing commands), they generate interactive traffic that the IDP engine can detect.</p> <p>See <a href="#">“Configuring Backdoor Rulebase Rules (NSM Procedure)” on page 233.</a></p>
SYN Protector rulebase	<p>Protects your network from SYN-floods by ensuring that the three-way handshake is performed successfully for specified TCP traffic. If you know that your network is vulnerable to a SYN-flood, use the SYN-Protector rulebase to prevent it.</p> <p>See <a href="#">“Configuring SYN Protector Rulebase Rules (NSM Procedure)” on page 235.</a></p>
Traffic Anomalies rulebase	<p>Protects your network from attacks by using traffic flow analysis to identify attacks that occur over multiple connections and sessions (such as scans).</p> <p>See <a href="#">“Configuring Traffic Anomalies Rulebase Rules (NSM Procedure)” on page 237.</a></p>
Network Honeypot rulebase	<p>Protects your network by impersonating open ports on existing servers on your network, alerting you to attackers performing port scans and other information-gathering activities.</p> <p>See <a href="#">“Configuring Network Honeypot Rulebase Rules (NSM Procedure)” on page 240.</a></p>

#### Related Documentation

The following additional related topics are included in the *IDP Series Administration Guide*:

- [Using Predefined Security Policies on page 197](#)
- [Using the New Policy Wizard \(NSM Procedure\) on page 198](#)

The following additional related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)



## Using Predefined Security Policies

The Juniper Networks Security Center team (J-Security Center) provides the default IDP security policy—named Recommended. We advise that you use this policy to protect your network from the likeliest and most dangerous attacks.

Table 18 on page 49 summarizes the settings of the recommended security policy.

**Table 49: Recommended Security Policy Settings**

Property	Value
Rulebase	IDP rulebase.
Rules	Nine rules, distinguished by attack object.
Source	<b>Any</b> , meaning the source setting is not used to match traffic.
Service	<b>Default</b> , meaning the matching property is based on the service bindings of the attack object specified by the rule.
Destination	<b>Any</b> , meaning the destination setting is not used to match traffic.
Attacks	<ul style="list-style-type: none"> <li>• Recommended IP</li> <li>• Recommended TCP</li> <li>• Recommended ICMP</li> <li>• Recommended HTTP</li> <li>• Recommended SMTP</li> <li>• Recommended DNS</li> <li>• Recommended FTP</li> <li>• Recommended POP3</li> <li>• Recommended IMAP</li> <li>• Recommended Trojan</li> <li>• Recommended Virus</li> <li>• Recommended Worm</li> </ul>
Action	<b>Recommended</b> , meaning the action is specified by the attack object
Notification	<b>Logging</b> .

If you prefer, you can copy this security policy and use it as a template for a custom security policy tailored for your network. You use the New Security Policy wizard to create a custom security policy based on a template.

Table 50 on page 198 describes other IDP security policy templates.

Table 50: IDP Security Policy Templates

Template	Description
all_with_logging	Includes all attack objects and enables packet logging for all rules. This policy is provided for lab use and is not recommended in production.
all_without_logging	Includes all attack objects but does not enable packet logging.
dmz_services	Protects a typical DMZ environment.
dns_server	Protects DNS services.
file_server	Protects file sharing services, such as SMB, NFS, FTP, and others.
getting_started	Contains very open rules. Useful in controlled lab environments, but should not be deployed on live networks with heavy traffic.
idp_default	Contains a set of attack groups that balances security and performance.
web_server	Protects HTTP servers from remote attacks.

If you use these templates, we advise you to customize them for your deployment. At a minimum, you should change the destination IP setting from **Any** to the IP addresses for specific servers you want to protect.



**NOTE:** Predefined policies include only client-to-server attack objects. If you are interested in tracking server-to-client attacks, be sure to add rules for them to your policy.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)
- [Using the New Policy Wizard \(NSM Procedure\) on page 198](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

## Using the New Policy Wizard (NSM Procedure)

You use the security policy wizard to create a security policy. The security policies you create with the wizard must have a new name but can be based on existing policies or templates.

To create a security policy:

1. From the NSM main menu, select **File > New Policy** to display the New Policy wizard.
2. On the first page, complete the settings described in [Table 51 on page 199](#) and then click **Next**.

**Table 51: New Policy Wizard: Page One**

Setting	Description
Name	A string to identify the policy.
Comments	Text to further identify the policy. In the security policy list, you can sort on comments.

3. On the second page, complete the settings described in [Table 52 on page 199](#) and then click **Next**.

**Table 52: New Policy Wizard: Page Two**

Setting	Description
Create new Policy for	<p>Select this option to create a security policy.</p> <p>If you select this option, the wizard displays the following set of device types:</p> <ul style="list-style-type: none"> <li>• Firewall/VPN</li> <li>• Firewall/VPN with IDP</li> <li>• Standalone IDP</li> </ul> <p>Select <b>Standalone IDP</b>.</p>
Use Existing Policy	<p>Use this option to assign an existing policy to one or more IDP Series devices.</p> <p>If you select this option, the wizard displays a drop-down list of existing policies.</p> <p>Select a policy from the list.</p> <p><b>NOTE:</b> This procedure involves creating a new policy. For this procedure, do not select Use Existing Policy.</p>

4. On the next pages, complete the preconfiguration options described in [Table 53 on page 199](#). Click **Next** to advance through the pages.

**Table 53: New Policy Wizard: Preconfiguration Options**

Setting	Description
Use Predefined Policy Template	<p>Select this option to create a security policy based on a predefined template.</p> <p>If you select this option, the wizard displays a drop-down list of predefined templates.</p> <p>Select one and click <b>Next</b>.</p>

Table 53: New Policy Wizard: Preconfiguration Options (*continued*)

Setting	Description
Configure IDP Policy	<p>Select this option and complete the rule properties on the next page to generate a policy with the following features:</p> <ul style="list-style-type: none"> <li>• IDP rulebase.</li> <li>• Multiple rules matching any source, any destination, and default services.</li> <li>• Multiple rules are distinguished by the attack object severity group, action, and notification option you configure in the next wizard page.</li> </ul>
Empty Policy	Select this option to create an empty policy that you can later modify.

5. On the next to the last page, select the device targets for the policy and then click **Next**.

6. Click **Finish** to save the policy.

The new policy appears in the security policy list.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)
- [Using Predefined Security Policies on page 197](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Components of an IDP Security Policy on page 41](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

# Simulation Mode

- [Enabling Simulation Mode on page 201](#)

## Enabling Simulation Mode

---

Simulation mode is an operational mode in which the IDP Series device examines traffic but only simulates security policy actions, generating simulation mode logs that indicate the programmed action. Simulation mode is disabled by default.

To enable simulation mode:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Run the following command to display the current setting for simulation mode:

```
[root@defaulthost conf]# scio const -s s0 get sc_simulation_mode
scio: sc_simulation_mode = 0x0
```

The value 0x0 indicates simulation mode is disabled.

3. Run the following command to enable simulation mode:

```
[root@defaulthost conf]# scio const -s s0 set sc_simulation_mode 1
scio: setting sc_simulation_mode to 0x1
```

The value 0x1 indicates simulation mode is enabled.

You do not need to restart the IDP engine (`idp.sh`) or push a policy to initiate changes to your simulation mode setting. Changes you make to kernel constants from the CLI do not persist across restarts.

To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the simulation mode constant within the `user_start_pre_policy()` section:

```
user_start_pre_policy ()
{
    # Disable ARP spoofing detection
    # -----
    # If you are running clusters with virtual MAC addresses, IDP will treat
    # these as spoofed ARP packets since the MAC addresses in the ethernet
```

```
# frame will be different from what is inside the ARP request/response. If
# you have multiple virtual routers, you need to perform this operation on
# all defined virtual routers.
#
# $SCIO const -v vr0 set sc_arp_spoof_detect 0
# $SCIO const -s s0 set sc_mpls_decapsulation 1
$SCIO const -s s0 set sc_simulation_mode 1
return;

}
```

3. Save the file.

**Related  
Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Viewing Simulation Mode Logs on page 461](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Simulation Mode Overview on page 33](#)

## CHAPTER 23

# Configuring Profiler

- [Profiler Task Summary on page 203](#)
- [Configuring Profiler Options \(NSM Procedure\) on page 204](#)
- [Modifying Profiler Settings on page 210](#)

### Profiler Task Summary

---

You use NSM Profiler to learn about your internal network so you can create effective security policies and minimize unnecessary log records. The Profiler queries and correlates attack logs and application usage logs from multiple IDP Series devices.

The Profiler feature is available only through Network and Security Manager (NSM).

IDP Series administration includes the following tasks related to the Profiler feature:

- Configuring Profiler options and preferences
- Starting and stopping the Profiler
- Viewing Profiler reports
- Managing the Profiler database

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring Profiler Options \(NSM Procedure\) on page 204](#)
- [Modifying Profiler Settings on page 210](#)
- [Starting and Stopping the Profiler \(NSM Procedure\) on page 328](#)
- [Using Profiler Viewer \(NSM Procedure\) on page 463](#)
- [Managing the Profiler Database \(NSM Procedure\) on page 328](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

## Configuring Profiler Options (NSM Procedure)

You configure Profiler options to enable Profiler features, set network addresses and applications subject to profiling, and set alerts.

The following topics describe how to configure Profiler options:

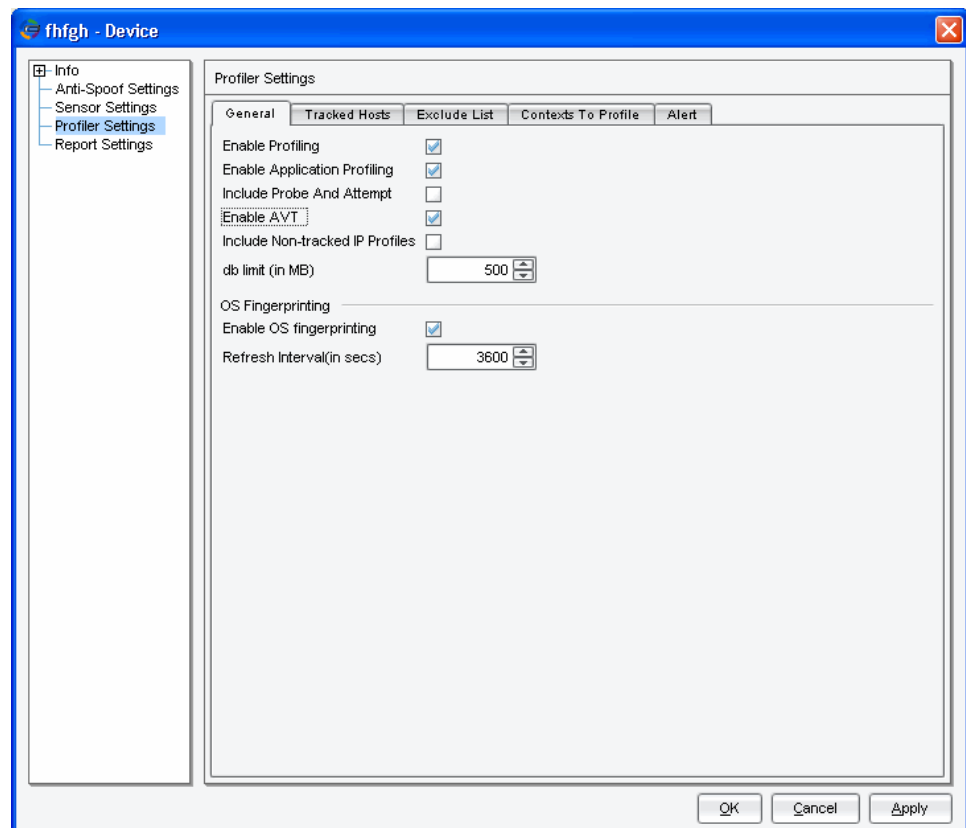
- [Configuring General Settings on page 204](#)
- [Configuring Tracked Hosts on page 205](#)
- [Configuring Context Targets on page 208](#)
- [Configuring Alert Options on page 209](#)

### Configuring General Settings

You use the Profiler Settings > General tab to enable Profiler features.

[Figure 31 on page 134](#) shows the General tab.

**Figure 76: Profiler Settings: Enable AVT**



To configure Profiler general settings:

1. From NSM Device Manager, double-click a device and then click **Profiler Settings**.
2. Click the **General** tab.



3. Configure options.
4. Click **Apply**.



**NOTE:** If you change Profiler settings, you must push a configuration update to the device before the new settings take effect. From the Device Manager, right-click the device, select **Update Device**, select the **Restart IDP Profiler After Device Update** check box, and click **OK**.

Table 54 on page 205 describes settings on the Profiler Settings > General tab.

**Table 54: Profiler Settings: General Tab**

Option	Function
Enable Profiling	Enables the Profiler.
Enable Application Profiling	Enables the Profiler to collect and track application data.  This setting is enabled automatically when you start the Profiler and becomes automatically disabled when you stop the Profiler.
Enable AVT	Enables Profiler to perform application volume tracking.
Include Probe and Attempt	Enables the Profiler to collect and track specific probes and attempts.
Include Non-tracked IP Profiles	Enables context-based profiling for hosts not in the tracked hosts list. If you enable this option, data for non-tracked hosts appears in the Protocol Profiler tab of the Profiler log viewer.
db limit (in MB)	Sets the maximum Profiler database size. By default, the maximum database size is 3 GB.
Enable OS Fingerprinting	Enables the Profiler to perform OS fingerprinting.  OS fingerprinting detects the operating system of a host by analyzing TCP handshake packets.  The OS fingerprinting process depends on an established TCP connection (one that has a SYN, a SYN/ACK, and a FIN connection).  The OS fingerprinting process is capable of detecting the operating systems listed in <code>/usr/idp/device/cfg/fingerprints.set</code> .
Refresh Interval (in secs)	Sets the time interval (in seconds) that the Profiler refreshes OS fingerprinting. By default, the Profiler refreshes OS fingerprinting data every 3600 seconds (60 minutes).

## Configuring Tracked Hosts

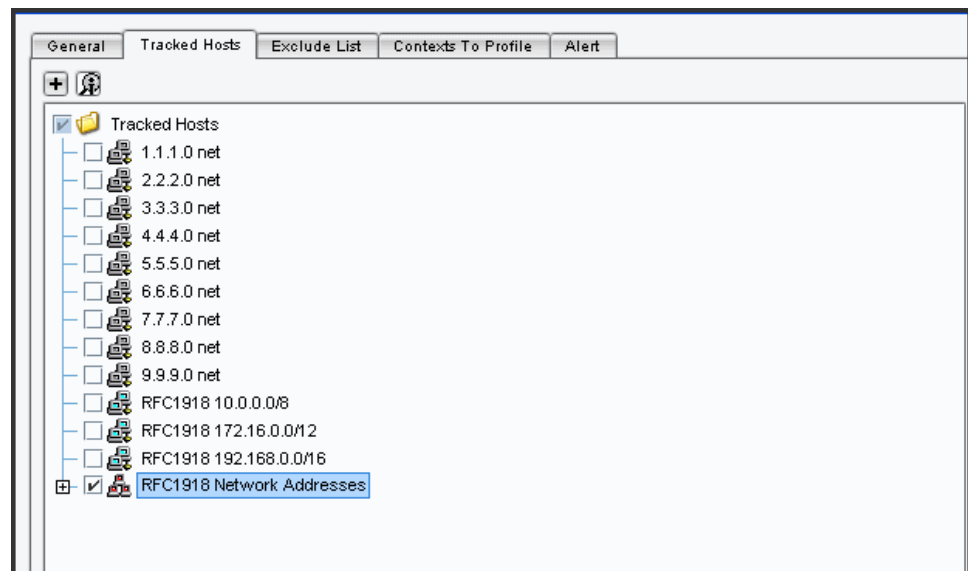
You configure Profiler tracked hosts and excluded host settings to specify the network segments where Profiler gathers data.



**NOTE:** Profiler tracks all traffic through the IDP Series device, including traffic for hosts not in your tracked hosts list. It records a value of 73.78.69.84 for the IP address for hosts not defined in the Tracked Hosts tab, such as external hosts you would not know and therefore could not configure.

Figure 28 on page 126 shows the Tracked Hosts tab.

**Figure 77: NSM Profiler Tracked Hosts Tab**



To configure the tracked hosts and excluded host settings:

1. From NSM Device Manager, double-click a device and then click **Profiler Settings**.
2. Click the **Tracked Hosts** tab.
3. Click the + icon and select one of the following options to display a dialog box to build a tracked host list:
  - **Add Host**
  - **Add Network**
  - **Add Group**
4. Configure tracked host settings as described in [Table 55 on page 207](#).
5. Click the **Exclude** tab.
6. Click the + icon and select one of the following options to display a dialog box to build an exclude host list:
  - **Add Host**
  - **Add Network**

- **Add Group**

7. Configure exclude host settings as described in [Table 55 on page 207](#).

8. Click **Apply**.



**NOTE:** If you change Profiler settings, you must push a configuration update to the device before the new settings take effect. From the Device Manager, right-click the device, select **Update Device**, select the **Restart IDP Profiler After Device Update** check box, and click **OK**.

**Table 55: Profiler Settings: Tracked Hosts or Exclude List**

Option	Function
<b>New Host</b>	
Name	Specifies the hostname.
Color	Specifies a color to help you monitor the host.
Comment	Describes the host.
IP/IP Address	Defines the host using an IP address.
Domain Name/Domain Name	Defines the host using a domain name.
Resolve	Uses DNS to resolve hostnames/IP addresses.
<b>New Network</b>	
Name	Specifies an object name.
IP Address	Specifies an IP address, used with the netmask, that defines the network.
Netmask	Specifies a 32-bit netmask, used with the IP address, that defines the network.
Use Wildcard Mask	Enables use of a wildcard mask.
Wildcard Mask	Specifies the wildcard mask. A wildcard mask is like a subnet mask, with ones and zeros inverted; for example, a wildcard mask of 0.0.0.255 corresponds to a subnet mask of 255.255.255.0.
Color	Specifies a color to help you monitor the network.
Comment	Describes the network.
<b>New Address Group</b>	
Name	Specifies an object name.
Color	Specifies a color to help you monitor the address group.

Table 55: Profiler Settings: Tracked Hosts or Exclude List (*continued*)

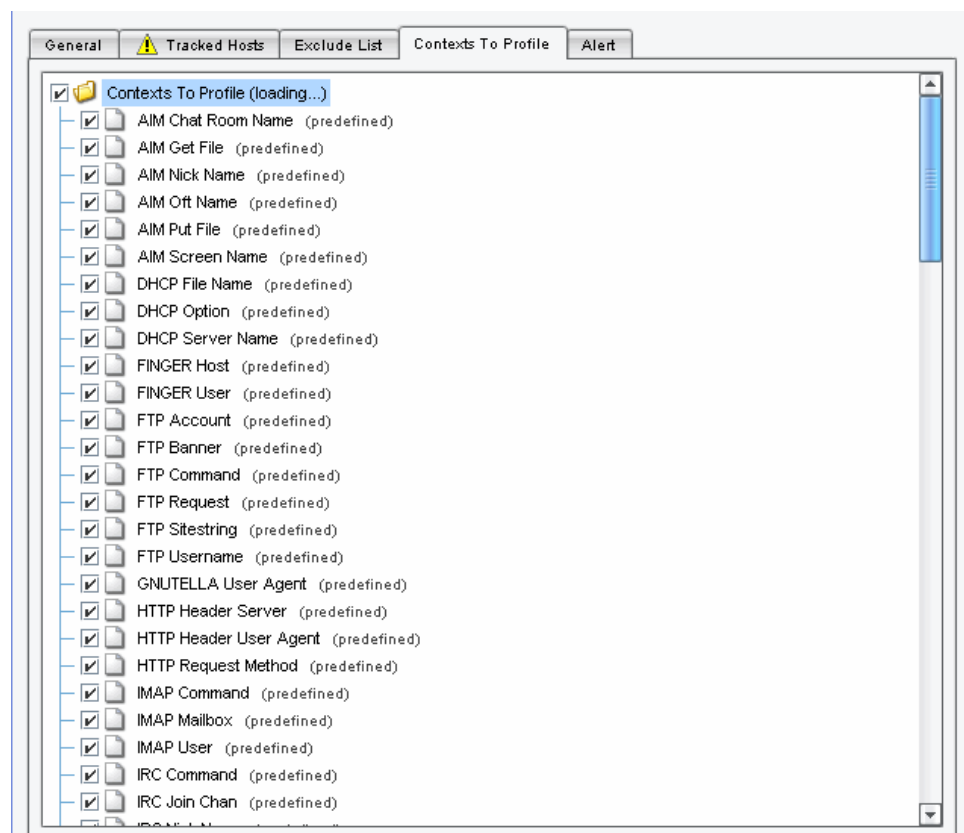
Option	Function
Comment	Describes the group.
Member List	Adds hosts that belong to the group.

## Configuring Context Targets

You configure Profiler context settings to determine whether Profiler logs include not only host and application data but also data pulled from application contexts. For example, if you specify context targets for FTP usernames, the Profiler logs will include the username specified for the FTP connection in addition to the hostname and service (FTP).

Figure 78 on page 208 shows the Contexts to Profile tab.

Figure 78: NSM Profiler Context to Profile Tab



To specify Profiler context targets:

1. From NSM Device Manager, double-click a device and then click **Profiler Settings**.
2. Click the **Contexts to Profile** tab.

3. Browse and select from the predefined list of contexts.
4. Click **Apply**.



**NOTE:** If you change Profiler settings, you must push a configuration update to the device before the new settings take effect. From the Device Manager, right-click the device, select **Update Device**, select the **Restart IDP Profiler After Device Update** check box, and click **OK**.

## Configuring Alert Options

You configure Profiler alert options to determine whether you receive alerts when Profiler detects new hosts, protocols, or ports in use.

If you are configuring the Profiler for the first time, do not enable the new host, protocol, or port alerts. As the Profiler runs, the device views all network components as new, which can generate unnecessary log records. After the Profiler has learned about your network and has established a baseline of network activity, you should reconfigure the device to record new hosts, protocols, or ports discovered on your internal network.

Figure 79 on page 209 shows the Alert tab.

**Figure 79: Profiler Alert Tab**

Profiler Settings	
<a href="#">General</a> <a href="#">Tracked Hosts</a> <a href="#">Exclude List</a> <a href="#">Contexts To Profile</a> <a href="#">Alert</a>	
New Host Detected	<input type="checkbox"/>
New Protocol Detected	<input type="checkbox"/>
New Port Detected	<input type="checkbox"/>
Database Limit Exceeded	<input type="checkbox"/>

To specify Profiler alert options:

1. From NSM Device Manager, double-click a device and then click **Profiler Settings**.
2. Click the **Alert** tab.
3. Configure alert settings as described in Table 56 on page 210.
4. Click **Apply**.



**NOTE:** If you change Profiler settings, you must push a configuration update to the device before the new settings take effect. From the Device Manager, right-click the device, select **Update Device**, select the **Restart IDP Profiler After Device Update** check box, and click **OK**.

Table 56: Profiler Alert Tab

Option	Function
New Host Detected	Sends an alert when Profiler detects a new host.
New Protocol Detected	Sends an alert when Profiler detects a new protocol.
New Port Detected	Sends an alert when Profiler detects a new port.
Database Limit Exceeded	Sends an alert to indicate the maximum database size has been reached. After a device reaches this limit, it begins purging the database.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

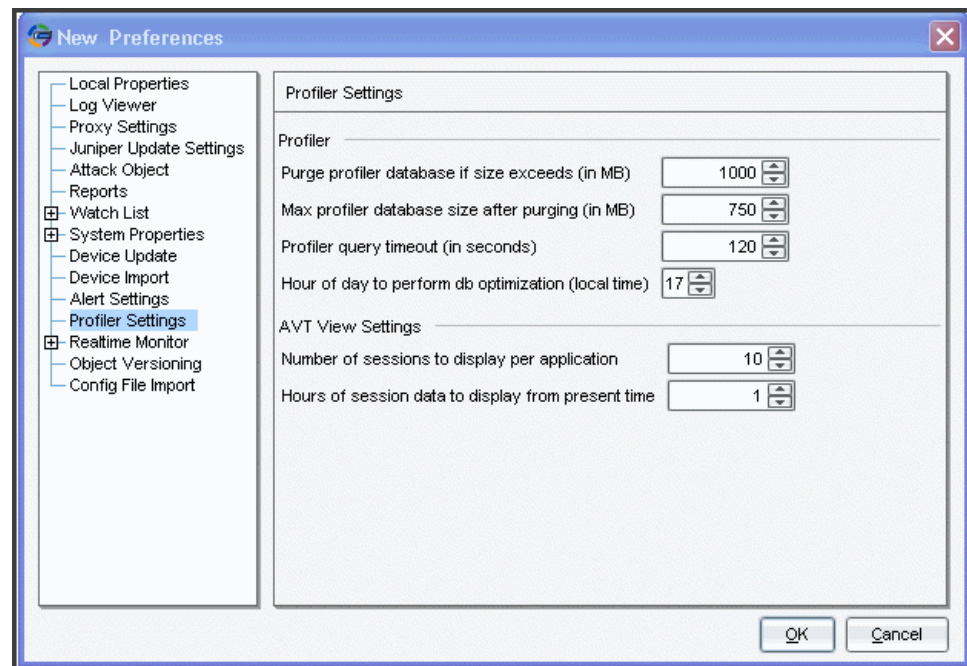
- [Profiler Task Summary on page 203](#)

## Modifying Profiler Settings

You can use the Profiler Settings dialog box to modify Profiler database settings and default settings for application volume tracking reports.

[Figure 80 on page 210](#) shows the New Preferences dialog box, where you can modify Profiler default settings.

Figure 80: New Preferences Dialog Box



To modify Profiler database and application volume tracking settings:

1. From the NSM main menu, select **Tools > Preferences**.
2. Click **Profiler Settings**.
3. Modify settings as described in [Table 57 on page 211](#).
4. Click **OK**.

**Table 57: Profiler Settings**

Setting	Description
<b>Profiler</b>	
Purge profiler database if size exceeds	Default is 1000 MB.
Max profiler database size after purging	Default is 750 MB.
Profiler query timeout	Default is 120 seconds.
Hour of day to perform database optimization	Default is 00:00 GMT.
<b>AVT View Settings</b>	
Number of sessions to display per application	<p>Determines the number of sessions displayed in the Application Profiler application volume tracking session tables.</p> <p>Default is 10 sessions. You can specify from 5 to 10,000 sessions.</p>
Hours of session data to display from present time	<p>Determines the hours of application volume tracking data displayed in the Application Profiler tab session tables.</p> <p>Default is 1 hour. You can specify from 1 to 24 hours.</p> <p>This setting is also a data retention policy. By default, data older than 1 hour is deleted. If you change to 12 hours, data older than 12 hours is deleted.</p>

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)





## CHAPTER 24

# Configuring the IDP Rulebase

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)
- [Specifying Rule Match Conditions \(NSM Procedure\) on page 215](#)
- [Specifying IDP Rulebase Attack Objects \(NSM Procedure\) on page 216](#)
- [Specifying Rule Session Action \(NSM Procedure\) on page 218](#)
- [Specifying IP Action \(NSM Procedure\) on page 220](#)
- [Specifying Rule Notification Options \(NSM Procedure\) on page 221](#)
- [Specifying Rule VLAN Matches \(NSM Procedure\) on page 223](#)
- [Specifying Rule Targets \(NSM Procedure\) on page 223](#)
- [Specifying Rule Severity \(NSM Procedure\) on page 224](#)
- [Specifying Rule Comments \(NSM Procedure\) on page 225](#)

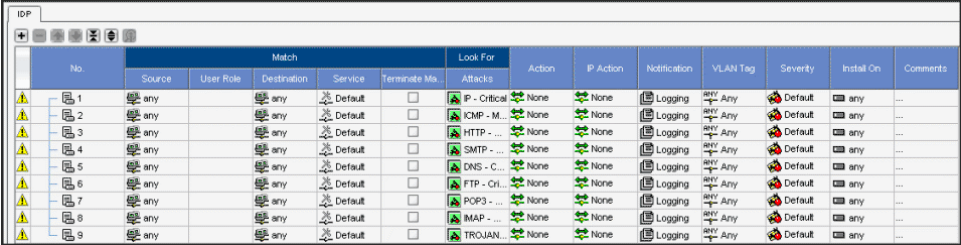
## Modifying IDP Rulebase Rules (NSM Procedure)

This procedure assumes you have used the New Policy wizard to create a basic policy that you can modify.

The primary IDP security policy rulebase is the IDP rulebase. The IDP rulebase enables the IDP engine to inspect matching traffic for attack signatures and protocol anomalies.

[Figure 81 on page 213](#) shows the IDP rulebase in the NSM security policy editor, where you can modify IDP rulebase rules. [Table 58 on page 214](#) lists the rule properties you can modify and provides references to documentation for these properties.

**Figure 81: NSM Security Policy Editor: IDP Rulebase**



No.	Source	User Role	Destination	Service	Terminate Ma.	Look For	Action	IP Action	Notification	VLAN Tag	Severity	Install On	Comments
1	any		any	Default		IP - Critical	None	None	Logging	Any	Default	any	...
2	any		any	Default		ICMP - M...	None	None	Logging	Any	Default	any	...
3	any		any	Default		HTTP - ...	None	None	Logging	Any	Default	any	...
4	any		any	Default		SMTP - ...	None	None	Logging	Any	Default	any	...
5	any		any	Default		DNS - C...	None	None	Logging	Any	Default	any	...
6	any		any	Default		FTP - Cri...	None	None	Logging	Any	Default	any	...
7	any		any	Default		POP3 - ...	None	None	Logging	Any	Default	any	...
8	any		any	Default		MAP - ...	None	None	Logging	Any	Default	any	...
9	any		any	Default		TROJAN...	None	None	Logging	Any	Default	any	...

To modify an IDP rulebase rule property:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Double-click the security policy you want to edit.
3. In the security policy editor, click the **IDP** tab to display the IDP rulebase table.
4. Add, delete, copy, or reorder rules by right-clicking the table cell for the rule number (No. column) and making your selection.
5. Modify rule settings by right-clicking the table cell for the setting and making your selection.
6. Click **OK** to save your changes.

**Table 58: IDP Rulebase Rule Properties**

Property	Reference
Match	"Specifying Rule Match Conditions (NSM Procedure)" on page 215
Look For	"Specifying IDP Rulebase Attack Objects (NSM Procedure)" on page 216
Action	"Specifying Rule Session Action (NSM Procedure)" on page 218
IP Action	"Specifying IP Action (NSM Procedure)" on page 220
Notification	"Specifying Rule Notification Options (NSM Procedure)" on page 221
VLAN Tag	"Specifying Rule VLAN Matches (NSM Procedure)" on page 223
Severity	"Specifying Rule Severity (NSM Procedure)" on page 224
Install On	"Specifying Rule Targets (NSM Procedure)" on page 223
Comments	"Specifying Rule Comments (NSM Procedure)" on page 225



**NOTE:** If not all of the columns you want to configure appear in the Security Policy editor, use NSM display features to show the hidden columns. For details, see the NSM online Help.

#### **Related Documentation**

The following additional related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

The following additional related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the IDP Rulebase on page 55](#)

## Specifying Rule Match Conditions (NSM Procedure)

The IDP engine inspects the session beginning with the first packet to determine whether the session matches a rule. If the session matches all rule settings for source, destination, service, and VLAN tag ID, the IDP engine decodes the traffic and inspects the session packets for the attack objects specified in the rule. If the first packet matches only some of the rule settings, the rule is not a match. [Table 59 on page 215](#) describes match condition columns for IDP rulebase rules.

To modify rule match settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Modify a rule match setting by right-clicking the table cell for the setting and making your selection.
5. Click **OK** to save your changes.

**Table 59: IDP Rulebase Match Condition Settings**

Column	Description
From zone/To zone	Not applicable for IDP Series appliances.
Source	<p><b>Select Address</b>—Displays the Select Source Address dialog box where you can select or configure address objects for traffic sources.</p> <hr/> <p><b>Any</b>—Turns off matching on source IP address. To guard against incoming attacks, which might come from anywhere, you typically specify <b>Any</b>.</p> <hr/> <p><b>Negate</b>—Matches any except those specified.</p> <p>To use address negation:</p> <ol style="list-style-type: none"> <li>1. Add the address object.</li> <li>2. Right-click the address object and select <b>Negate</b>.</li> </ol>
User Role	<p><b>Select User Role</b>—Displays the Select User Role dialog box where you can select or configure user role matches.</p> <p>You must choose to configure either source IP address or user role as match criteria for a rule. User role-based rules are evaluated before IP address-based rules. If a user-role based rule matches, the rule is applied and IP address-based rules are not consulted.</p> <p><b>NOTE:</b> Matching based on user role depends on integration with a compatible Juniper Networks IC Series Unified Access Control appliance.</p>

Table 59: IDP Rulebase Match Condition Settings (*continued*)

Column	Description
Destination	<p><b>Select Address</b>—Displays the Select Destination Address dialog box where you can select or configure address objects for destination servers.</p> <hr/> <p><b>Any</b>—Turns off matching based on destination IP address.</p> <hr/> <p><b>Negate</b>—Specifies any except those specified.</p> <p>To use address negation:</p> <ol style="list-style-type: none"> <li>1. Add the address object.</li> <li>2. Right-click the address object and select <b>Negate</b>.</li> </ol>
Service	<p><b>Default</b>—Matches the service(s) specified in the rule attack object(s).</p> <p>With the application identification feature enabled, the IDP engine identifies services even if they are running on nonstandard ports. The application identification feature is enabled by default.</p> <p>If you have disabled application identification and specify <b>Default</b>, the IDP engine assumes that standard ports are used for the service.</p> <p><b>NOTE:</b> If you disable application identification and your service uses nonstandard ports, you must create a custom service object.</p> <hr/> <p><b>Any</b>—Turns off matching based on service.</p> <hr/> <p><b>Select Service</b>—Displays the Select Services dialog box where you can select predefined or custom service objects.</p>
Terminate Match	<p>Select this option to mark the rule as a terminal rule. If a session matches a terminal rule, the IDP engine does not process any subsequent rules. It takes action, if any, according to the terminal rule.</p>

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Rule Match Settings on page 56](#)
- [User-Role-Based Policy Feature Overview on page 58](#)
- [Using Application Identification on page 43](#)

## Specifying IDP Rulebase Attack Objects (NSM Procedure)

Attack objects are the signatures and protocol anomalies the IDP engine looks for in traffic that matches the rule. In general, you specify attack objects related to the service and destination server set for the rule.

To add attack objects:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Right-click the table cell for attacks and select **Select Attacks**.
5. In the All Attacks/Groups box, expand **Attack Groups** and add attack objects:
  - To add attack objects recommended by Juniper Networks Security Center (J-Security Center), expand **Recommended Attacks**. Then browse groups and select groups or individual attack objects. [Table 60 on page 217](#) describes the hierarchy of recommended attack groups.
  - To add other predefined attack objects, expand **All Attacks**. Then browse groups and select groups or individual attack objects. [Table 60 on page 217](#) describes the hierarchy of predefined attack groups.
  - To add attack objects that belong to custom groups, expand the node for the custom group. Then browse subgroups and select groups or individual attack objects.
  - To add custom attack objects that do not belong to groups, expand **Attack List**. Then select from custom attack objects.
6. Click **OK** to save your changes.

**Table 60: Attack Object Group Hierarchy**

Group	Contents
Attack Type	Contains two subgroups: anomaly and signature. Within each subgroup, attack objects are grouped by severity.
Category	Contains subgroups based on category. Within each category, attack objects are grouped by severity.
Operating System	Contains the following subgroups: BSD, Linux, Solaris, and Windows. Within each operating system, attack objects are grouped by services and severity.
Severity	Contains the following subgroups: Critical, Major, Minor, Warning, Info. Within each severity, attack objects are grouped by category.  <b>NOTE:</b> Our severity rating is not based on CVSS (Common Vulnerability Scoring System). We do include data from Bugtraq (Symantec) and CVE (Common Vulnerabilities and Exposures).
Web Services	Contains subgroups based on Web services. Within services, attacked objects are grouped by severity.
Miscellaneous	Contains attack objects that have a significant affect on performance.
Response	Contains attack objects that are relevant to server-to-client traffic. This group contains a hierarchy of subgroups that includes all of the above categories.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)
- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)

## Specifying Rule Session Action (NSM Procedure)

Actions are responses to sessions that match the source/destination condition and attack object pattern. Actions are what protect your network from attacks.

If a packet triggers multiple rule actions, the IDP engine takes the most severe action. For example, if a rule with a DiffServ marking action and a rule with a drop action both match, the IDP engine takes the drop action.

[Table 23 on page 64](#) describes the actions you can set for IDP rulebase rules.

To modify action settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Modify action settings by right-clicking the table cell and selecting your setting.
5. Click **OK** to save your changes.

**Table 61: IDP Rulebase Actions**

Action	Function
Recommended	Takes the action recommended in the predefined attack object. The recommended action is related to severity. <a href="#">Figure 51 on page 157</a> lists the recommended actions by severity.
None	Inspects the session but takes no action against the connection.
Ignore	Ignores the match and does not inspect the remainder of the connection.
Drop Packet	Drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a denial of service (DoS) condition that prevents you from receiving traffic from a legitimate source address.  <b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. Therefore, this action has no effect in sniffer mode.

Table 61: IDP Rulebase Actions (*continued*)

Action	Function
Drop Connection	<p>Drops the connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic not prone to spoofing.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. Therefore, this action has no effect in sniffer mode.</p>
Close Client	<p>Closes the connection to the client but not to the server.</p> <p>In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the device can send an RST packet to both the client and server but does not close the connection.</p> <p><b>NOTE:</b> In VLAN tagged MPLS traffic, the Close Client action drops the connection instead of closing it.</p>
Close Server	<p>Closes the connection to the server but not to the client.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the device can send an RST packet to both the client and server but does not close the connection.</p>
Close Client and Server	<p>Closes the connection and sends an RST packet to both the client and the server.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. However, if you use ACM to configure a sniffer mode reset interface, the device can send an RST packet to both the client and server but does not close the connection.</p>
Diffserv Marking	<p>Assigns the indicated service-differentiation value to the packet, and then passes it on normally. Set the service-differentiation value in the dialog box that appears when you select this action in the rulebase.</p> <p><b>NOTE:</b> In sniffer mode, the IDP Series device is not in the path of network traffic. Therefore, this action has no effect in sniffer mode.</p>

Figure 51 on page 157 describes the logic applied to the value Recommended, a setting coded in predefined attack objects provided by Juniper Networks Security Center.

Table 62: IDP Rulebase Actions: Recommended Actions by Severity

Severity	Description	Recommended Action
Critical	Attacks attempt to evade an intrusion prevention system, crash a machine, or gain system-level privileges.	Drop Packet, Drop Connection
Major	Attacks attempt to crash a service, perform a denial of service, install or use a Trojan, or gain user-level access to a host.	Drop Packet, Drop Connection
Minor	Attacks attempt to obtain critical information through directory traversal or information leaks.	None
Warning	Attacks attempt to obtain noncritical information or scan the network. They can also be obsolete attacks.	None

Table 62: IDP Rulebase Actions: Recommended Actions by Severity (*continued*)

Severity	Description	Recommended Action
Info	Attacks are normal, harmless traffic containing URLs, DNS lookup failures, and SNMP public community strings. You can use informational attack objects to obtain information about your network.	None



**NOTE:** Our severity rating is not based on CVSS (Common Vulnerability Scoring System). We do include data from Bugtraq (Symantec) and CVE (Common Vulnerabilities and Exposures).

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)
- [Specifying IP Action \(NSM Procedure\) on page 220](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Actions on page 63](#)

## Specifying IP Action (NSM Procedure)

If the IDP Series device matches an attack, it can take action not only against the current session but also against future network traffic that uses the same IP address. Such actions are called *IP actions*. By default, the IP action persists permanently (timeout = 0). If you prefer, you can set a timeout period in seconds. [Table 63 on page 221](#) describes IDP rulebase actions.

To modify settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Modify IP action settings by right-clicking the table cell for the setting and making your selection.
5. Click **OK** to save your changes.



Table 63: IDP Rulebase IP Actions

IP Action	Function
IP Block	<p>Blocks the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP Address</li> <li>• Source Subnet</li> <li>• Protocol</li> <li>• Destination IP Address</li> <li>• Destination Subnet</li> <li>• Destination Port</li> <li>• From Zone</li> </ul> <p><b>NOTE:</b> You can reset the IP block table when a security policy is (re)loaded. In NSM Device Manager, select Sensor Settings &gt; Run-Time Parameters and select the <b>Reset block table with policy load/unload</b> option.</p>
IP Close	<p>Closes the matching connection and future connections that match combinations of the following properties you specify:</p> <ul style="list-style-type: none"> <li>• Source IP Address</li> <li>• Source Subnet</li> <li>• Protocol</li> <li>• Destination IP Address</li> <li>• Destination Subnet</li> <li>• Destination Port</li> <li>• From Zone</li> </ul> <p><b>NOTE:</b> The IP Close action might not work as expected for MPLS traffic. In MPLS traffic, when a rule triggers an IP Close action, the IDP engine cannot send a TCP reset packet to the source with a correct server-to-client label. The IDP engine sends a TCP reset packet without an MPLS label. Some MPLS routers can add packets without a label to an existing MPLS tunnel; others drop such packets.</p>
IP Notify	Logs the event or sends an alert.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)
- [Specifying Rule Session Action \(NSM Procedure\) on page 218](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Actions on page 63](#)

## Specifying Rule Notification Options (NSM Procedure)

Notification options determine how events that match the rule are logged. [Table 25 on page 66](#) describes IDP rulebase notification options.

To modify notification settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Modify notification settings by right-clicking the table cell for the setting and making your selection.
5. Click **OK** to save your changes.

**Table 64: IDP Rulebase Notification Options**

Option	Function
Event logs and alerts	<p>Enables or disables the following delivery and handling options for logs:</p> <ul style="list-style-type: none"> <li>• Send to NSM Log Viewer</li> <li>• Send to NSM Log Viewer and flag as an alert</li> <li>• Send to an e-mail address list</li> <li>• Send to syslog</li> <li>• Send to SNMP trap</li> <li>• Save in XML format</li> <li>• Save in CVS format</li> <li>• Process with a script</li> </ul>
Packet captures	<p>Enables packet capture. Viewing the packets used in an attack on your network can help you determine the extent of the attempted attack, its purpose, whether or not the attack was successful, and any possible damage to your network.</p> <p>If multiple rules with packet capture enabled match the same attack, the IDP engine captures the maximum specified number of packets. For example, you configure rule 1 to capture 10 packets before and after the attack, and you configure rule 2 to capture 5 packets before and after the attack. If both rules match the same attack, the IDP engine attempts to capture 10 packets before and after the attack.</p> <p>You can capture up to 256 packets before the event and 256 packets after the event.</p> <p><b>NOTE:</b> If necessary, you can improve performance by logging only the packets received after the attack.</p>

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Notification Options on page 65](#)

## Specifying Rule VLAN Matches (NSM Procedure)

If you deploy an IDP Series device in a virtual local area network (VLAN), you can configure rules that require a match of VLAN tag. [Table 65 on page 223](#) describes VLAN tag settings.

To modify VLAN match settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Modify VLAN match settings by right-clicking the table cell for the setting and making your selection.
5. Click **OK** to save your changes.

**Table 65: IDP Rulebase VLAN Tag Settings**

Option	Function
None	Matches only traffic that has no VLAN tag.
Any	Turns off matching on VLAN tag.
Select VLAN Tags	Displays the Select VLAN Tags dialog box where you can set a single VLAN tag or a range of VLAN tags.
Delete VLAN Tags	Displays a dialog box that prompts you to confirm you want to delete the VLAN tag match setting.

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Rule Match Settings on page 56](#)

## Specifying Rule Targets (NSM Procedure)

By default, IDP security policy rules can be applied to any IDP Series device. If you want, you can specify that the rule applies to only specified IDP Series devices.

To modify rule targets:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.

4. Modify rule target settings by right-clicking the table cell for the setting and making your selection.
5. Click **OK** to save your changes.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)
- [Assigning a Security Policy to a Device \(NSM Procedure\) on page 335](#)

## Specifying Rule Severity (NSM Procedure)

Severity is a rating of the danger posed by the threat the rule is designed to prevent. [Table 66 on page 224](#) describes rule severity settings.

To modify severity settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Modify severity settings by right-clicking the table cell for the setting and making your selection.
5. Click **OK** to save your changes.

**Table 66: IDP Rulebase Severity**

Severity	Guideline
Default	Select <b>Default</b> to inherit severity from that specified in the attack object.
Critical	Attacks that attempt to evade an IPS, crash a machine, or gain system-level privileges. We recommend that you drop the packets or drop the connection for such attacks.
Major	Attacks that attempt to crash a service, perform a denial of service, install or use a Trojan, or gain user-level access to a host. We recommend that you drop the packets or drop the connection for such attacks.
Minor	Attacks that attempt to obtain critical information through directory traversal or information leaks. We recommend that you log such attacks.
Warning	Attacks that attempt to obtain noncritical information or scan the network. They can also be obsolete attacks (but probably harmless) traffic. We recommend that you log such attacks.

Table 66: IDP Rulebase Severity (*continued*)

Severity	Guideline
Info	<p>Attacks that are normal, harmless traffic containing URLs, DNS lookup failures, and SNMP public community strings. You can use informational attack objects to obtain information about your network.</p> <p>We recommend that you log such attacks.</p>



**NOTE:** Our severity rating is not based on CVSS (Common Vulnerability Scoring System). We do include data from Bugtraq (Symantec) and CVE (Common Vulnerabilities and Exposures).

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)

## Specifying Rule Comments (NSM Procedure)

Comments are notations about the rule. Comments do not affect the functionality of the security policy rule.

To modify severity settings:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to edit.
3. In the security policy pane, click the **IDP** tab to display the IDP rulebase table.
4. Add a comment by right-clicking the table cell for comments and selecting **Edit Comments** to display the Edit Comments dialog box, where you can enter a comment up to 1024 characters in length.
5. Click **OK** to save your changes.

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying IDP Rulebase Rules \(NSM Procedure\) on page 213](#)



## CHAPTER 25

# Configuring Additional Security Policy Rulebases

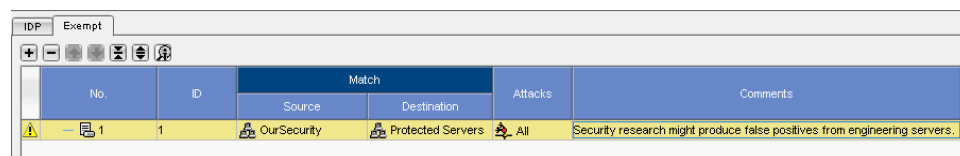
- [Configuring Exempt Rulebase Rules \(NSM Procedure\) on page 227](#)
- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Configuring Backdoor Rulebase Rules \(NSM Procedure\) on page 233](#)
- [Configuring SYN Protector Rulebase Rules \(NSM Procedure\) on page 235](#)
- [Configuring Traffic Anomalies Rulebase Rules \(NSM Procedure\) on page 237](#)
- [Configuring Network Honeypot Rulebase Rules \(NSM Procedure\) on page 240](#)

### Configuring Exempt Rulebase Rules (NSM Procedure)

The Exempt rulebase enables you to categorically exempt traffic segments you know to be safe from IDP rulebase processing.

[Figure 71 on page 175](#) shows the Exempt rulebase in the NSM security policy editor, where you can modify Exempt rules. [Table 67 on page 228](#) describes the rule settings you can configure.

**Figure 82: NSM Security Policy Editor: Exempt Rulebase**



No.	ID	Match		Attacks	Comments
		Source	Destination		
1	1	OurSecurity	Protected Servers	All	Security research might produce false positives from engineering servers.

To create Exempt rulebase rules:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy you want to which you want to add Exempt rulebase rules.
3. Add the Exempt rulebase by clicking the + icon in the upper right region of the policy viewer and selecting **Add Exempt Rulebase**.
4. Add a rule by clicking the + icon within the rules viewer.

5. Modify rule settings by right-clicking the table cell for the setting and making your selection.
6. Click **OK** to save your changes.

**Table 67: Exempt Rulebase Rule Properties**

Setting	Function
No.	Adds, deletes, copies, or reorders rules. Right-click the table cell for the rule number and make your selection.
Match	Sets source, destination, and service matches.
Look For	Sets attack matches.
VLAN Tag	Sets VLAN tag matches.
Install On	Specifies target IDP Series devices for the rule. By default, IDP security policy rules can be applied to any IDP Series device. Right-click the table cell and select <b>Select Target</b> to display a dialog box where you can specify the IDP Series devices to which the rule can be installed.
Comments	Adds notations about the rule. This setting is optional. Right-click the table cell and select <b>Edit Comments</b> to display a dialog box where you can make notations about the rule. Comments do not affect the functionality of the security policy rule.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Exempt Rulebase on page 67](#)

## Configuring the APE Rulebase (NSM Procedure)

The application policy enforcement (APE) rulebase triggers actions based on the application detected.

[Figure 83 on page 228](#) shows the APE rulebase in the NSM security policy editor, where you can modify APE rules. [Table 68 on page 229](#) describes the rule settings you can configure.

**Figure 83: NSM Security Policy Editor: APE Rulebase**

No.	Match						Action	Notification	VLAN Tag	Severity	Install On	Comments
	Source	User Role	Destination	Service	Application	Extended Application						
1	any		any	Default	any	any	None	Logging	Any	Default	any	...

To create APE rulebase rules:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy to which you want to add APE rulebase rules.



3. Click the + icon in the upper right region of the Security Policy viewer and select **Add Application Rulebase** to add the rulebase to the security policy.
4. Click the + icon within the rules viewer to add a blank rule.
5. Right-click the table cell for a setting and display an editor for the setting. Use the controls to complete the configuration.
6. Click **OK** to save your changes.



**TIP:** In NSM, you can jump from the Application Profiler tab to the Security Policy Editor by right-clicking the application name in the Profiler log and selecting a new or existing security policy. For details, see the NSM online Help.

**Table 68: APE Rulebase Rule Properties**

Setting	Function
No.	<p>Adds, deletes, copies, or reorders rules. Right-click the table cell for the rule number and make your selection.</p> <p>The APE rulebase is a terminal rulebase. Rules are evaluated in numerical order. The first rule to match is applied, and subsequent rules are not processed.</p>
Match	<p>The matching tuple for APE rules includes the following elements:</p> <ul style="list-style-type: none"> <li>• Source or user role</li> <li>• Destination</li> <li>• Service or the combined list of applications and extended applications</li> <li>• VLAN tag</li> </ul> <p>The Boolean logic of the matching tuple is as follows:</p> <p>(src OR user role) AND destination AND vlan AND (service OR application list)</p> <p><b>NOTE:</b> You can use the <b>Any</b> wildcard to “remove” a property from the tuple. For example, if you specify <b>Any</b> for source, destination, or VLAN tag, you are creating a “traffic lane” that treats all traffic matching the specified application the same. However, <b>Any</b> has a different significance when building the service or application list. When setting service or application guidelines, be sure to follow the guidelines below.</p> <p><b>Source</b>—Requires a match of one of the specified source IP addresses. You can add address objects for hosts, groups, or network address ranges.</p> <p><b>NOTE:</b> If a value for User Role matches, the Source parameter is not used.</p> <p><b>User Role</b>—Requires a match of one of the specified user roles. If a value for User Role matches, the Source parameter is not consulted.</p> <p>Matching based on user role depends on integration with a compatible Juniper Networks IC Series UAC appliance.</p> <p><b>Destination</b>—Requires a match of one of the specified destination IP addresses. You can add address objects for hosts, groups, or network address ranges.</p>

Table 68: APE Rulebase Rule Properties (*continued*)

Setting	Function
	<p><b>Service</b>—Requires a match of one of the specified services.</p> <p>A single rule can match a service object definition or an application list, but not both. We recommend you create rules that match an application list whenever possible. Matching based on application uses the application identification feature, which can identify the application regardless of port. We support rules that match service object definitions for cases where there is not a suitable application object.</p> <p>If your rule includes application or extended application objects, specify <b>Default</b> for the service parameter.</p> <p>If you do not want to match on service or application list, specify <b>Any</b> for all three (service, application, and extended application).</p> <p>If there are no suitable application objects, create a rule that uses the service object and set the application and extended application columns to <b>Any</b>.</p> <p>If the service uses standard ports, you can select from predefined services. If the service uses nonstandard ports, you can create a custom service object. The IDP engine can inspect services that use TCP, UDP, RPC, and ICMP transport layer protocols.</p>
	<p><b>Application</b>—Requires one of the specified applications to match the session for the rule to be applied.</p> <p>You use the Application and Extended Application columns to build a list of applications to match the rule. You can specify individual applications or application groups. When you add a group, you are in effect adding its members to the list. The group object itself is not evaluated. The list is evaluated as a Boolean OR, so if one of the application or extended application objects specified in the rule is identified, the “service or application” component of the tuple matches. If any application or member of a group matches, the rule matches.</p> <p>The predefined list of applications is populated by the application signatures included in J-Security Center signature updates. The application identification feature uses both heuristic methods and signature pattern matching to identify the application regardless of port. Port-independent application identification simplifies rule configuration and ensures that you do not miss applications that are running on nonstandard ports. For this reason, we recommend that you use the application parameter instead of the service parameter whenever possible.</p> <p>Specify <b>Any</b> in the Application column when creating a service-based rule or when creating an application-based rule where the application list consists only of extended application objects.</p> <p>You can use the Shared Objects for Policy viewer (located below the rule editor) to browse application objects and explore object properties. You can create custom application objects.</p> <p><b>NOTE:</b> Extended application matching is more granular than application matching. Do not select HTTP in the application column if you also plan to specify extended application objects in the same rule. If you specify HTTP and HTTP:Facebook, for example, the rule matches HTTP or HTTP:Facebook. The result is indistinguishable from a rule matching only HTTP. We recommend you list rules targeting Extended Applications before a rule targeting HTTP.</p>

Table 68: APE Rulebase Rule Properties (*continued*)

Setting	Function
	<p><b>Extended Application</b>—Requires one of the specified <i>extended applications</i> to match the session for the rule to be applied. Extended applications are also called <i>nested applications</i>. The Juniper Networks Security Center (J-Security Center) provides predefined application signatures for many Web 2.0 applications running over HTTP. Matching on these signatures depends on the application identification feature, which is enabled by default.</p> <p>You use the Application and Extended Application columns to build a list of applications to match the rule. The list is evaluated as a Boolean OR, so if one of the application or extended application objects specified in the rule is identified, the “service or application” component of the tuple matches.</p> <p>Specify <b>Any</b> in the Extended Application column when you are creating a service-based rule or when you are creating an application-based rule where the application list consists only of application objects.</p> <p>You can use the Shared Objects for Policy viewer (located below the rule editor) to browse extended application objects and explore object properties. You cannot create custom extended application objects.</p>
	<p><b>VLAN Tag</b>—Requires a match of one of the specified VLAN tags.</p>

Table 68: APE Rulebase Rule Properties (*continued*)

Setting	Function
Action	<p><b>None</b>—Does not perform rate limiting. Logs that are generated for traffic that match this rule display <b>Accepted</b>.</p> <hr/> <p><b>Drop Connection</b>—Drops the connection without sending an RST packet to the sender, thereby preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.</p> <hr/> <p><b>Close Client</b>—Closes the connection to the client but not to the server.</p> <hr/> <p><b>Close Server</b>—Closes the connection to the server but not to the client.</p> <hr/> <p><b>Close Client and Server</b>—Closes the connection and sends an RST packet to both the client and the server. If the IDP Series device is in sniffer mode, it sends an RST packet to both the client and server but does not close the connection.</p> <hr/> <p><b>DiffServ Marking</b>—Assigns the DiffServ value you specify to the packet. This action is useful when your network has a class of service (CoS) design, and you want to use the IDP Series device to rewrite the CoS code point based on APE rules processing. The CoS rules you have implemented for the next devices in the network path ultimately determine the effect on the transmission rate.</p> <p><b>NOTE:</b> In sniffer mode, this action has no effect because the IDP Series device is not in the path of network traffic.</p> <hr/> <p><b>Rate Limiting</b>—Enforces a rate limit for all current sessions that match the rule. If a session matches an APE rule in which a rate limit has been set, the IDP engine performs a rate-limit check. If the limit is not reached, the IDP Series device forwards the packets. If the limit is reached, the IDP Series device behaves as if no bandwidth is available: it drops packets until the aggregate bandwidth falls below the limit. When the IDP Series device drops packets, the TCP or UDP endpoints identify the packet loss and slow the transmission rate.</p> <p>The rate limits that are best suited for your business case depend on the bandwidth for your links. If you have a 1-Gbps link and want no more than 10% available to peer-to-peer traffic, the sum of the rate limits you specify for all peer-to-peer rules must be less than 102.4 Mbps (in each direction).</p> <p>If you implement user-role-based rules, you can apply rate limiting to all users who belong to the specified role or to individual users who belong to the specified role. By default, rate limiting is applied to all users who belong to the specified role. In this case, you would configure a larger limit. You can change this setting with the command-line interface. If you change the default to enable rate limiting per user, configure a smaller limit.</p> <p>You configure separate rate limits for client-to-server and server-to-client directions. For peer-to-peer traffic, we recommend that you set the same rate for each direction.</p> <p><b>NOTE:</b> For TFTP traffic, all traffic is considered client-to-server traffic. A TFTP server responds to get requests by establishing an ephemeral port from which to send the reply. In this case, both directions appear to the IDP Series device as client-to-server flows. We recommend you set the same rate for each direction.</p> <p>Logs generated for traffic that match this rule display <b>Rate Limit</b> and traffic direction (c2s or s2c).</p> <p><b>NOTE:</b> In sniffer mode, this action has no effect because the IDP Series device is not in the path of network traffic.</p> <hr/> <p><b>DiffServ Marking &amp; Rate Limiting</b>—Takes both actions described above.</p>

Table 68: APE Rulebase Rule Properties (*continued*)

Setting	Function
Notification	Specifies logging options. Right-click the table cell and select <b>Configure</b> to display a dialog box that allows you to configure logging options.  <b>NOTE:</b> Packet capture is not applicable for APE rulebase rules.
Severity	(Optional) Specifies rule severity. Right-click the table cell and select a severity rating to appear in logs that are generated when sessions match the rule.
Install On	Specifies target IDP Series devices for the rule. By default, IDP security policy rules can be applied to any IDP Series device. Right-click the table cell and select <b>Select Target</b> to display a dialog box that allows you to specify the IDP Series devices to which the rule can be installed.
Comments	(Optional) Adds notations about the rule. Right-click the table cell and select <b>Edit Comments</b> to display a dialog box that allows you to make notations about the rule. Comments do not affect the functionality of the security policy rule.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Enabling Per-User Rate Limiting for User-Role-Based Rules on page 321](#)
- [Verifying the APE Rulebase on page 491](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

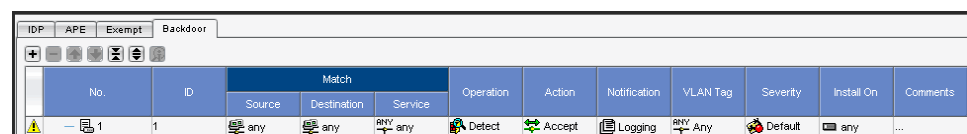
- [Understanding the APE Rulebase on page 69](#)
- [User-Role-Based Policy Feature Overview on page 58](#)
- [Using Application Objects on page 73](#)
- [Using Application Identification on page 43](#)

## Configuring Backdoor Rulebase Rules (NSM Procedure)

The Backdoor rulebase triggers actions when the IDP engine detects the kind of interactive traffic produced during backdoor attacks.

[Figure 84 on page 233](#) shows the Backdoor rulebase in the NSM security policy editor, where you can modify Backdoor rules. [Table 69 on page 234](#) describes the rule settings you can configure.

Figure 84: NSM Security Policy Editor: Backdoor Rulebase



To create Backdoor rulebase rules:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy to which you want to add Backdoor rulebase rules.
3. Add the Backdoor rulebase by clicking the + icon in the upper right region of the policy viewer and selecting **Add Backdoor Rulebase**.
4. Add a rule by clicking the + icon within the rules viewer.
5. Modify rule settings by right-clicking the table cell for the setting and making your selection.
6. Click **OK** to save your changes.

**Table 69: Backdoor Rulebase Rule Settings**

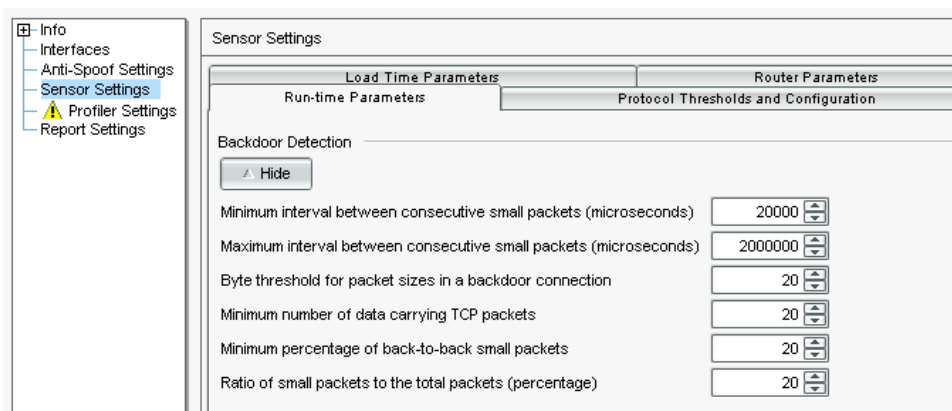
Setting	Function
No.	Adds, deletes, copies, or reorders rules. Right-click the table cell for the rule number and make your selection.
Match	Sets source, destination, and service matches.
Operation	<b>Detect</b> —Enables detection of interactive traffic.
	<b>Ignore</b> —Disables detection of interactive traffic.
Action	<b>Accept</b> —Accepts the interactive traffic.
	<b>Drop Connection</b> —Drops the interactive connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic not prone to spoofing.
	<b>Close Client and Server</b> —Closes the interactive connection and sends an RST packet to both the client and the server. If the IDP Series device is in sniffer mode, it sends an RST packet to both the client and server but does not close the connection.
	<b>Close Client</b> —Closes the interactive connection to the client but not to the server.
	<b>Close Server</b> —Closes the interactive connection to the server but not to the client.
Notification	Sets logging and packet capture options.
VLAN Tag	Sets VLAN tag matches.
Severity	Sets severity ratings.
Install On	Specifies target IDP Series devices for the rule. By default, IDP security policy rules can be applied to any IDP Series device. Right-click the table cell and select <b>Select Target</b> to display a dialog box where you can specify the IDP Series devices to which the rule can be installed.

Table 69: Backdoor Rulebase Rule Settings (*continued*)

Setting	Function
Comments	Adds notations about the rule. This setting is optional. Right-click the table cell and select <b>Edit Comments</b> to display a dialog box where you can make notations about the rule. Comments do not affect the functionality of the security policy rule.

If necessary, you can use the NSM Device Manager to tune the thresholds for backdoor detection. [Figure 20 on page 86](#) shows the backdoor detection settings in the NSM Device Manager configuration editor.

Figure 85: NSM Device Manager: Sensor Settings &gt; Run-Time Parameters



#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Modifying the IDP Series Device Configuration on page 351](#)
- [Developing Security Policies Task Summary on page 195](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

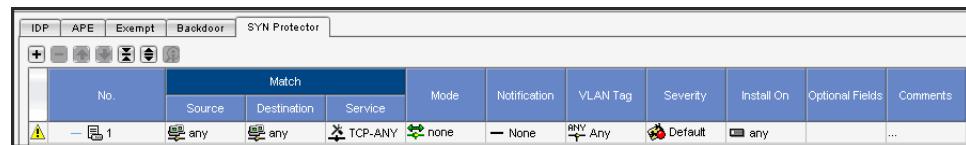
- [Understanding the Backdoor Rulebase on page 85](#)

## Configuring SYN Protector Rulebase Rules (NSM Procedure)

The SYN-Protector rulebase triggers actions when the IDP engine detects traffic that has properties of SYN-flood attacks.

[Figure 86 on page 236](#) shows the SYN Protector rulebase in the NSM security policy editor, where you can modify SYN Protector rules. [Table 70 on page 236](#) describes the rule settings you can configure.

Figure 86: NSM Security Policy Editor: SYN Protector Rulebase



To create SYN Protector rulebase rules:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy to which you want to add SYN Protector rulebase rules.
3. Add the SYN Protector rulebase by clicking the + icon in the upper right region of the policy viewer and selecting **Add SYN Protector Rulebase**.
4. Add a rule by clicking the + icon within the rules viewer.
5. Modify settings by right-clicking the table cell for the setting and making your selection.
6. Click **OK** to save your changes.

Table 70: SYN Protector Rulebase Rule Properties

Setting	Function
No.	Adds, deletes, copies, or reorders rules. Right-click the table cell for the rule number and make your selection.
Match	Sets match criteria for source, destination, and service.  <b>NOTE:</b> We recommend that you do not change the default setting in the Services field: <b>TCP-Any</b> .
Mode	<p><b>None</b>—Turns off the SYN Protector rule.</p> <p><b>Passive</b>—Enables passive mode. In passive mode, the IDP system monitors session startup. If the client does not send an ACK within a timeout period, the IDP engine sends a TCP reset.</p> <p><b>Relay</b>—Enables relay mode. In relay mode, the IDP system performs the three-way handshake with the client host on behalf of the server. Relay mode guarantees that the server allocates resources only to connections that are already in an ESTABLISHED state. The relay is transparent to both the client host and the server.</p> <p><b>NOTE:</b> Relay mode might not work as expected for MPLS traffic. When the IDP engine processes MPLS traffic, it stores the MPLS label information for traffic in each direction. In the case of traffic that matches SYN Protector rules in relay mode, the IDP system is programmed to send a SYN-ACK before the traffic has reached the server. In these cases, the IDP engine does not have server-to-client MPLS label information. Therefore, the SYN-ACK packet does not include an MPLS label. Some MPLS routers can add packets without a label to an existing MPLS tunnel; others drop such packets.</p>
Notification	Sets logging options.  <b>NOTE:</b> Packet capture is not available for SYN Protector rulebase rules.
VLAN Tag	Sets match criteria for VLAN tags.
Severity	Sets severity ratings.

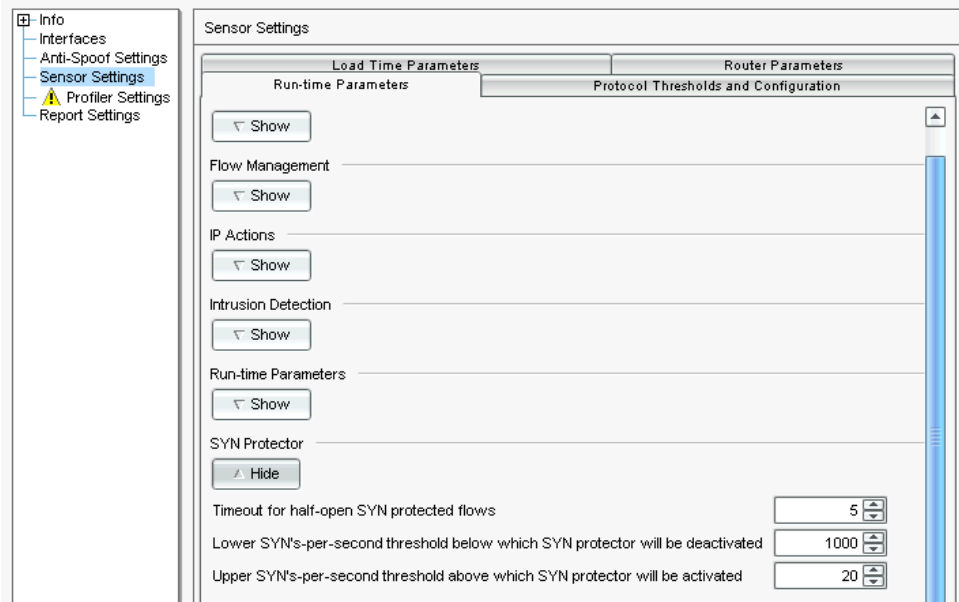


Table 70: SYN Protector Rulebase Rule Properties (*continued*)

Setting	Function
Install On	Specifies target IDP Series devices for the rule. By default, IDP security policy rules can be applied to any IDP Series device. Right-click the table cell and select <b>Select Target</b> to display a dialog box where you can specify the IDP Series devices to which the rule can be installed.
Comments	Adds notations about the rule. This setting is optional. Right-click the table cell and select <b>Edit Comments</b> to display a dialog box where you can make notations about the rule. Comments do not affect the functionality of the security policy rule.

When the SYN Protector rulebase is enabled, the IDP engine detects traffic that exceeds the traffic thresholds you set as runtime parameters. [Figure 21 on page 92](#) shows the SYN protector detection settings in the NSM Device Manager configuration editor.

Figure 87: NSM Device Manager: Sensor Settings &gt; Run-Time Parameters



#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Modifying the IDP Series Device Configuration on page 351](#)
- [Developing Security Policies Task Summary on page 195](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the SYN Protector Rulebase on page 91](#)

## Configuring Traffic Anomalies Rulebase Rules (NSM Procedure)

The Traffic Anomalies rulebase employs a traffic flow analysis method to detect attacks that occur over multiple connections and sessions (such as scans).

Figure 88 on page 238 shows the Traffic Anomalies rulebase in the NSM security policy editor, where you can modify Traffic Anomalies rules. Table 71 on page 238 describes the rule settings you can configure.

**Figure 88: NSM Security Policy Editor: Traffic Anomalies Rulebase**



To create Traffic Anomalies rulebase rules:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy to which you want to add Traffic Anomalies rulebase rules.
3. Add the Traffic Anomalies rulebase by clicking the + icon in the upper right region of the policy viewer and selecting **Add Traffic Anomalies Rulebase**.
4. Add a rule by clicking the + icon within the rules viewer.
5. Modify the setting by right-clicking the table cell for the setting and making your selection.
6. Click **OK** to save your changes.

**Table 71: Traffic Anomalies Rulebase Rule Properties**

Setting	Function
No.	Adds, deletes, copies, or reorders rules. Right-click the table cell for the rule number and make your selection.
Match	Sets match criteria for source, destination, and service.
Traffic Anomalies	<p><b>Ignore</b>—Turns off traffic anomaly detection for traffic that matches the rule.</p> <p><b>Detect</b>—Turns on detection for traffic that matches the rule and displays the View Detect Options dialog box where you can set detection settings.</p> <p>Table 72 on page 239 describes the Traffic Anomalies rulebase detection settings that you can set in the View Detection Options dialog box.</p>
IP Action	Sets IP block, close connection, or notify settings.
Notification	<p>Sets logging settings.</p> <p><b>NOTE:</b> Packet capture is not available for Traffic Anomalies rulebase rules.</p>
VLAN Tag	Sets match criteria for VLAN tags.
Severity	Sets severity ratings.

Table 71: Traffic Anomalies Rulebase Rule Properties (*continued*)

Setting	Function
Install On	Specifies target IDP Series devices for the rule. By default, IDP security policy rules can be applied to any IDP Series device. Right-click the table cell and select <b>Select Target</b> to display a dialog box where you can specify the IDP Series devices to which the rule can be installed.
Comments	Adds notations about the rule. This setting is optional. Right-click the table cell and select <b>Edit Comments</b> to display a dialog box where you can make notations about the rule. Comments do not affect the functionality of the security policy rule.

Table 72 on page 239 describes Traffic Anomalies rulebase detection settings.

Table 72: Traffic Anomalies Rulebase Detection Settings

Setting	Function
TCP scans, UDP Port Scans	<p>Sets a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default port count is 20. The default time threshold is 120 seconds. The rule is matched if the same source scans 20 TCP ports on your internal network within 120 seconds or if the same source scans 20 UDP ports on your internal network within 120 seconds.</p>
Distributed Port Scan	<p>A distributed port scan is an attack that uses multiple source IP addresses to scan ports.</p> <p>Sets a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default IP count is 50. The default time threshold is 120 seconds. The rule is matched if 50 IP addresses attempt to scan ports on your internal network within 120 seconds.</p>
ICMP Sweep	<p>An ICMP sweep is an attack where a single source IP pings multiple IP addresses.</p> <p>Sets a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default IP count is 50. The default time threshold is 120 seconds. The rule is matched if the same source IP attempts to ping 50 IP addresses within 120 seconds.</p>
Network Scan	<p>A network scan is an attack where a single source IP scans multiple IP addresses</p> <p>Sets a port count (number of ports scanned) and the time threshold (the time period that ports are counted) in seconds.</p> <p>The default IP count is 50. The default time threshold is 120 seconds. The rule is matched if the same source IP attempts to scan 50 IP addresses within 120 seconds.</p>
Session Limit	<p>Sets a threshold number of sessions allowed from a single host within a second. The default is 100 sessions.</p> <p>For example, assume your internal network typically has low volume traffic. To detect a sudden increase in traffic from a specific host (which might indicate a worm), configure a rule that matches traffic over your internal network and configure a limit of 200. To block traffic that exceeds the session limit, set the rule IP Action to <b>IP Block</b> and set Blocking Options to <b>Source, Protocol</b>.</p>

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Traffic Anomalies Rulebase on page 97](#)

## Configuring Network Honeypot Rulebase Rules (NSM Procedure)

The Network Honeypot rulebase is a method to detect reconnaissance activities. For background on the Network Honeypot rulebase, see the *IDP Series Concepts and Examples Guide*.

Figure 89 on page 240 shows the Network Honeypot rulebase in the NSM security policy editor, where you can modify Network Honeypot rules. Table 73 on page 240 describes the rule settings you can modify.

**Figure 89: NSM Security Policy Editor: Network Honeypot Rulebase**

No.	Source Addr.	Impersonate	Operation	IP Action	Notification	VLAN Tag	Severity	Install On	Optional Fields	Comments
		Destination A.	Service							
1	any	any	TCP-ANY	Imperso...	None	None	ANY Any	Default	any	...

To create Network Honeypot rulebase rules:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Select the security policy to which you want to add Network Honeypot rulebase rules.
3. Add the Network Honeypot rulebase by clicking the + icon in the upper right region of the policy viewer and selecting **Add Network Honeypot Rulebase**.
4. Add a rule by clicking the + icon within the rules viewer.
5. Modify the property of a rule by right-clicking the table cell for the property and making your selection.
6. Click **OK** to save your changes.

**Table 73: Network Honeypot Rulebase Rule Properties**

Setting	Function
No.	Adds, deletes, copies, or reorders rules. Right-click the table cell for the rule number and make your selection.
Match	Sets match criteria for source, destination, and service.
Source Address	Sets match criteria for source IP addresses or network objects.
Impersonate	Sets match criteria for the destination server and service you want to impersonate.

Table 73: Network Honeypot Rulebase Rule Properties (*continued*)

Setting	Function
Operation	<b>Ignore</b> —Turns off the network honeypot.
	<b>Impersonate</b> —Turns on the network honeypot. The IDP system sends a TCP SYN/ACK in response to TCP requests.
IP Action	Sets IP block, close, or notify actions.
Notification	Sets logging and packet capture settings.
VLAN Tag	Sets match criteria for VLAN tags.
Severity	Sets severity ratings.
Install On	Specifies target IDP Series devices for the rule. By default, IDP security policy rules can be applied to any IDP Series device. Right-click the table cell and select <b>Select Target</b> to display a dialog box where you can specify the IDP Series devices to which the rule can be installed.
Comments	Adds notations about the rule. This setting is optional. Right-click the table cell and select <b>Edit Comments</b> to display a dialog box where you can make notations about the rule. Comments do not affect the functionality of the security policy rule.



**NOTE:** The IDP Series device drops MPLS traffic that matches a Network Honeypot rule. When the IDP engine processes MPLS traffic, it stores the MPLS label information. It stores separate labels for client-to-server and server-to-client communication. In the case of traffic that matches Network Honeypot rules, there is no genuine server-to-client communication, so the IDP engine does not have server-to-client MPLS label information. Therefore, the impersonation operation cannot be supported.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Developing Security Policies Task Summary on page 195](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Network Honeypot Rulebase on page 103](#)



## CHAPTER 26

# Working with Attack Objects

- [Using Attack Objects on page 243](#)
- [Attack Objects Task Summary on page 246](#)
- [Viewing Predefined Attack Objects \(NSM Procedure\) on page 247](#)
- [Working with Attack Groups \(NSM Procedure\) on page 251](#)
- [Creating a Signature Attack Object on page 253](#)
- [Creating a Compound Attack Object on page 273](#)

## Using Attack Objects

---

If the session matches rule settings for source, destination, service, and VLAN tag ID, the IDP engine decodes the traffic and inspects the session packets for the attack objects specified in the rule. The following topics provide guidelines for using attack objects in IDP rulebase rules:

- [Attack Objects Overview on page 243](#)
- [Understanding Predefined Attack Objects and Attack Object Groups on page 244](#)
- [Using Attack Object Groups on page 245](#)
- [Using Custom Attack Objects on page 246](#)

## Attack Objects Overview

When traffic matches an IDP rulebase source/destination/service condition, the IDP engine inspects the traffic for the attack objects you specify.

A *signature attack object* detects known attacks using stateful attack signatures. An attack signature is a pattern that always exists within an attack; if the attack is present, so is the attack signature. With stateful signatures, the IDP engine can look for the specific protocol or service used to perpetrate the attack, the direction and flow of the attack, and the context in which the attack occurs. Stateful signatures produce few false positives because the context of the attack is defined, eliminating huge sections of network traffic in which the attack would not occur.

A *protocol anomaly* identifies unusual activity on the network. It detects abnormal or ambiguous messages within a connection according to the set of rules for the particular protocol being used. Protocol anomaly detection works by finding deviations from protocol standards, most often defined by RFCs and common RFC extensions. Most legitimate

traffic adheres to established protocols. Traffic that does not, produces an anomaly, which may be created by attackers for specific purposes, such as evading an IPS. You cannot create protocol anomaly objects. You can specify a predefined protocol anomaly object as a component of a compound attack object.

A *compound attack object* combines multiple signatures and/or protocol anomalies into a single object. Traffic must match all of the combined signatures and/or protocol anomalies to match the compound attack object; you can specify the order in which signatures or anomalies must match. Use compound attack objects to refine your IDP policy rules, reduce false positives, and increase detection accuracy. A compound attack object enables you to be very specific about the events that need to occur before the IDP engine identifies traffic as an attack. You can use **And**, **Or**, and **Ordered and** operations to define the relationship among different attack objects within a compound attack and the order in which events occur.

Attack object definitions also include data fields to help you group and manage attack objects and use them in security policies. These data fields include category, severity, keywords, and a recommended flag.

*Predefined attack objects* provided by the Juniper Networks Security Center (J-Security Center) team also contain a recommended action for the IDP Series device to take against the attack session.

*Custom attack objects* are ones you create, if your security policy requires more or less protection, or more or less accounting than what the predefined attack objects provide.

Both predefined and custom attack objects are stored in the attack object database.

When you add attack objects to an IDP rulebase rule, you can add attack objects by group or individually.

## Understanding Predefined Attack Objects and Attack Object Groups

The Juniper Networks Security Center (J-Security Center) team has developed more than 600 attack objects and these are included in the attack object database used in IDP security policies.

[Table 21 on page 61](#) describes the attack object groups provided by the J-Security Center.

**Table 74: Predefined Attack Object Groups**

Group	Contents
Attack Type	Contains two subgroups: anomaly and signature. Within each subgroup, attack objects are grouped by severity.
Category	Contains subgroups based on category. Within each category, attack objects are grouped by severity.
Operating System	Contains the following subgroups: BSD, Linux, Solaris, and Windows. Within each operating system, attack objects are grouped by services and severity.



Table 74: Predefined Attack Object Groups (*continued*)

Group	Contents
Severity	Contains the following subgroups: Critical, Major, Minor, Warning, Info. Within each severity, attack objects are grouped by category. Our severity rating is not based on CVSS (Common Vulnerability Scoring System). We do include data from Bugtraq (Symantec) and CVE (Common Vulnerabilities and Exposures).
Web Services	Contains subgroups based on Web services. Within services, attacked objects are grouped by severity.
Miscellaneous	Contains attack objects that have a significant affect on IDP performance.
Response	Contains attack objects where the attack is detected in the server-to-client direction. This group contains a hierarchy of subgroups that includes all of the above categories.

J-Security Center updates the attack object database to provide new attack objects, to revise severities or recommendations, or to remove obsolete attack objects. We recommend you schedule routine, automatic updates.

## Using Attack Object Groups

A *dynamic group* contains members that match properties you specify for the group. You use dynamic groups so that an attack database update automatically populates the group with relevant members. This eliminates the need to review each new signature to determine if you need to use it in your existing security policy. A predefined or custom dynamic group can only contain attack objects and not attack groups. Dynamic group members can be either predefined or custom attack objects.

A *static group* is not automatically updated with new members. It contains only the attack objects or groups you have added. Use static groups when you do not want your attack group dynamically populated during NSM updates. For example, if you customize the action for predefined attack objects to meet your company's security policy guidelines, you can create one or more static groups to contain these attack objects. When you perform an NSM attack object update, your static group will not be affected.

There are two types of static groups: predefined static groups and custom static groups. Predefined static groups are categories of groups provided by default.

A custom static group can include the same members as a predefined static group (predefined attack objects, predefined static groups, and predefined dynamic groups), plus the following members:

- Custom attack objects
- Custom dynamic groups
- Other custom static groups

Static groups require more maintenance than dynamic groups because you must manually add or remove attack objects in a static group to manage the members. However, you can include a dynamic group within a static group to automatically update some attack objects. For example, the predefined attack object group Operating System is a static

group that contains four predefined static groups: BSD, Linux, Solaris, and Windows. The BSD group contains the predefined dynamic group BSD-Services-Critical, to which attack objects can be added during an attack database update.

## Using Custom Attack Objects

The attack objects provided by the Juniper Networks Security Center (J-Security Center) team cover most cases for small business, enterprise, and service provider networks. Your business might encounter cases where you must modify a predefined attack object or create a new one. For example:

- You read a security advisory about a known attack and want to create an attack object that detects the malicious traffic described in that advisory.
- You need to update or improve an existing third-party signature (such as a Snort signature).
- You want to customize an existing signature or protocol anomaly attack object for your local environments. For example, you might need to customize a signature to prevent false positives generated by a specific application running on your network.
- You want to detect specific activity on your network. For example, you might want to detect abnormal traffic (possibly malicious), remote log-ins, or brute force attacks that attempt to guess usernames and passwords.

For a complete tutorial on creating custom attack objects, see the [IDP Series Custom Attack Objects Reference and Examples Guide](#).

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [J-Security Center Updates Overview on page 19](#)
- [Understanding the IDP Rulebase on page 55](#)
- [IDP Rulebase Example: Using Recommended Attack Objects on page 155](#)
- [Exempt Rulebase Example: Exempting an Attack Object on page 176](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)

---

## Attack Objects Task Summary

You use the NSM Object Manager to manage NSM administrative objects, including the attack objects used in IDP security policies.

IDP Series administration can include the following tasks related to attack objects:

- Updating IDP detector engine and the NSM attack database
- Viewing predefined attack object definitions

- Working with attack object groups
- Creating custom attack objects

For details on how to use NSM Object Manager features to copy objects, find references to objects, search for unused objects, or configure object versions, see the NSM online Help.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)
- [Viewing Predefined Attack Objects \(NSM Procedure\) on page 247](#)
- [Working with Attack Groups \(NSM Procedure\) on page 251](#)
- [Creating a Signature Attack Object on page 253](#)
- [Creating a Compound Attack Object on page 273](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)

---

## Viewing Predefined Attack Objects (NSM Procedure)

**Purpose** Juniper Networks Security Center (J-Security Center) develops predefined attack objects and attack object groups for IDP rulebase rules. In most cases, the predefined attack objects are the only attack objects you need to protect your network.

[Figure 90 on page 248](#) shows the attack object viewer in NSM. You can use the attack object viewer to view the following summary for each attack object:

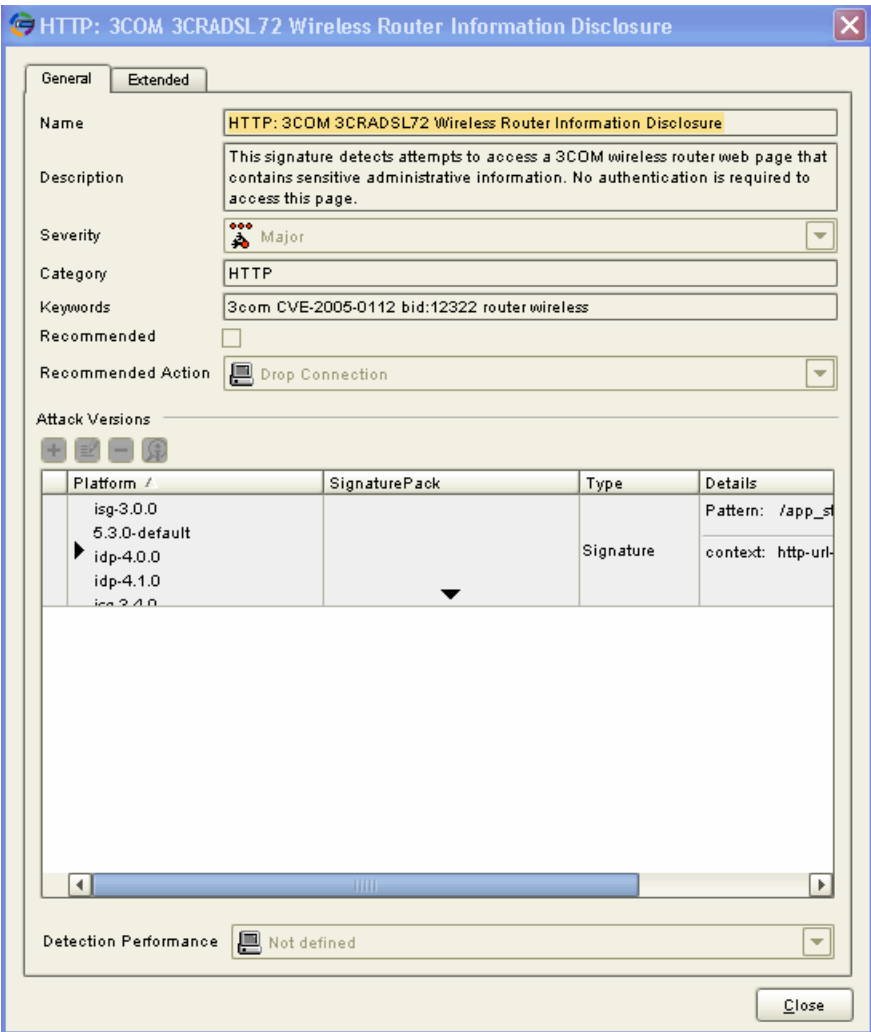
- Name of the attack object
- Severity of the attack: critical, major, minor, warning, info
- Category
- Keywords
- Common Vulnerabilities and Exposures database (CVE) number
- Security Focus Bugtraq database number

Figure 90: NSM Object Manager: Predefined Attack Objects

[illegible]

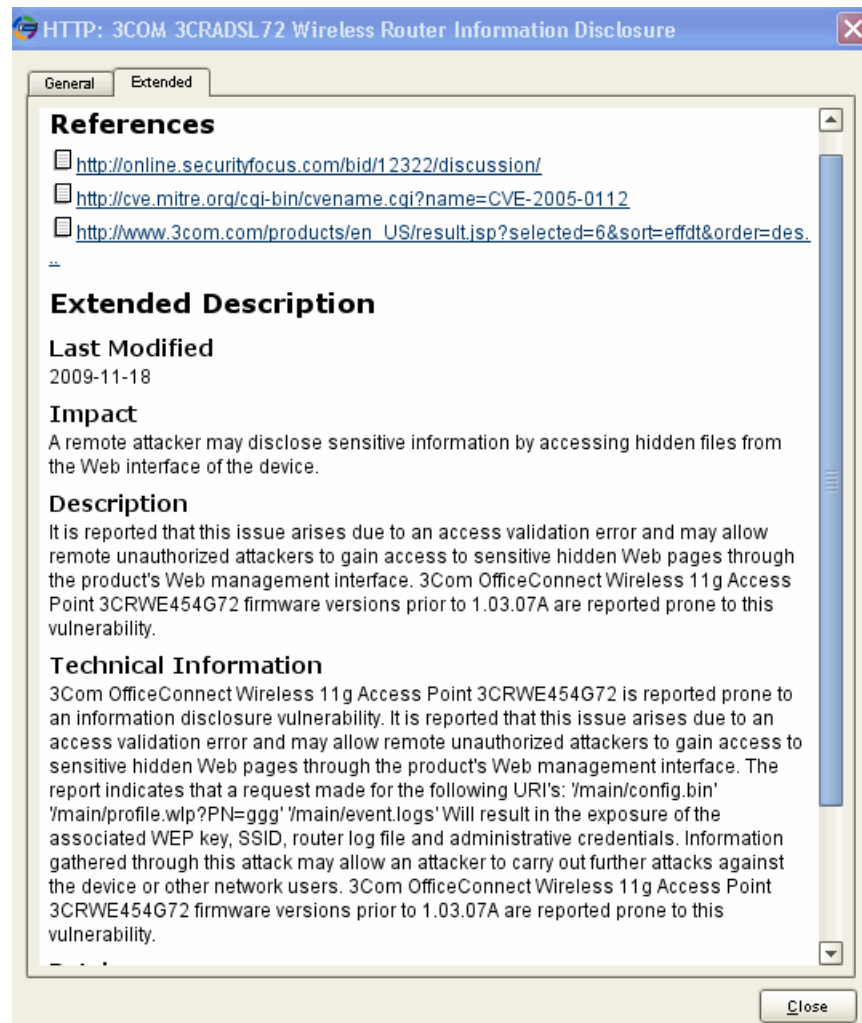
You can double-click the entry in the attack object table to view its details, such as the context and pattern for signature attacks. [Figure 91 on page 249](#) shows attack details for an HTTP attack.

Figure 91: NSM Object Manager Predefined Attack Object Details



Click the Extended tab to see a technical information and security community references that describe the attack. [Figure 92 on page 250](#) shows the information available in the Extended tab.

Figure 92: NSM Object Manager Predefined Attack Object Extended Details



**Action** To view predefined attack object details:

1. In the Object Manager, select **Attack Objects > IDP Objects**.
2. Click either the **Predefined Attacks** or **Predefined Attack Groups** tab to view the predefined attack object list.
3. Double-click the table row entry for the attack object to display its details.



**NOTE:** You cannot create, edit, or delete predefined attack objects.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)

---

## Working with Attack Groups (NSM Procedure)

NSM groups are administrative objects that facilitate configuration and monitoring tasks. You can add attack groups or individual attack objects to IDP rulebase rules and Exempt rulebase rules.

This section includes the following topics:

- [Creating Dynamic Groups on page 251](#)
- [Creating Static Groups on page 252](#)

### Creating Dynamic Groups

A dynamic group contains attack objects that are automatically added or deleted based on specified criteria for the group. The NSM Object Manager includes predefined dynamic groups that work with recommended attack objects, predefined attack objects, the recommended security policy, and predefined policy templates.

When you run an NSM attack database update job, the process automatically performs the following tasks:

- For all new attack objects, compares the predefined attributes of each attack object to each dynamic group criteria and adds the attack objects that match.
- For all updated attack objects, determines if they meet dynamic group criteria and adds or removes them, as appropriate.
- For all deleted attack objects, removes the attack objects from their dynamic groups.

Use of dynamic groups eliminates the need to review each new signature to determine if you need to use it in your existing security policy.

A predefined or custom dynamic group can contain only attack objects and not attack groups. Dynamic group members can be either predefined or custom attack objects.

To create a custom dynamic group:

1. In Object Manager, select **Attack Objects > IDP Objects** to display the IDP Objects dialog box.
2. Click the **Custom Attack Groups** tab, then click the + icon and select **Add Dynamic Group** to display the New Dynamic Group dialog box.
3. Enter a name and description for the static group. Select a color for the group icon.

4. In the Filters tab, click the + icon and add select filters as described in [Table 75 on page 252](#).
5. Click the **Members** tab to view the attack objects now belonging to the group.
6. Click **OK** to save your settings.

**Table 75: Dynamic Attack Group Filters**

Filter	Description
Add Products Filter	Filters attack objects based on the application that is vulnerable to the attack.
Add Severity Filter	Filters attack objects based on attack severity.  <b>NOTE:</b> All predefined attack objects are assigned a severity level by Juniper Networks. However, you can edit this setting to match the needs of your network.
Add Category Filter	Filters attack objects based on category.
Add Last Modified Filter	Filters attack objects based on their last modification date.
Add Recommended Filter	Filters attack objects based on whether they have been marked by J-Security Center as Recommended attack objects.

## Creating Static Groups

A static group contains a specific, finite set of attack objects or groups. There are two types of static groups: predefined static groups and custom static groups.

A custom static group can include the same members as a predefined static group (predefined attack objects, predefined static groups, and predefined dynamic groups), plus the following members:

- Custom attack objects
- Custom dynamic groups
- Other custom static groups

Use custom static groups when you do not want NSM attack object database updates to affect group members.

Static groups require more maintenance than dynamic groups because you must manually add or remove attack objects in a static group to change the members. However, you can include a dynamic group within a static group to automatically update some attack objects. For example, the predefined attack object group Operating System is a static group that contains four predefined static groups: BSD, Linux, Solaris, and Windows. The BSD group contains the predefined dynamic group BSD-Services-Critical, to which attack objects can be added during an attack database update.



To create a custom static group:

1. In Object Manager, select **Attack Objects > IDP Objects** to display the IDP Objects dialog box.
2. Click the **Custom Attack Groups** tab, then click the + icon and select **Add Static Group** to display the New Static Group dialog box.
3. Enter a name and description for the static group.
4. Select a color for the group icon.
5. Select the attack or group from the Attacks/Group list and click **Add**.
6. Click **OK**.

**Related  
Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)

---

## Creating a Signature Attack Object

---

A signature attack object is a pattern you want the system to detect. You use a DFA expression to represent the pattern. All of the other signature properties you can set (such as service or protocol context, direction, and other constraints) are provided so you can optimize performance of the system in detecting the pattern and eliminate false positives. In general, you want to tune settings of a signature attack object so that the system looks for it in every context where it might occur and in no other context.

To configure a signature attack object:

1. In the Object Manager, select **Attack Objects > IDP Objects**.
2. Click the **Custom Attacks** tab.
3. Click the + icon to display the Custom Attack dialog box.
4. Configure attack object settings. [Figure 93 on page 254](#) shows the General tab. [Table 76 on page 254](#) provides guidelines for completing the settings.

Figure 93: Custom Attack Object: General Tab

**New - Custom Attack**

General Extended

Name

Description

Severity Info

Category

Keywords

Recommended ☐

Attack Versions

Platform	SignaturePack	Type	Details
----------	---------------	------	---------

Detection Performance Not defined

OK Cancel

Table 76: Custom Attack Dialog Box: General Tab Settings

Setting	Description
Name	The name displayed in the UI.  <i>TIP:</i> Include the protocol the attack uses as part of the attack name.
Description	(Optional) Information about the attack. Although a description is optional when you create a new attack object, it can help you remember important information about the attack. For examples, view the attack descriptions for predefined attacks.
Severity	Info, Warning, Minor, Major, or Critical. Critical attacks are attempts to crash your server or gain control of your network. Informational attacks are the least dangerous and typically are used by network administrators to discover holes in their own security system.
Category	A predefined or new category.

Table 76: Custom Attack Dialog Box: General Tab Settings (*continued*)

Setting	Description
Keywords	Unique identifiers that can be used to search and sort log records.
Recommended	Indicates that this attack object is among your highest-risk set of attack objects. Later, when you add this attack object to dynamic groups, you can specify whether to include only recommended attack objects.
Attack Versions	Skip this for now.
Detection Performance	Select <b>High</b> , <b>Medium</b> , <b>Low</b> , or <b>Not Defined</b> .

5. Configure additional attack details on the Extended tab. [Figure 94 on page 255](#) shows the Extended tab. [Table 77 on page 256](#) provides guidelines for completing the settings.

Figure 94: Custom Attack Object: Extended Tab

**New - Custom Attack**

General Extended

Specify web sites (using URLs) that provide details about this attack.

Primary URL

Secondary URL

Tertiary URL

Standard References

CVE

BugTraq

Provide more detailed information about this attack.

Impact

Description

Tech Info

Patches

**Hint:** HTML tags can be used to include links etc.

Table 77: Custom Attack Dialog Box: Extended Tab Settings

Setting	Description
Primary URL	Up to three URLs (primary, secondary, tertiary) to external references you used to research the attack.
Secondary URL	
Tertiary URL	
CVE	The Common Vulnerabilities and Exposures (CVE) ID that the attack object addresses. CVE is a standardized list of vulnerabilities and other information security exposures. The CVE number is an alphanumeric code, such as CVE-2209.
BugTraq	The BugTraq ID number that the attack object addresses. BugTraq is a moderated mailing list that discusses and announces computer security vulnerabilities. The BugTraq ID number is a three-digit code, such as 831 or 120.
Impact	Information about the impact of a successful attack, including information about system crashes and access granted to the attacker.
Description	Additional information.
Tech Info	Information about the vulnerability, the commands used to execute the attack, which files are attacked, registry edits, and other low-level information.
Patches	Any patches available from the product vendor, as well as information about how to prevent the attack.

6. Click the **General** tab.
7. Under Attack Versions, click the + icon to display the New Attack wizard.
8. On the Target Platform and Type page, select a device platform and attack type. [Figure 95 on page 257](#) shows the Target Platform and Type page. [Table 78 on page 258](#) describes the attack types.

Figure 95: Custom Attack: Target Platform and Type Page

**Custom Attack -- Target Platform and Type**  
Please select one or more target platforms for this version.

- ☐ mx 9.4 and above
- ☐ j-series 9.5 and above
- ☐ srx-branch 9.4 and above
- ☐ isg-3.0.0
- ☐ isg-3.1.134269
- ☐ isg-3.1.135801
- ☐ isg-3.4.0
- ☐ isg-3.4.125129
- ☐ isg-3.4.135816
- ☐ isg-3.5.0
- ☐ isg-3.5.134268
- ☐ isg-3.5.135816
- ☐ srx 9.2 and above
- ☐ idp-4.0.0
- ☐ idp-4.0.110090709
- ☐ idp-4.0.110090831
- ☐ idp-4.1.0
- ☐ idp-4.1.110100209
- ☐ idp-4.1.110100525
- ☐ idp-4.2.0
- ☐ idp-4.2.110100209
- ☐ idp-4.2.110100525
- ☐ idp-5.0.0
- ☐ idp-5.0.110100209
- ☐ idp-5.0.110100525
- ☒ idp-5.1.110100525
- ☐ 5.3.0-default
- ☐ 5.3.0-ro-bo
- ☐ 5.3.0-smb-server
- ☐ 5.3.0-worm

Type: Signature

Next Cancel Help

Table 78: Attack Object Types

Type	Description
Signature	<p>Uses a stateful attack signature (a pattern that always exists within a specific section of the attack) to detect known attacks.</p> <p>Stateful signature attack objects also include the protocol or service used to perpetrate the attack and the context in which the attack occurs.</p> <p>If you know the exact attack signature, the protocol, and the attack context used for a known attack, select this option.</p>
Compound Attack	<p>Detects attacks that use multiple methods to exploit a vulnerability. This object combines multiple signatures or protocol anomalies into a single attack object, forcing traffic to match all combined signatures or anomalies within the compound attack object before traffic is identified as an attack.</p> <p>By combining and even specifying the order in which signatures or anomalies must match, you can be very specific about the events that must place before the IDP engine identifies traffic as an attack.</p> <p>If you need to detect an attack that uses several benign activities to attack your network, or if you want to enforce a specific sequence of events to occur before the attack is considered malicious, select this option.</p>

9. Select **Signature** and click **Next**.
10. On the Custom Attack – General Properties page, configure constraints and other settings. [Figure 96 on page 259](#) shows the Custom Attack – General Properties page. [Table 79 on page 259](#) provides guidelines for completing the settings.

Figure 96: Custom Attack - General Properties Page

Custom Attack -- General Properties

Specify Signature Information

Info

Platform(s)

idp5.1.0

Type

Signature

False-Positives

Unknown

Service Binding

Protocol Type

Any

Time Binding

Enabled

Scope

Peer

Count/Min.

2

With-in Bytes Constraint

Lower Limit / Upper LimitStart Point

With-in Packets Constraint

Lower Limit / Upper Limit

Context Check

Constraint / Comparison OperatorOperand

Back

Next

Cancel

Help

Table 79: Custom Attack – General Properties

Property	Description
Info	

Table 79: Custom Attack – General Properties (*continued*)

Property	Description
False Positives	<p>Select the frequency that the attack object produces a false positive on your network: <b>Unknown</b>, <b>Rarely</b>, <b>Occasionally</b>, <b>Frequently</b>.</p> <p>Typically, you do not initially know the frequency of false positives. You can update this setting as your observations change.</p>
<b>Service Binding</b>	
Protocol Type	<p><b>Service</b>—If you were able to determine the service through your research, select <b>Service</b>. Later in the wizard, you can specify a service context.</p> <hr/> <p><b>IP</b>—If you are not sure of the service but you know IP details, select <b>IP</b> and specify a protocol type number.</p> <hr/> <p><b>TCP, UDP, or ICMP</b>—If you do not know the service context but you know protocol details, select the protocol.</p> <p>For TCP and UDP protocol types, specify the port ranges.</p> <hr/> <p><b>RPC</b>—If you are detecting threats over remote procedure call (RPC) protocol, select this option and specify the program ID.</p> <p>RPC is used by distributed processing applications to handle interaction between processes remotely. When a client makes a remote procedure call to an RPC server, the server replies with a remote program. Each remote program uses a different program number.</p> <hr/> <p><b>IPv6 or ICMPv6</b>—Do not select these options. IDP Series devices do not support inspection of IPv6.</p> <hr/> <p><b>Any</b>—If you are unsure of the correct service, select <b>Any</b> to match the signature in all services. Matching any service essentially turns off service binding and has a significant performance impact. Specify <b>Any</b> when you know that attacks are using multiple services to attack your network.</p> <p><b>NOTE:</b> You must select a service binding other than <b>Any</b> if you want to select a context for the attack.</p>



Table 79: Custom Attack – General Properties (*continued*)

Time Binding	
Enable	Time binding attributes track how many times a signature is repeated. By configuring the scope and count of an attack, you can detect a sequence of the same attacks over a period of time (one minute) across sessions. This method is useful for detecting brute force attacks that attempt to guess authentication credentials or overwhelm system capacity to handle data.
Scope	<p>Select the scope within which the count occurs:</p> <ul style="list-style-type: none"> <li>• <b>Source</b>—Detects the signature in traffic from the source IP address for the specified number of times, regardless of the destination IP address.</li> <li>• <b>Destination</b>—Detects the signature in traffic from the destination IP address for the specified number of times, regardless of the source IP address.</li> <li>• <b>Peer</b>—Detects the signature in traffic between source and destination IP addresses of the sessions for the specified number of times.</li> </ul>
Count/Min	<p>Enter the number of times per minute that the signature must be detected within the specified scope before the device identifies the traffic as a match.</p> <p>The minute timer starts when the signature first matches the event. If the signature matches the same event for the specific count or higher within the next 60 seconds, the traffic is a match.</p> <p>The system increments the count each time it detects the signature, either regardless of port (application identification) or according to your port binding settings. For example, when the system detects the signature on TCP/80 and then on TCP/8080, the count is 2.</p>

Table 79: Custom Attack – General Properties (*continued*)

Constraints	
Within Bytes Constraint	<p>Use this constraint to require that the pattern be found within a byte range:</p> <ul style="list-style-type: none"> <li>• Lower limit—Specify the beginning of the range.</li> <li>• Upper limit—Specify the end of the range.</li> <li>• Start point—Your selection must be consistent with your pattern context setting. For example, if you configured one of the service contexts, select <b>Context</b>. If you configured one of the packet contexts, select <b>Packet</b>. If you configured one of the stream contexts, select <b>Stream</b>.</li> </ul> <p>In NSM, it is possible to select a start point that is inconsistent with the pattern context setting. For example, the NSM object editor allows you to configure a pattern context http-variable and then set a within bytes start point that is start-of-packet. However, the within bytes match logic will be resolved to the start point you should have selected: context.</p> <p>Inspection for this object terminates when the range limit is reached.</p> <p>Example: If you know a threat can be identified either completely within the first 20 bytes of the http-variable context or not identified at all, you set the context to http-variable and use the within-bytes constraint to terminate inspection after bytes 1-20 of the generated http-variable context are processed.</p> <p>You can set multiple constraints. The constraints are evaluated as a Boolean OR.</p> <p>Example: You configure two start-of-stream constraints with byte ranges of 20-40 and 80-100. The constraint rules out matches unless found within either byte range.</p>
Within Packets Constraint	<p>Use this constraint to require that the pattern be found completely within a packet range:</p> <ul style="list-style-type: none"> <li>• Lower limit—Specify the beginning of the range.</li> <li>• Upper limit—Specify the end of the range.</li> </ul> <p>Inspection (for this object) terminates when the range limit is reached.</p> <p>Example: If you know a threat can be identified either in the first 2 packets or not identified at all, you set a stream context and use the within packets constraint to terminate inspection after 2 packets.</p>
Context Check	<p>Use this constraint to require the matching context be of a specified byte length to be a hit:</p> <ul style="list-style-type: none"> <li>• Constraint—Select <b>length</b>.</li> <li>• Comparison operator—Select =, !, &gt;, or &lt;.</li> <li>• Operand—Select a byte length.</li> </ul> <p>Example: You can use the context check constraint as a tuning device to limit processing for harmless traffic. For example, if you know that a certain class of attack, like a buffer overflow attack, always has an unusually large byte length in a given context, you can use this constraint to ignore contexts of normal length. If you set the FTP username context length requirement to be &gt; 18, you would only see signature hits if the FTP username context is longer than 18 bytes.</p> <p>You can specify multiple constraints. For example, if you add a &lt; 25 constraint to the previous example, you would only see hits if the username context is between 18 and 25 bytes.</p>

Click **Next**.

11. On the Custom Attack – Attack Pattern page, configure pattern settings. [Figure 97 on page 263](#) shows the Custom Attack – Attack Pattern page. [Table 80 on page 263](#) provides guidelines for completing the settings.

Figure 97: Custom Attack – Attack Pattern Page

Custom Attack -- Attack Pattern

Specify Signature Information

Detection

Pattern

Negate

☐

Context

Please select...

Direction

Client to Server

Flow

Both

Back

Next

Cancel

Help

Table 80: Custom Attack – Attack Pattern

Setting	Description
Pattern	A DFA expression. The following rows summarize DFA syntax conventions. For detailed information, consult a standard source on programming with regular expressions.

Table 80: Custom Attack – Attack Pattern (*continued*)

Setting	Description
\B.0.1..00\B	<p>Bit-level matching for binary protocols. The length of the bitmask must be in multiples of 8.</p> <p>The first \B denotes the start of the bitmask. The last \B denotes the end of the bitmask.</p> <p>The decimal (.) indicates the bit can be either 0 or 1.</p> <p>A 0 or 1 indicates the bit at that position must be 0, or must be 1.</p>
\0 <octal_number>	For a direct binary match.
\X<hexadecimal-number>\X	For a direct binary match.
\[<character-set>\]	For case-insensitive matches.
.	To match any symbol.
*	To match 0 or more symbols.
+	To match 1 or more symbols.
?	To match 0 or 1 symbol.
()	Grouping of expressions.
	<p>Alternation. Typically used with ().</p> <p>Example: The following expression matches dog or cat: (dog   cat).</p>
[]	<p>Character class. Any explicit value within the bracket at the position matches.</p> <p>Example: [Dd]ay matches Day and day.</p>
[<start>--<end>]	<p>Character range. Any value within the range (denoted with a hyphen). You can mix character class and a hexadecimal range.</p> <p>Example: [AaBbCcDdEeFf0-9].</p>
[^<start>--<end>]	<p>Negation of character range.</p> <p>Example: [^Dd]ay matches Hay and ray, but not Day or day.</p> <p><b>NOTE:</b> To negate an entire signature pattern, select the Negate option under the pattern text box.</p>
\u<string>\u	Unicode insensitive matches.
\s	Whitespace.

Setting	Description
---------	-------------

Character	Escaped
*	\*
(	\(
)	\)
.	\.
+	\+
\	\\
[	\0133
]	\0135

Table 80: Custom Attack – Attack Pattern (*continued*)

Setting	Description
Context	<p>Binds pattern matching to a context.</p> <p>For known services, such as HTTP, select the service in the first box, and select the HTTP context you discovered with <b>scio ccap</b>, such as HTTP POST Parsed Param, in the second box.</p> <p>If you were unable to discover the context, select <b>Other</b> in the first box, and select one of the following contexts in the second box:</p> <ul style="list-style-type: none"> <li>• <b>Packet</b>—Detects the pattern in any packet.</li> <li>• <b>First Packet</b>—Inspects only the first packet of a stream. When the flow direction is set to any, the detector engine checks the first packet of both the server-to-client (STC) and client-to-server (CTS) flows. Less processing means greater performance. If you know that the pattern appears in the first packet of a session, select <b>First Packet</b>.</li> <li>• <b>First Data Packet</b>—Inspection ends after the first packet of a stream. Select this option to detect the attack in only the first data packet of a stream. If you know that the pattern appears in the first data packet of a stream, select <b>First Data Packet</b>.</li> <li>• <b>Stream 256</b>—Reassembles packets and searches for a pattern match within the first 256 bytes of a traffic stream. Stream 256 is often the best choice for non-UDP attacks. When the flow direction is set to <b>any</b>, the detector engine checks the first 256 bytes of both the STC and CTS flows. If you know that the pattern will appear in the first 256 bytes of a session, select <b>Stream 256</b>.</li> <li>• <b>Stream 8K</b>—Like Stream 256 except reassembles packets and searches for a pattern match within the first 8192 bytes of a traffic stream.</li> <li>• <b>Stream 1K</b>—Like Stream 256 except reassembles packets and searches for a pattern match within the first 1024 bytes of a traffic stream.</li> <li>• <b>Line</b>—Detects a pattern within a specific line. Use this context for line-oriented applications or protocols (such as FTP).</li> <li>• <b>Stream</b>—Reassembles packets and extracts the data to search for a pattern match. However, the IDP engine does not recognize packet boundaries for stream contexts, so data for multiple packets is combined. Select this option only when no other context option contains the attack.</li> </ul> <p><b>NOTE:</b> If you select a line, stream, or service context, you do not configure match criteria for IP settings and protocol header fields.</p>
Direction	<p>Select the direction in which to detect the pattern:</p> <ul style="list-style-type: none"> <li>• <b>Client to Server</b>—Detects the pattern only in client-to-server traffic.</li> <li>• <b>Server to Client</b>—Detects the pattern only in server-to-client traffic.</li> <li>• <b>Any</b>—Detects the pattern in either direction.</li> </ul> <p>The session initiator is considered the client, even if that source IP is a server.</p>
Flow	<p>Select the flow in which to detect the attack:</p> <ul style="list-style-type: none"> <li>• <b>Control</b>—Detects the pattern in the initial connection that is established to issue commands, requests, and so on. Ninety-nine percent of signatures use control.</li> <li>• <b>Auxiliary</b>—Detects the pattern in the response connection that is established intermittently to transfer requested data. This option supports a small number of protocols, such as PTP.</li> <li>• <b>Both</b>—Detects the pattern in the initial and response connections.</li> </ul> <p><b>TIP:</b> Using a single flow (instead of Both) improves performance and increases detection accuracy.</p>

Click **Next** to display the Custom Attack – IP Settings and Header Matches page. [Figure 98 on page 267](#) shows the Custom Attack – IP Settings and Header Matches page. [Table 81 on page 268](#) provides guidelines for completing the settings.

**Figure 98: Custom Attack – IP Settings and Header Matches Page**

**Custom Attack -- IP Settings and Header Matches**  
Specify Signature Information

IP Protocols

IP Version  
☒ IPv4 ☐ IPv6

Type-of-service

Packet length

Id

Time-to-live

Protocol

Source

Destination

RB

MF

DF

Back Finish Cancel Help

12. If you have selected a line, stream, stream 256, or service context, do not configure match criteria for IP settings and protocol header fields. Click **Finish**.

If you are using a packet context, you can refine matching by adding criteria for IP flags and packet headers, as described in the following tables.



**TIP:** If you are unsure of the IP flags and IP fields you want to match, leave all fields blank. If no values are set, the IDP engine attempts to match the signature for all header contents.

**Table 81: Custom Attack – IP Settings and Header Matches Page**

Setting	Description
IP Version	Select <b>IPv4</b> . IDP Series devices do not support inspection of IPv6.
Type of Service	Service type. Common service types are: <ul style="list-style-type: none"> <li>• 0000 Default</li> <li>• 0001 Minimize Cost</li> <li>• 0002 Maximize Reliability</li> <li>• 0003 Maximize Throughput</li> <li>• 0004 Minimize Delay</li> <li>• 0005 Maximize Security</li> </ul>
Packet Length	Number of bytes in the packet, including all header fields and the data payload.
ID	Unique value used by the destination system to reassemble a fragmented packet.
Time-to-live	Time-to-live (TTL) value of the packet. This value represents the number of routers the packet can pass through. Each router that processes the packet decrements the TTL by 1; when the TTL reaches 0, the packet is discarded.
Protocol	Protocol used in the attack.
Source	IP address of the attacking device.
Destination	P address of the attack target.
RB	Reserved bit. This bit is not used.
MF	More fragments. When set (1), this option indicates that the packet contains more fragments. When unset (0), it indicates that no more fragments remain.
DF	Don't fragment. When set (1), this option indicates that the packet cannot be fragmented for transmission.

Figure 99 on page 269 shows the Custom Attack – IP Settings and Header Matches page. Table 82 on page 269 provides guidelines for completing the settings.



Figure 99: Custom Attack Object: TCP Packet Header Fields

Custom Attack -- IP Settings and Header Matches

Specify Signature Information

IP

Protocols

TCP/UDP/ICMP Header Matches

TCP Packet Header Fields

Source Port

none

none

Dest Port

none

none

Seq. Number

none

none

Ack. Number

none

none

Header Length

none

none

Data Length

none

none

Window Size

none

none

UrgPtr

none

none

Urgent bit

none

ACK bit

none

PSH bit

none

RST bit

none

SYN bit

none

FIN bit

none

R1 bit

none

R2 bit

none

Back

Finish

Cancel

Help

Table 82: Custom Attack Object: TCP Packet Header Fields

Setting	Description
Source Port	Port number on the attacking device.
Destination Port	Port number of the attack target.
Sequence Number	Sequence number of the packet. This number identifies the location of the data in relation to the entire data sequence.

Table 82: Custom Attack Object: TCP Packet Header Fields (*continued*)

Setting	Description
ACK Number	ACK number of the packet. This number identifies the next sequence number; the ACK flag must be set to activate this field.
Header Length	Number of bytes in the TCP header.
Window Size	Number of bytes in the TCP window size.
Data Length	Number of bytes in the data payload. For SYN, ACK, and FIN packets, this field should be empty.
Urgent Pointer	Data in the packet is urgent; the URG flag must be set to activate this field.
URG Bit	When set, the urgent flag indicates that the packet data is urgent.
ACK Bit	Acknowledgment flag. When set, acknowledges receipt of a packet.
PSH Bit	Push flag. When set, indicates that the receiver should push all data in the current sequence to the destination application (identified by the port number) without waiting for the remaining packets in the sequence.
RST Bit	Reset flag. When set, resets the TCP connection, discarding all packets in an existing sequence.
FIN Bit	Final flag. When set, indicates that the packet transfer is complete and the connection can be closed.
R1 Bit, R2 Bit	Reserved bit. Unused.

[Figure 100 on page 271](#) shows the Custom Attack – IP Settings and Header Matches page. [Table 83 on page 271](#) provides guidelines for completing the settings.

Figure 100: Custom Attack Object: UDP Packet Header Fields

Custom Attack -- IP Settings and Header Matches

Specify Signature Information

IP

Protocols

TCP/UDP/ICMP Header Matches

UDP Packet Header Fields

Source Port

none

none

Dest. Port

none

none

Data Length

none

none

Back

Finish

Cancel

Help

Table 83: Custom Attack Object: UDP Header Fields

Setting	Description
Source Port	Port number on the attacking device.
Destination Port	Port number of the attack target.
Data Length	Number of bytes in the data payload.

Figure 101 on page 272 shows the Custom Attack -- IP Settings and Header Matches page. Table 84 on page 272 provides guidelines for completing the settings.

Figure 101: Custom Attack Object: ICMP Packet Header Fields

**Custom Attack -- IP Settings and Header Matches**  
Specify Signature Information

IP Protocols

TCP/UDP/ICMP Header Matches ICMP Packet Header Fields

Type none none

Code none none

Seq. Number none none

Id none none

Data Length none none

Back Finish Cancel Help

Table 84: Custom Attack Object: ICMP Packet Header Fields

Setting	Description
<b>ICMP</b>	
ICMP Type	Primary code that identifies the function of the request or reply.

Table 84: Custom Attack Object: ICMP Packet Header Fields (*continued*)

Setting	Description
ICMP Code	Secondary code that identifies the function of the request or reply within a given type.
Sequence Number	Sequence number of the packet. This number identifies the location of the request/reply in relation to the entire sequence.
ICMP ID	Identification number, which is a unique value used by the destination system to associate requests and replies.
Data length	Number of bytes in the data payload.



**NOTE:** ICMPv6 header fields are not applicable. IDP Series devices do not support inspection of IPv6.

13. Click **Finish**.

#### Related Documentation

The following related topics are included in the *IDP Series Custom Attack Object Reference and Examples Guide*:

- [Creating a Compound Attack Object on page 273](#)
- [Creating Custom Attack Objects Overview](#)
- [Reference: Custom Attack Object Protocol Numbers](#)
- [Reference: Custom Attack Object Service Properties](#)
- [Reference: Custom Attack Object Service Contexts](#)
- [Example: Custom Attack Object DFA Expressions](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)

## Creating a Compound Attack Object

Use compound attack objects in cases where:

- Attacks use multiple methods to exploit a vulnerability and, inspected independently, the individual contexts appear benign.
- Matching multiple contexts reduces false positives.
- Coupling a signature with a protocol anomaly reduces false positives.

You select signature attack objects or predefined anomalies as “members” of the compound object, and you use Boolean expressions to specify matching logic.

To configure a compound attack object:

1. Configure general attack object properties and reference information as described for signature attack objects.

On the Target Platform and Type page, select a target platform, select **Compound Attack**, and click **Next**.

2. On the Custom Attack – General Properties page, configure the settings described in [Table 85 on page 274](#).

**Table 85: Custom Attack – General Properties**

Property	Description
False Positives	Same guidelines as for signature attack objects.
Service Binding	
Time Binding	

---

Click **Next**.

3. On the Compound Members page, specify compound attack parameters and add members. [Figure 102 on page 275](#) shows the Custom Attack – Compound Members page. [Table 86 on page 275](#) provides guidelines for completing the settings.

Figure 102: Custom Attack – Compound Members

**Custom Attack -- Compound Members**  
Specify Signature Information

Compound Attack Parameters

Scope: Session Reset

Boolean Expression:

Member Name	Member Type	Compound Attack Member

Context-Check Constraint

Constraint /	Comparis...	Operand

Match within same context: (empty)

With-in Bytes Constraint

Lower Limit /	Upper Limit	Member N...

With-in Packet Constraint

Lower Limit /	Upper Limit	Member N...

Back Finish Cancel Help

Table 86: Compound Attack Parameters

Setting	Description
Scope	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li><b>Session</b>—Allows multiple matches for the object within the same session.</li> <li><b>Transaction</b>—Matches the object across multiple transactions that occur within the same session.</li> </ul>
Reset	Enable to detect multiple occurrences of the attack object in the same session. Disable to log multiple occurrences as one.

Table 86: Compound Attack Parameters (*continued*)

Setting	Description
Boolean Expression	<p>Type a Boolean expression using the following Boolean operators:</p> <ul style="list-style-type: none"> <li>• OR—If either of the member name patterns match, the expression matches.</li> <li>• AND—If both of the member name patterns match, the expression matches. It does not matter which order the members appear in.</li> <li>• OAND—If both member name patterns match, and if they appear in the same order as in the Boolean expression, the expression matches.</li> </ul> <p>For example, the Boolean expression ((s1 OAND s2) OR (s1 OAND s3)) AND (s4 AND s5) would match an attack that contains s1 followed by either s2 or s3, and that also contains s4 and s5 in any location.</p>
Add member	<p>Click the + icon, select <b>Signature</b> or <b>Protocol Anomaly</b>, and complete the configuration details.</p> <p>For signature members, specify the same contextual information as you do for a signature attack object.</p> <p>For protocol anomaly members, select from a list of predefined protocol anomalies.</p> <p><b>BEST PRACTICE:</b> Our signature team uses the following naming convention for members: m01, m02, m03, and so on. We recommend you use this same naming convention.</p>
Context Check	<p>Use this constraint to require the matching context be of a specified byte length to be a hit:</p> <ul style="list-style-type: none"> <li>• Constraint—Select <b>length</b>.</li> <li>• Comparison operator—Select =, !, &gt;, or &lt;.</li> <li>• Operand—Select a byte length.</li> </ul> <p>Example: You can use the context check constraint as a tuning device to limit processing for harmless traffic. For example, if you know that a certain class of attack, like a buffer overflow attack, always has an unusually large byte length in a given context, you can use this constraint to ignore contexts of normal length. If you set the FTP username context length requirement to be &gt; 18, you see signature hits only when the FTP username context is longer than 18 bytes.</p> <p>You can specify multiple constraints. For example, if you add a &lt; 25 constraint to the previous example, you see hits only when the username context is between 18 and 25 bytes.</p>
Match within same context	<p>Use this constraint to require selected signature members to be found in the same context instance (in any order). You can select up to 32 signature members.</p> <p>Protocol anomaly members are not selectable and are not a component of this constraint.</p> <p>Example: You design a compound attack with service context ftp-filename, and you enable this restraint. The pattern for member 1 is <b>test</b>; the pattern for member 2 is <b>hello</b>. A user opens an FTP session and requests files test.txt and hello.txt. Each file transfer occurs in its own context, not within the same context instance, so the FTP session does not trigger this attack object. Instead, consider what happens when the user requests a file named test-hello.txt. In this case, both members are found in a single context instance, so the FTP session is a match.</p>
Within Bytes Constraint	<p><b>NOTE:</b> IDP OS Release 5.1 does not support the within bytes constraint for compound attack objects.</p>



Table 86: Compound Attack Parameters (continued)

Setting	Description
Within Packets Constraint	<p>Use this constraint to require that the pattern be found completely within a packet range of a stream:</p> <ul style="list-style-type: none"><li>• Lower limit—Specify the beginning of the range.</li><li>• Upper limit—Specify the end of the range.</li><li>• Member—Select one or two members. You cannot configure a relationship for more than two members.</li></ul> <p>If you set a packet constraint for one member, the program logic counts packets beginning with the start-of-stream. The member must be found completely within the packet range indicated.</p> <p>If you select two members and apply a packet constraint to them, the program logic counts the first match as packet 0. If you specify a range of 1-2 with member 1 and member 2, the second pattern must occur within one or two packets from the packet containing the first match.</p> <p>Specifying 0-1 requires the pattern to appear in the same packet or within one packet from the first match. Order does not matter unless you use an Boolean ordered AND to specify the order in which the patterns must appear.</p> <p>Inspection for this object terminates when the range limit has been reached.</p>

4. Click **Finish**.

**Related Documentation**

The following related topics are included in the *IDP Series Custom Attack Object Reference and Examples Guide*:

- [Creating a Signature Attack Object on page 253](#)
- [Creating Custom Attack Objects Overview](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)



## CHAPTER 27

# Working with Application Objects

- [Using Application Objects on page 279](#)
- [Application Objects Task Summary on page 286](#)
- [Viewing Predefined Application Objects \(NSM Procedure\) on page 287](#)
- [Viewing Predefined Extended Application Objects \(NSM Procedure\) on page 290](#)
- [Creating Application Groups \(NSM Procedure\) on page 292](#)
- [Creating a Custom Application \(NSM Procedure\) on page 292](#)

## Using Application Objects

---

You specify application objects in APE rules as a key element in the matching tuple. This topic provides an overview of application objects and includes the following sections:

- [Application Objects Overview on page 279](#)
- [Understanding Predefined Application Objects on page 280](#)
- [Using Application Groups on page 284](#)
- [Using Custom Application Objects on page 285](#)

## Application Objects Overview

Application objects are also called *application signatures*. The signature comprises the Layer 7 protocol, protocol contexts, and a DFA pattern found in client-to-server and server-to-client traffic flows. An application object adds program logic to signatures, such as the capability of chaining signatures to create an ordered or unordered compound expression, a maximum number of transactions wherein the signature must occur to be a match, an order value that sets match precedence in cases where multiple signatures are identified, and a unique ID that the system uses both for logical processing and reporting. Extended application objects, also called nested applications, identify Web 2.0 applications running over HTTP.

Application objects are stored in the NSM database application signature table (also referred to as the appsig table) and extended application signature table (also referred to as the extappsig table). Juniper Networks Security Center (J-Security Center) makes predefined application objects and predefined extended application objects available for download to NSM during signature database updates. You use the NSM Object Manager to manage application objects. You specify application objects in APE rules as

a key element in the matching tuple. You push the application signatures from NSM to your devices when you push policy updates.

## Understanding Predefined Application Objects

J-Security Center makes predefined application objects and predefined extended application objects available for download to NSM during signature database updates. A complete list of application objects is maintained on the J-Security Center [website](#). We recommend that you become familiar with these objects and leverage them in your APE rules as much as possible.

Figure 12 on page 74 shows the NSM Object Manager Predefined Application Objects tab. This view displays the following properties for predefined application objects:

- Name—A unique, descriptive name.
- Application category—A classification used for sorting the list. Not unique.
- Port range—A range of ports on which the application might run. The application is identified only if the server port is within the specified range.
- Application type—A unique identifier used by the application identification feature.
- Port binding—The standard ports known to be used by the application.
- Match order—In case traffic matches protocol, port, and pattern for two or more applications, the match order determines which object is considered the match (the object with the lower match order number is considered the match).

Figure 103: NSM Object Manager: Predefined Application Objects

Application Objects					
Predefined Application Objects		Custom Application Objects	Predefined Extended Application Objects		Application Group Objects
Name	Application Category	Port Ranges	Application Type	Port Binding	Match Order
HPOVTRACE	ENTERPRISE-INFRASTRUCTURE	TCP:5051-5053	HPOVTRACE	TCP:5051-5053	146
HSS-SSL-TCP	MSC	TCP:0-65535	HSS-SSL-TCP	...	32
HSS-SSL-UDP	MSC	UDP:0-65535	HSS-SSL-UDP	...	150
HTTP	WEB	TCP:0-65535	HTTP	TCP:80,3128,80...	99
ICA-TCP	REMOTE-ACCESS	TCP:0-65535	ICA-TCP	TCP:1494	5
ICA-UDP	REMOTE-ACCESS	UDP:0-65535	ICA	UDP:1804	51
ICCP	SCADA	TCP:102	ICCP	TCP:102	147
ICQ	PEER-TO-PEER-CHAT	TCP:0-65535	ICQ	...	22
IDENT	MSC	TCP:113	IDENT	TCP:113	68
IEC104	SCADA	TCP:2404	IEC104	TCP:2404	155
IMAP	MESSAGING	TCP:0-65535	IMAP	TCP:143	115
IPSEC-IKE-MAIN-AO...	ENCRYPTION	UDP:0-65535	IKE	UDP:500	25
IRC	PEER-TO-PEER-CHAT	TCP:0-65535	IRC	TCP:6666,6667,...	46
JABBER	PEER-TO-PEER-CHAT	TCP:0-65535	JABBER	TCP:5222	114
JAVA-RMI	REMOTE-COMMAND	TCP:1099	JAVA-RMI	TCP:1099	188
JONDO-PROXY	WEB	TCP:0-65535	JONDO-PROXY	...	140
KADEMLIA-KAD	PEER-TO-PEER-FILE-SHARING	UDP:0-65535	KADEMLIA-KAD	...	83
KADEMLIA-OVERNET	PEER-TO-PEER-FILE-SHARING	TCP:0-65535 UDP:0-65535	KADEMLIA-OVERNET	...	79
KAZAA	PEER-TO-PEER-FILE-SHARING	TCP:0-65535 UDP:0-65535	KAZAA	...	119
KRB4	ENCRYPTION	UDP:0-65535	KRB4	...	113
KRB5	ENCRYPTION	TCP:0-65535 UDP:0-65535	KRB5	TCP:543 UDP:88	27
KUGOO	PEER-TO-PEER-FILE-SHARING	UDP:0-65535	KUGOO	UDP:7000	78
LDAP	ENTERPRISE-INFRASTRUCTURE	TCP:0-65535	LDAP	TCP:389	111
LOTUSNOTES	MESSAGING	TCP:0-65535	LOTUS-NOTES	TCP:1352	134

You can double-click the table entry to view additional details, including the signature pattern regular expression to match in client-to-server and server-to-client directions.

Figure 13 on page 75 shows the general properties of the predefined application object for HTTP.

Figure 104: NSM Object Manager: Predefined Application: General Tab

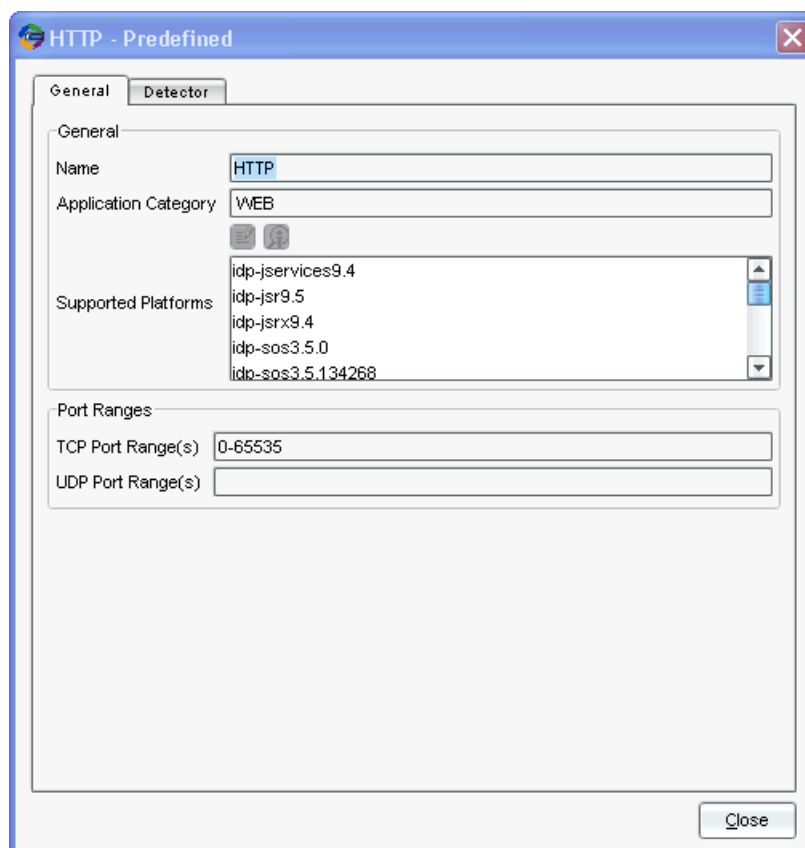


Figure 14 on page 76 shows the signature properties of the predefined application object for HTTP.

Figure 105: NSM Object Manager: Predefined Application: Detector Tab

**HTTP - Predefined**

**General** **Detector**

**Port Binding**

Application Type: HTTP

TCP Port Binding: 80,3128,8000,8080

UDP Port Binding:

**Signature**

**Client-to-server**

DFA Pattern: (\\OPTIONS|HEAD|GET|POST|PUT|B?DELETE|TRACE|SEARCH|B?PROPFIND|PROPPATCH|MKCOL|B?COPY|B?MOVE|LOCK|UNLOCK|CHECKOUT|

PCRE Pattern:

**Server-to-client**

DFA Pattern: (. \*HTTP/1\\.([01])s\\.\\.?.?w<\\DOCTYPE\\w\\.\\.?.?w<\\HTML\\w\\.\\.?.?w<\\?xml\\w\\.\\.?.?w<\\Content-type\\: .). \*

PCRE Pattern:

Minimum data length: 20

Signature Match Order: 122

Close

You can use extended application objects in APE rules if you want to treat various Web 2.0 applications running over HTTP differently. [Figure 15 on page 77](#) shows the NSM Object Manager Predefined Extended Application Objects tab. This view displays the following properties for predefined extended application objects:

- Name—A unique, descriptive name.
- Application category—A classification used for sorting the list. Not unique.
- Extended application ID—A unique identifier. The system uses the unique ID for both logical processing and reporting.
- Application type—A unique identifier used by the application identification feature.
- L7 protocol—Only HTTP is supported.
- Chain order—Indicates whether or not the member signatures are ordered.

Figure 106: NSM Object Manager: Predefined Extended Application Objects

**Application Objects**

Predefined Application Objects   Custom Application Objects   **Predefined Extended Application Objects**   Application Group Objects

Name	Application Category /	Ext ID	Application Type	L7 Protocol	Chain Ord
MYSPACE	SOCIAL-NETWORKING	316	MYSPACE	HTTP	No
MYSPACE-CHAT	SOCIAL-NETWORKING	317	MYSPACE-CHAT	HTTP	No
MYSPACE-MAIL	SOCIAL-NETWORKING	321	MYSPACE-MAIL	HTTP	No
MYSPACE-VIDEO	SOCIAL-NETWORKING	322	MYSPACE-VIDEO	HTTP	No
BEBO	SOCIAL-NETWORKING	329	BEBO	HTTP	No
CLASSMATES	SOCIAL-NETWORKING	290	CLASSMATES	HTTP	No
DOOF	SOCIAL-NETWORKING	343	DOOF	HTTP	No
BLOGGER-POST	SOCIAL-NETWORKING	351	BLOGGER-POST	HTTP	No
MYSPACE-CHAT	SOCIAL-NETWORKING	352	MYSPACE-CHAT	HTTP	No
MYSPACE-VIDEO	SOCIAL-NETWORKING	360	MYSPACE-VIDEO	HTTP	No
VKONTAKTE	SOCIAL-NETWORKING	501	VKONTAKTE	HTTP	No
MIXI	SOCIAL-NETWORKING	444	MIXI	HTTP	No
TIANYA	SOCIAL-NETWORKING	445	TIANYA	HTTP	No
KAIXIN001	SOCIAL-NETWORKING	447	KAIXIN001	HTTP	No
ODNOKLASSNIKI	SOCIAL-NETWORKING	448	ODNOKLASSNIKI	HTTP	No
RENREN	SOCIAL-NETWORKING	449	RENREN	HTTP	No
ADULTFRIENDFINDER	SOCIAL-NETWORKING	480	ADULTFRIENDFINDER	HTTP	No
TARINGA	SOCIAL-NETWORKING	481	TARINGA	HTTP	No
BADDOO	SOCIAL-NETWORKING	483	BADDOO	HTTP	No
NING	SOCIAL-NETWORKING	484	NING	HTTP	No
NETLOG	SOCIAL-NETWORKING	492	NETLOG	HTTP	No
HYVESDOTNL	SOCIAL-NETWORKING	506	HYVESDOTNL	HTTP	No
PLENTYOFFISH	SOCIAL-NETWORKING	508	PLENTYOFFISH	HTTP	No
NATEON	SOCIAL-NETWORKING	403	NATEON	HTTP	No
BLOGSPOT-POST	SOCIAL-NETWORKING	413	BLOGSPOT-POST	HTTP	No
PING-FM	SOCIAL-NETWORKING	509	PING-FM	HTTP	No

You can double-click the table entry to view additional details, including the matching HTTP context, signature pattern, and client-to-server or server-to-client direction. [Figure 16 on page 78](#) shows the properties of the HTTP:Facebook-Access application object.

Figure 107: NSM Object Manager: Extended Application Details

The screenshot shows the 'FACEBOOK-ACCESS - Predefined' dialog box with the 'General' tab selected. The fields are as follows:

- Name: FACEBOOK-ACCESS
- L7 Protocol: HTTP
- Chain Order: No
- Application type: FACEBOOK-ACCESS
- Maximum Transactions: none
- Signature Match Order: 33323

Below the fields is a 'Members' section with a table:

Member /	Context	pattern	direction
m01	http-header-host	(.*)?(facebook\.com fbcdn\.net)	CTS

At the bottom right is a 'Close' button.

An application signature can include one member or more members in a compound signature. Double-click the table row entry for the member to display its details. [Figure 17 on page 78](#) shows the properties of the HTTP:Facebook-Access application object.

Figure 108: NSM Object Manager: Extended Application Member Details

The screenshot shows the 'Signature' dialog box with the following fields:

- Member: m01
- Context: http-header-host
- pattern: (.\*)?(facebook\.com|fbcdn\.net)
- direction: CTS

At the bottom right is a 'Close' button.

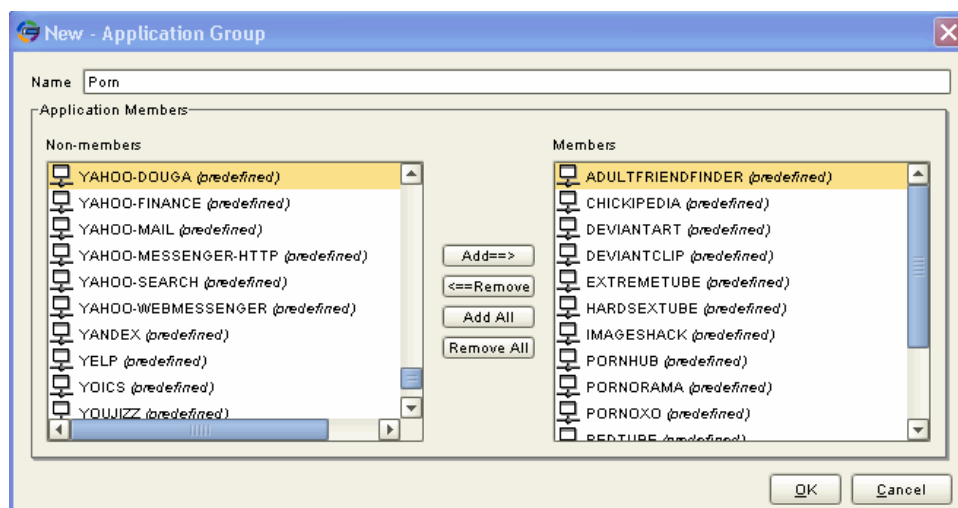
## Using Application Groups

Application groups are administrative objects you can use to simplify rule configuration. A group comprises application objects that you want to treat the same—that is, you want to apply the same action to matching traffic. [Figure 18 on page 79](#) shows the Application Group dialog box. In the Non-Members box, applications are listed first, followed by extended applications. You can nest a group within another group. In the Application



Group dialog box, the icons next to group object names and application object names differ so you can distinguish the two when you browse the lists.

Figure 109: NSM Object Manager: Application Group Dialog Box



## Using Custom Application Objects

You can create rules for most business cases with the predefined application objects provided by J-Security Center. In some cases, you might need to manage traffic for applications not yet supported by J-Security Center. First, check with your Juniper Networks representative to see if a predefined application object is forthcoming. If support for your application object is not forthcoming, you can use the NSM Object Manager to define a custom application object. You can then specify that object as a match for APE rules.

[Figure 19 on page 80](#) shows the Custom Application dialog box.

Figure 110: NSM Object Manager: Custom Application Dialog Box

The screenshot shows the 'Aspera FASP - Custom' dialog box. It has a title bar with the Aspera logo and the text 'Aspera FASP - Custom'. There are two tabs: 'General' and 'Detector'. The 'General' tab is selected. Inside the 'General' tab, there is a 'General' section with three fields: 'Name' (Aspera FASP), 'Application Category' (File-Server), and 'Supported Platforms' (idp5.1.0). Below this is a 'Port Ranges' section with two fields: 'TCP Port Range(s)' (22,33001) and 'UDP Port Range(s)' (0-65535). At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Apply'.

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [J-Security Center Updates Overview on page 19](#)
- [Understanding the APE Rulebase on page 69](#)
- [APE Rulebase Example: Using Extended Application Objects on page 165](#)
- [APE Rulebase Example: Matching Custom Application Objects on page 171](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

## Application Objects Task Summary

Application objects are stored in the NSM database application signature table (also referred to as the appsig table) and extended application signature table (also referred to as the extappsig table). Juniper Networks Security Center (J-Security Center) develops predefined application signatures and makes them available for download to NSM during updates to the signature database. You use the NSM Object Manager to manage application objects. When you push security policy updates, the application signatures

are pushed to the IDP Series device. During traffic inspection, the application identification feature matches traffic to the application objects that are specified in APE rules.

IDP Series administration can include the following tasks related to application objects:

- Updating the application signature table
- Viewing predefined application object definitions
- Creating custom application objects
- Working with application object groups

For information about how to use NSM Object Manager search features, see the NSM online Help.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)
- [Viewing Predefined Application Objects \(NSM Procedure\) on page 287](#)
- [Viewing Predefined Extended Application Objects \(NSM Procedure\) on page 290](#)
- [Creating a Custom Application \(NSM Procedure\) on page 292](#)
- [Creating Application Groups \(NSM Procedure\) on page 292](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)

---

## Viewing Predefined Application Objects (NSM Procedure)

**Purpose** In most cases, the predefined application objects and predefined extended application objects developed by the Juniper Networks Security Center (J-Security Center) are the only ones you need to create APE rules that meet your business objectives. J-Security Center maintains a list of predefined application objects on its [website](#). We recommend that you become familiar with the predefined objects and leverage them in your APE rules as much as possible.

You can also use the NSM Object Manager to view a sortable table of predefined application objects.

[Figure 111 on page 288](#) shows the NSM Object Manager Predefined Application Objects tab.

Figure 111: NSM Object Manager: Predefined Application Objects

Application Objects					
Predefined Application Objects		Custom Application Objects	Predefined Extended Application Objects	Application Group Objects	
Name	Application Category	Port Ranges	Application Type	Port Binding	Match Order
HPOVTRACE	ENTERPRISE-INFRASTRUCTURE	TCP-5051-5053	HPOVTRACE	TCP-5051-5053	146
HSS-SSL-TCP	MSC	TCP-0-65535	HSS-SSL-TCP	...	32
HSS-SSL-UDP	MSC	UDP-0-65535	HSS-SSL-UDP	...	150
HTTP	WEB	TCP-0-65535	HTTP	TCP-80,3128,80...99	99
ICA-TCP	REMOTE-ACCESS	TCP-0-65535	ICA-TCP	TCP-1494	5
ICA-UDP	REMOTE-ACCESS	UDP-0-65535	ICA	UDP-1604	51
ICCP	SCADA	TCP-102	ICCP	TCP-102	147
ICQ	PEER-TO-PEER-CHAT	TCP-0-65535	ICQ	...	22
IDENT	MSC	TCP-113	IDENT	TCP-113	88
IEC104	SCADA	TCP-2404	IEC104	TCP-2404	155
IMAP	MESSAGING	TCP-0-65535	IMAP	TCP-143	115
IPSEC-KE-MAN-AGOR...	ENCRYPTION	UDP-0-65535	IK	UDP-500	25
IRC	PEER-TO-PEER-CHAT	TCP-0-65535	IRC	TCP-6666,6667,...	46
JABBER	PEER-TO-PEER-CHAT	TCP-0-65535	JABBER	TCP-5222	114
JAVA-RMI	REMOTE-COMMAND	TCP-1099	JAVA-RMI	TCP-1099	188
JONDO-PROXY	WEB	TCP-0-65535	JONDO-PROXY	...	140
KADEMLIA-KAD	PEER-TO-PEER-FILE-SHARING	UDP-0-65535	KADEMLIA-KAD	...	83
KADEMLIA-OVERNET	PEER-TO-PEER-FILE-SHARING	TCP-0-65535 UDP-0-65535	KADEMLIA-OVERNET	...	79
KAZAA	PEER-TO-PEER-FILE-SHARING	TCP-0-65535 UDP-0-65535	KAZAA	...	119
KRB4	ENCRYPTION	UDP-0-65535	KRB4	...	113
KRB5	ENCRYPTION	TCP-0-65535 UDP-0-65535	KRB5	TCP-543 UDP-88	27
KUOOO	PEER-TO-PEER-FILE-SHARING	UDP-0-65535	KUOOO	UDP-7000	78
LDAP	ENTERPRISE-INFRASTRUCTURE	TCP-0-65535	LDAP	TCP-389	111
LOTUSNOTES	MESSAGING	TCP-0-65535	LOTUS-NOTES	TCP-1352	134

You can double-click the table entry to view additional details, including the signature pattern regular expression to match in client-to-server and server-to-client directions. [Figure 13 on page 75](#) shows the general properties of the predefined application object for HTTP.

Figure 112: NSM Object Manager: Predefined Application: General Tab

HTTP - Predefined

General

Detector

General

Name

HTTP

Application Category

WEB

Supported Platforms

idp-jservices9.4

idp-jsr9.5

idp-jsrx9.4

idp-sos3.5.0

idp-sos3.5.134268

Port Ranges

TCP Port Range(s)

0-65535

UDP Port Range(s)

Close

Figure 14 on page 76 shows the signature properties of the predefined application object for HTTP.

Figure 113: NSM Object Manager: Predefined Application: Detector Tab

**HTTP - Predefined**

**General** | Detector

**Port Binding**

Application Type: HTTP

TCP Port Binding: 80,3128,8000,8080

UDP Port Binding:

**Signature**

**Client-to-server**

DFA Pattern: (\\OPTIONS|HEAD|GET|POST|PUT|B?DELETE|TRACE|SEARCH|B?PROPFIND|PROPPATCH|MKCOL|B?COPY|B?MOVE|LOCK|UNLOCK|CHECKOUT|

PCRE Pattern:

**Server-to-client**

DFA Pattern: (.\*HTTP/1\\.([01])s\\.?.?u<[\\DOCTYPE]\\u\\.?.?u<[\\HTML]\\u\\.?.?u<[\\?xml]\\u[\\Content-type: ].\*)

PCRE Pattern:

Minimum data length: 20

Signature Match Order: 122

Close

**Action** To view the table that lists predefined application objects:

1. In the Object Manager, select **Application Objects**.
2. Click the **Predefined Application Objects** tab.
3. Click a column heading to sort the table by the column property. Double-click the table row entry for the application object to display additional details.



**NOTE:** You cannot edit or delete predefined application objects. You can create custom application objects.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

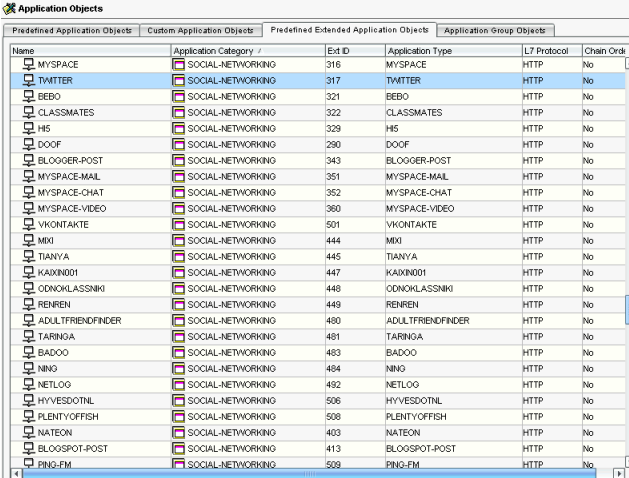
- [Using Application Objects on page 73](#)

## Viewing Predefined Extended Application Objects (NSM Procedure)

**Purpose** Extended application objects, also called nested applications, identify Web 2.0 applications running over HTTP. Extended application objects are predefined objects developed by the Juniper Networks Security Center (J-Security Center) and distributed during NSM signature database updates. You can use extended application objects in APE rules to treat various Web 2.0 applications running over HTTP differently.

[Figure 114 on page 290](#) shows the NSM Object Manager Predefined Extended Application Objects tab.

**Figure 114: NSM Object Manager: Predefined Extended Application Objects**



Name	Application Category	Ext ID	Application Type	L7 Protocol	Chain Ord
MYSPACE	SOCIAL-NETWORKING	316	MYSPACE	HTTP	No
TWITTER	SOCIAL-NETWORKING	317	TWITTER	HTTP	No
BEBO	SOCIAL-NETWORKING	321	BEBO	HTTP	No
CLASSMATES	SOCIAL-NETWORKING	322	CLASSMATES	HTTP	No
HS	SOCIAL-NETWORKING	329	HS	HTTP	No
DOOF	SOCIAL-NETWORKING	290	DOOF	HTTP	No
BLOGGER-POST	SOCIAL-NETWORKING	343	BLOGGER-POST	HTTP	No
MYSACE-MAIL	SOCIAL-NETWORKING	351	MYSACE-MAIL	HTTP	No
MYSACE-CHAT	SOCIAL-NETWORKING	352	MYSACE-CHAT	HTTP	No
MYSACE-VIDEO	SOCIAL-NETWORKING	360	MYSACE-VIDEO	HTTP	No
VKONTAKTE	SOCIAL-NETWORKING	501	VKONTAKTE	HTTP	No
MXI	SOCIAL-NETWORKING	444	MXI	HTTP	No
TIANYA	SOCIAL-NETWORKING	445	TIANYA	HTTP	No
KAXIN001	SOCIAL-NETWORKING	447	KAXIN001	HTTP	No
ODNOKLASSNIKI	SOCIAL-NETWORKING	448	ODNOKLASSNIKI	HTTP	No
RENREN	SOCIAL-NETWORKING	449	RENREN	HTTP	No
ADULTFRIENDFINDER	SOCIAL-NETWORKING	480	ADULTFRIENDFINDER	HTTP	No
TARINGA	SOCIAL-NETWORKING	481	TARINGA	HTTP	No
BADOO	SOCIAL-NETWORKING	483	BADOO	HTTP	No
NING	SOCIAL-NETWORKING	484	NING	HTTP	No
NETLOG	SOCIAL-NETWORKING	492	NETLOG	HTTP	No
HYVESDOTNL	SOCIAL-NETWORKING	506	HYVESDOTNL	HTTP	No
PLENTYOFFISH	SOCIAL-NETWORKING	508	PLENTYOFFISH	HTTP	No
NATEON	SOCIAL-NETWORKING	403	NATEON	HTTP	No
BLOGSPOT-POST	SOCIAL-NETWORKING	413	BLOGSPOT-POST	HTTP	No
PING-FM	SOCIAL-NETWORKING	509	PING-FM	HTTP	No

You can double-click the table entry to view additional details, including the matching HTTP context, signature pattern, and client-to-server or server-to-client direction. [Figure 16 on page 78](#) shows the properties of the HTTP:Facebook-Access application object.

Figure 115: NSM Object Manager: Extended Application Details

FACEBOOK-ACCESS - Predefined

General

Name: FACEBOOK-ACCESS

L7 Protocol: HTTP

Chain Order: No

Application type: FACEBOOK-ACCESS

Maximum Transactions: none

Signature Match Order: 33323

Members

Member	Context	pattern	direction
m01	http-header-host	(.*)?(facebook\.com fbcdn\.net)	CTS

Close

**Action** To view table listings of predefined extended application objects:

1. In the Object Manager, select **Application Objects**.
2. Click **Predefined Extended Application Objects** tab.
3. Double-click the table row entry to display additional details.



**NOTE:** You cannot edit or delete predefined extended application objects. You cannot create custom extended application objects.

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)

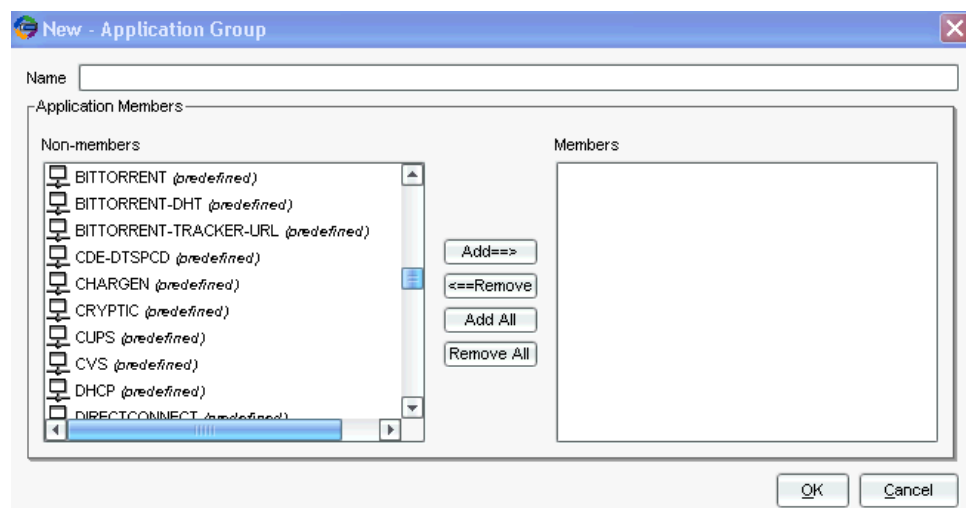
## Creating Application Groups (NSM Procedure)

You use the NSM Object Manager to create application groups.

To create an application group:

1. In the NSM Object Manager, select **Application Objects**.
2. Click the **Application Group Objects** tab.
3. Click the + icon to display the New Application Group dialog box, shown in [Figure 116 on page 292](#).
4. Give the group a name. Then use the selector controls to add or remove members to or from the group.
5. Click **OK** to save the object.

Figure 116: NSM Object Manager: New Application Group



### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)
- [APE Rulebase Example: Using Extended Application Objects on page 165](#)

## Creating a Custom Application (NSM Procedure)

You use the NSM Object Manager to create a custom application.



To create a custom application object:

1. In the NSM Object Manager, select **Application Objects**.
2. Click the **Custom Application Objects** tab.
3. Click the + icon to display the New Custom Application dialog box, shown in [Figure 117 on page 293](#).
4. Configure custom application properties, as described in [Table 87 on page 294](#).
5. Click **OK** to save the object.

**Figure 117: NSM Object Manager: New Custom Application Dialog Box**

The screenshot shows a window titled "New - Custom" with a close button in the top right corner. Inside the window, there are two tabs: "General" and "Detector". The "General" tab is selected. The "General" tab contains several sections:

- Port Binding**: This section contains three text input fields: "Application Type", "TCP Port Binding", and "UDP Port Binding".
- Signature**: This section contains two sub-sections:
  - Client-to-server**: Contains a "DFA Pattern" text input field.
  - Server-to-client**: Contains a "DFA Pattern" text input field.
- Minimum data length**: A dropdown menu currently set to "none".
- Signature Match Order**: A dropdown menu currently set to "none".

At the bottom right of the dialog box, there are two buttons: "OK" and "Cancel".

Table 87: NSM Object Manager: Custom Application Objects

Tab	Property	Configuration Guidelines
General	Name	Specify a descriptive name. Use the conventions of the predefined application object names as a model.
	Application Category	Specify an application category. Use the same categories as the predefined application objects, or specify a new category if needed.
	Supported Platforms	Click the edit icon to display the selection box. Then select the platforms you plan to test against.
	Port Ranges	Specify the range of TCP and UDP ports where the application might run. The application is identified only if the server port is within the specified range.
Detector	Port Binding	(Optional) Specify the standard ports on which the application usually runs.
	Signature	Specify a pattern match for client-to-server and server-to-client directions. IDP OS Release 5.1 supports only DFA patterns, not PCRE.
	Minimum data length	Specify a minimum data length to examine to match this pattern.
	Signature Match Order	Specify a signature match order. Order numbers are relative to each other. In cases where traffic matches multiple objects, an application object with the lower signature match-order number is considered the match. Be sure to examine all applications that might have the same protocol, port, and pattern; and then select a relative match order suited for the results you expect.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Application Objects Task Summary on page 286](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)
- [APE Rulebase Example: Matching Custom Application Objects on page 171](#)

# Configuring Logging Features

- IDP Series Logs and Reports in NSM Task Summary on page 295
- Configuring Interface Aliasing (ACM Procedure) on page 296
- Configuring Log Storage Limits on page 297
- Configuring Log Suppression (NSM Procedure) on page 298
- Configuring an SNMP Agent (NSM Procedure) on page 299
- Configuring Syslog Collection (NSM Procedure) on page 301
- Enabling Collection of Packet Data in NSM Logs (NSM Procedure) on page 303

## IDP Series Logs and Reports in NSM Task Summary

---

IDP Series devices generate logs about *device status* based on built-in criteria and about *security events* based on the security policy notification settings. These logs are automatically sent to the NSM GUI server and can be viewed in the NSM log viewer.

IDP Series administration includes the following log-related tasks:

- Viewing device status, logs, and reports.
- Viewing attack logs and reports.
- Viewing application usage logs and reports.
- Configuring interface aliasing, if you want to identify IDP Series traffic interfaces by name in logs and reports.
- Configuring log suppression, if you want to reduce the number of identical log files.
- Configuring communication with an SNMP or syslog server, if you use external log programs to view alerts or analyze or archive log data.
- Ensuring collection of packet data in NSM logs is enabled, if you want to drill into packet data from NSM logs.



**NOTE:** To avoid issues with reports, we highly recommend that you synchronize the network clocks for all devices to the same NTP server. For example, the network clocks for all IDP devices and NSM clients should be synchronized to the NTP server specified in the NSM configuration.

## Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing Device Status \(NSM Procedure\) on page 448](#)
- [Using NSM Logs on page 453](#)
- [Viewing NSM Predefined Reports \(NSM Procedure\) on page 469](#)
- [Configuring Interface Aliasing \(ACM Procedure\) on page 296](#)
- [Configuring Log Suppression \(NSM Procedure\) on page 298](#)
- [Configuring an SNMP Agent \(NSM Procedure\) on page 299](#)
- [Configuring Syslog Collection \(NSM Procedure\) on page 301](#)
- [Enabling Collection of Packet Data in NSM Logs \(NSM Procedure\) on page 303](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

## Configuring Interface Aliasing (ACM Procedure)

You configure interface aliasing if you want the traffic interface fields in logs to display a text string instead of the default traffic interface names. For IDP Series devices, the interface fields in the logs that appear in NSM are srcIntf and dstIntf (source and destination interfaces). The default traffic interface names are eth2, eth3, and so forth.

[Figure 118 on page 296](#) shows the ACM Configure Network Interface Hardware page, where you can configure interface aliasing.

**Figure 118: ACM Configure Network Interface Hardware Page**

**Configure Network Interface Hardware**

In this step, you can optionally force the speed and duplex settings of the network interfaces. If you decide to force the settings of an interface, be sure to do the same on your switch.

**Note:** If you change any of the interface settings below, the IDP appliance must be rebooted for the changes to take effect. If you want the IDP to be automatically rebooted after this configuration session, click the checkbox at the bottom of the page. (You will have the opportunity to change your mind at the end of the configuration session.)

Interface	Alias	Hardware	Mode
eth0	mg-idp8200	Gigabit Ethernet	auto
eth1	he-idp8200	Gigabit Ethernet	auto
eth2	eth2-idp8200-ex024	10 Gigabit Ethernet	auto
eth3	eth3-idp8200-ex055	10 Gigabit Ethernet	auto
eth4		Gigabit Ethernet	auto
eth5		Gigabit Ethernet	auto
eth6		Gigabit Ethernet	auto
eth7		Gigabit Ethernet	auto
eth8		Gigabit Ethernet	auto
eth9		Gigabit Ethernet	auto
eth10		Gigabit Ethernet	auto
eth11		Gigabit Ethernet	auto
eth12		Gigabit Ethernet	auto
eth13		Gigabit Ethernet	auto
eth14		Gigabit Ethernet	auto
eth15		Gigabit Ethernet	auto

☐ Reboot IDP at the end so changes can take effect.

[Next Step](#)

To configure interface aliasing:

1. Connect to the Appliance Configuration Manager (ACM).
2. From the main menu, click **Reconfigure Network Interface Hardware**.
3. Specify an alias for a traffic interface by typing a string in the Alias box.
4. Save and apply your changes.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Connecting to ACM on page 191](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## Configuring Log Storage Limits

You can modify log storage if you want to change the number of logs and packet logs stored on the IDP Series device. [Figure 119 on page 297](#) shows the NSM Report Settings page, where you configure log storage limits.

**Figure 119: NSM Device Configuration Editor: Report Settings**

The screenshot shows the 'idp-75 - Device' configuration window. On the left is a tree view with 'Report Settings' selected. The main area is divided into two sections: 'Report Settings' and 'SNMP Settings'. In 'Report Settings', 'Enable Syslog' is unchecked, 'Syslog Server IP' is empty, 'Include packet data in log' is checked, 'Maximum number of packet captures that can be stored' is set to 10000, and 'Maximum number of logs that can be stored' is set to 50000. In 'SNMP Settings', 'Enable SNMP' is unchecked, 'SNMP Read Only Community' is 'public', 'SNMP Manager IP' is '127.0.0.1', 'SNMP Contact' is 'admin', and 'SNMP Location' is 'juniper'. Below these are icons for adding, deleting, and saving configurations, followed by a table with headers 'Network/Host IP' and 'Network/Host Netmask'. At the bottom right are 'OK', 'Cancel', and 'Apply' buttons.

To modify log storage limits:

1. In the NSM Device Manager, double-click the IDP Series device to display the configuration editor.
2. Click **Report Settings**.
3. Complete the settings related to log storage limits described in [Table 88 on page 298](#).

**Table 88: IDP Series Device Configuration: Log Storage Limit Settings**

Setting	Function
Maximum number of packet captures that can be stored	Determines the limit for packet capture files stored on the IDP Series device. The default is 10,000. The minimum value is 1,000. The maximum is 65,535.
Maximum number of logs that can be stored	<p>Determines the limit of how many log files are stored on the IDP Series device before the older logs are pruned. The default is 50,000. The minimum value is 1,000. The maximum is 65,535.</p> <p><b>NOTE:</b> This setting sets a limit for log files and not log entries. A log file may contain many logs entries.</p>

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Developing a Log Storage Strategy on page 332](#)

## Configuring Log Suppression (NSM Procedure)

You configure log suppression if you want to reduce the number of logs displayed in the NSM log viewer. If you enable log suppression, NSM displays a single record for multiple occurrences of similar events, along with a count of all such occurrences. Logs that match all elements of a tuple but trigger different IDP rulebase rules are treated as non-similar events.



**NOTE:** When examining log records where log suppression has been applied (logs for which counts are given), you might encounter difficulty analyzing any packet captures contained therein. This is because the packets might have different destination addresses, or even though tuples are matching, different patterns might match to a single custom signature.

To enable and configure log suppression:

1. In the NSM Device Manager, double-click the IDP Series device to display the configuration editor.
2. Click **Sensor Settings**.

3. Click **Parameters**.
4. Complete the settings related to log suppression described in [Table 89 on page 299](#).

**Table 89: IDP Series Device Configuration: Log Suppression Settings**

Setting	Description
Enable log suppression	Log suppression is enabled by default. Use this setting to turn log suppression off and on.
Include destination IPs when performing log suppression	When log suppression is enabled, multiple occurrences of events with the same source IP, service, and matching attack object generate a single log record with a count of occurrences. If you enable this option, log suppression combines log records for events with the same destination IP.
Number of log occurrences after which log suppression begins	This number represents the number of identical log records received before suppression starts. The default is 1 (meaning log suppression begins with the first redundancy).
Maximum number of logs that log suppression can operate on	When log suppression is enabled, the IDP Series device must cache log records so that it can identify when multiple occurrences of the same event occur. This number represents the number of log records cached for this purpose. The default is 16,384 log records.
Time (seconds) after which suppressed logs will be reported	When log suppression is enabled, the IDP Series device maintains a count of multiple occurrences of the same event. This number represents the number of seconds that pass before IDP reports a single log entry containing the count of occurrences. The default is 10 seconds.

- Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## Configuring an SNMP Agent (NSM Procedure)

You configure an SNMP agent in order to send device event logs to an SNMP server.

You can configure an SNMP agent for NSM (if you want to send the NSM collection to SNMP) as well as an SNMP agent for each IDP Series device. [Figure 120 on page 300](#) shows the NSM Report Settings page, where you configure SNMP.

Figure 120: NSM Device Configuration Editor: Report Settings

The screenshot shows the 'idp-10.209.83.5 - Device' configuration window. The left sidebar has a tree view with 'Report Settings' highlighted. The main panel is divided into two sections: 'Report Settings' and 'SNMP Settings'.  
**Report Settings:**  
 - 'Enable Syslog' is checked.  
 - 'Syslog Server IP' is 10.209.89.42.  
 - 'Syslog Server Port' is 514.  
 - 'Protocol' is set to UDP.  
 - 'Include packet data in log' is checked.  
 - 'Maximum number of packet captures that can be stored' is 10000.  
 - 'Maximum number of logs that can be stored' is 50000.  
**SNMP Settings:**  
 - 'Enable SNMP' is unchecked.  
 - 'SNMP Read Only Community' is public.  
 - 'SNMP Manager IP' is 127.0.0.1.  
 - 'SNMP Contact' is admin.  
 - 'SNMP Location' is juniper.  
 At the bottom of the window are 'OK', 'Cancel', and 'Apply' buttons.

For instructions on how to configure an SNMP agent for NSM, see the NSM online Help.

To configure an SNMP agent for a single IDP Series device:

1. In the NSM Device Manager, double-click the IDP Series device to display the device configuration editor.
2. Click **Report Settings**.
3. Complete SNMP settings as described in [Table 90 on page 300](#).
4. Click **OK**.

Table 90: IDP Series Device Configuration: SNMP Settings

Setting	Description
Enable SNMP	Enables forwarding to a network management system that reads SNMP.
SNMP Read Only Community	Specifies the read-only community string, which is like a password used for the exchange between the IDP Series device and the network management system.
SNMP Manager IP	Specifies the IP address of the SNMP server.



Table 90: IDP Series Device Configuration: SNMP Settings (*continued*)

Setting	Description
SNMP Contact	Specifies an e-mail address for the IDP administrator contact. The contact is included in SNMP communications. If the network management system encounters a problem with the SNMP communication, an administrator can use the contact information to follow up.
SNMP Location	Specifies the location of the IDP Series device. Location is included in SNMP communications.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## Configuring Syslog Collection (NSM Procedure)

You configure syslog settings if you want to forward a copy of IDP Series device logs to a syslog server.

You have the option of configuring NSM to forward a copy of its log collection to a syslog server or configuring syslog settings for each IDP Series device. [Figure 121 on page 302](#) shows the NSM Report Settings page, where you configure syslog settings.

Figure 121: NSM Device Configuration Editor: Report Settings

The screenshot shows the 'Report Settings' configuration window for an IDP device. The left sidebar contains a tree view with the following items: Info, Interfaces, Anti-Spoof Settings, Sensor Settings, Profiler Settings, and Report Settings (which is highlighted). The main configuration area is titled 'Report Settings' and contains the following fields:

- Enable Syslog:** A checked checkbox.
- Syslog Server IP:** A text field containing '10.209.89.42'.
- Syslog Server Port:** A spin box set to '514'.
- Protocol:** A dropdown menu showing 'UDP'.
- Include packet data in log:** A checked checkbox.
- Maximum number of packet captures that can be stored:** A spin box set to '10000'.
- Maximum number of logs that can be stored:** A spin box set to '50000'.
- SNMP Settings:** A section header.
- Enable SNMP:** An unchecked checkbox.
- SNMP Read Only Community:** A text field containing 'public'.
- SNMP Manager IP:** A text field containing '127.0.0.1'.
- SNMP Contact:** A text field containing 'admin'.
- SNMP Location:** A text field containing 'juniper'.

At the bottom of the window are three buttons: 'OK', 'Cancel', and 'Apply'.

To configure syslog forwarding for NSM, see the NSM online Help.

To configure syslog forwarding for a single IDP Series device:

1. In the NSM Device Manager, double-click the IDP Series device to display the device configuration editor.
2. Click **Report Settings**.
3. Select **Enable Syslog**.
4. Specify the syslog server IP address, port, and protocol. Port 514 and UDP are industry standards and are used as the default.
5. Specify whether to forward packet logs to the syslog server.
6. Click **OK**.

The following example shows a syslog message record:

```
Mar 31 18:04:31 10.209.83.9 1 2010-06-23T18:05:55 10.209.83.9 Jnpr Syslog 23414
1
[syslog@juniper.net dayId="20100623" recordId="0" timeRecv="2010/06/23 18:05:55"
timeGen="2010/06/23 18:05:51"
domain="" devDomVer2="0" device_ip="10.209.83.9" cat="Config" attack=""
srcZn="NULL"
srcIntf="" srcAddr="0.0.0.0" srcPort="0" natSrcAddr="NULL" natSrcPort="0"]
```

```
dstZn="NULL" dstIntf="NULL" dstAddr="0.0.0.0" dstPort="0" natDstAddr="NULL"
natDstPort="0" protocol="IP"
ruleDomain="" ruleVer="0" policy="" rulebase="NONE" ruleNo="0" action="NONE"
severity="INFO" alert="no"
elapsedTime="0" inbytes="0" outbytes="0" totBytes="0" inPak="0" outPak="0"
totPak="0" repCount="0" packetData="no" varEnum="0"
misc="Interface eth2,eth3 is in Normal State" user="NULL" app="NULL" uri="NULL"]
```

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## Enabling Collection of Packet Data in NSM Logs (NSM Procedure)

When you configure security policy rule notification options, you have the option of logging the packets surrounding the security event.

Packet capture logs are stored locally on the IDP Series device in numbered directories: `/usr/idp/device/var/pktlogs/0/`, `/usr/idp/device/var/pktlogs/1/`, `/usr/idp/device/var/pktlogs/2/`, and so forth. Each directory stores 100,000 logs. The total number of directories is determined by the value set in NSM for **Maximum number of packet captures that can be stored**. For example, if you retain the default (10,000), all packet logs are stored in `/usr/idp/device/var/pktlogs/0/`. If you set the maximum to 200,000, packet logs are stored two directories: `/usr/idp/device/var/pktlogs/0/` and `/usr/idp/device/var/pktlogs/1/`.

The first 100,000 packet capture logs are stored in `/usr/idp/device/var/pktlogs/0/`. Files are named `1.pcap`, `2.pcap`, ..., `100000.pcap`. The next 100,000 logs are stored in `/usr/idp/device/var/pktlogs/1/`. Files are named `1.pcap`, `2.pcap`, ..., `100000.pcap`. The log agent continues to create directories and files in this manner until the user-specified limit is reached or the disk usage for the partition reaches 90% capacity.

When the *user-specified maximum* is reached, the log agent begins overwriting packet log files, beginning with `/usr/idp/device/var/pklogs/0/1.pcap`.

If the packet capture repository reaches the *disk limit* before the user-specified limit:

1. The log agent deletes all 100,000 logs in the first directory, `/usr/idp/device/var/pktlogs/0/`, in order to reuse the directory and disk space.
2. The next logs are written to `/usr/idp/device/var/pktlogs/0/` and the files are named `1.pcap`, `2.pcap`, ..., `100000.pcap`.
3. When the limit is reached again, the log agent deletes all of the logs in the next directory, `/usr/idp/device/var/pktlogs/1/`. It continues writing in the current directory until it reaches `100000.pcap`.
4. Then, it begins writing in the next directory, which had been emptied in the previous step.

The IDP Series device forwards the packet data to NSM according to your NSM Report Settings:

- **Include packet data in log** selected. Forwards the packet capture to NSM automatically when it sends the corresponding event log.
- **Include packet data in log** not selected. Forwards a reference to the packet capture file to NSM automatically but forwards the packet data itself only on-demand (when an NSM user takes action to display the packet data).

Figure 122: NSM Device Configuration Editor: Report Settings

To configure packet log collection:

1. In the NSM Device Manager, double-click the IDP Series device to display the configuration editor.
2. Click **Report Settings**.
3. Select **Include packet data in log**.
4. Optionally, modify the default for **Maximum number of packet captures that can be stored**. The maximum value you can specify is 102,400,000.
5. Click **Apply** and **OK** to save your settings.



**NOTE:** Be careful when modifying the maximum packet captures limit. If you first configure a large limit and later configure a smaller limit, you might delete directories of logs. For example, suppose you first set a maximum 1,000,000. The log agent begins storing logs in up to 10 log directories. Later, you change the maximum to 100,000. The log agent cleans up the previous configuration, deleting unnecessary directories 1-9. Before you change the setting to a lower value, be sure you have copied all the logs you want saved to a remote location.



**NOTE:** You might encounter unexpected behavior if the following circumstances apply:

- You change the maximum to a lower value—from 200,000 to 100,000, for example.
- At the same time, the agent process is handling requests from NSM for logs from `/usr/idp/device/var/pktlogs/` subdirectories.

In the typical case, we expect the agent to mark unnecessary subdirectories for deletion and clean them up after the new maximum is applied. If the agent has locked a subdirectory marked for deletion in order to retrieve files for NSM, it will not delete the subdirectory.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Example: Packet Logging Workflow on page 145](#)



## CHAPTER 29

# Using the scio Command to Implement Advanced Features

- [scio Configuration Commands Task Summary on page 307](#)
- [Using the SSL Private Server Key to Enable Inspection of SSL Traffic on page 308](#)
- [Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic on page 311](#)
- [Exempting HTTPS Traffic from Inspection on page 312](#)
- [Enabling Inspection of GRE Traffic on page 313](#)
- [Enabling Inspection of GTP Traffic on page 315](#)
- [Enabling Inspection of IPsec VPN Traffic on page 317](#)
- [Enabling Inspection of MPLS Traffic on page 318](#)
- [Enabling the Flow Bypass Feature on page 319](#)
- [Configuring a Default Rate Limit on page 321](#)
- [Enabling Per-User Rate Limiting for User-Role-Based Rules on page 321](#)
- [Configuring Advanced Settings for the User-Role-Based Policy Feature on page 322](#)

### scio Configuration Commands Task Summary

You can use the **scio** configuration commands to enable advanced features, such as:

- Inspection of encrypted or encapsulated traffic
- Flow bypass when the IDP engine is under heavy stress
- Advanced settings for the APE rulebase
- Advanced settings for the user-role-based feature

#### **Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [scio const on page 505](#)
- [Using the SSL Private Server Key to Enable Inspection of SSL Traffic on page 308](#)
- [Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic on page 311](#)
- [Exempting HTTPS Traffic from Inspection on page 312](#)

- [Enabling Inspection of GRE Traffic on page 313](#)
- [Enabling Inspection of GTP Traffic on page 315](#)
- [Enabling Inspection of IPsec VPN Traffic on page 317](#)
- [Enabling Inspection of MPLS Traffic on page 318](#)
- [Enabling the Flow Bypass Feature on page 319](#)

---

## Using the SSL Private Server Key to Enable Inspection of SSL Traffic

---

To inspect the HTTP payload of HTTPS traffic, the IDP Series device must first decrypt the session. Your security policy can examine both the SSL session and the HTTP payload.

The IDP Series solution supports SSL inspection in two ways:

- Using server private keys. Use this method when inspecting traffic to internal servers where you have access to the server private key.
- Using the SSL forward proxy feature. Use this method when the server private key method is not practical (for example, for traffic to servers on the WWW).



.....

**NOTE:** If you enable both methods, the IDP Series device performs SSL inspection using the SSL forward proxy method and does not use the server private keys.

.....

The following procedure provides the basic steps you take to implement inspection using the SSL server private keys.

To use the SSL private server key to enable inspection of SSL traffic:

1. Log into the CLI as **admin** and enter **su -** to switch to root.



2. Add the private keys for known destination servers to the IDP Series device keystore:

- a. Use SCP or FTP to copy your SSL server private key file to the IDP Series device. The IDP Series device does not run an FTP server, so you have to initiate the FTP session from the IDP Series device.
- b. If necessary, change permissions so you can use the **scio** utility to manage the file. For example:

```
[root@default host admin]# chmod 777 /tmp/server.key
```



**NOTE:** Changing permissions for the file should suffice. If you still encounter issues, change ownership as well:

```
[root@default host admin]# chown idp:idp /tmp/server.key
```

- c. Add the key to the IDP Series device keystore using the following syntax:

```
[root@default host admin]# scio ssl add key key_path [password password] server server_IP
```

For example:

```
[root@default host admin]# scio ssl add key /tmp/server.key server 10.1.1.1
```

- d. Display the key ID from the IDP Series device keystore by entering the following command:

```
[root@default host admin]# scio ssl list all
```

- e. Add any other servers that use the same key using the following syntax:

```
[root@default host admin]# scio ssl add server server_IP key key_ID
```

3. Enter the following command to enable decryption:

```
[root@default host admin]# scio const -s s0 set sc_ssl_decryption 1
scio: setting sc_ssl_decryption to 0x1
```

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the constant below the line `user_start_end()`. For example:

```
user_start_end()
{
    $SCIO const -s s0 set sc_ssl_decryption 1
}
```

3. Save the file.

4. Restart the IDP engine:

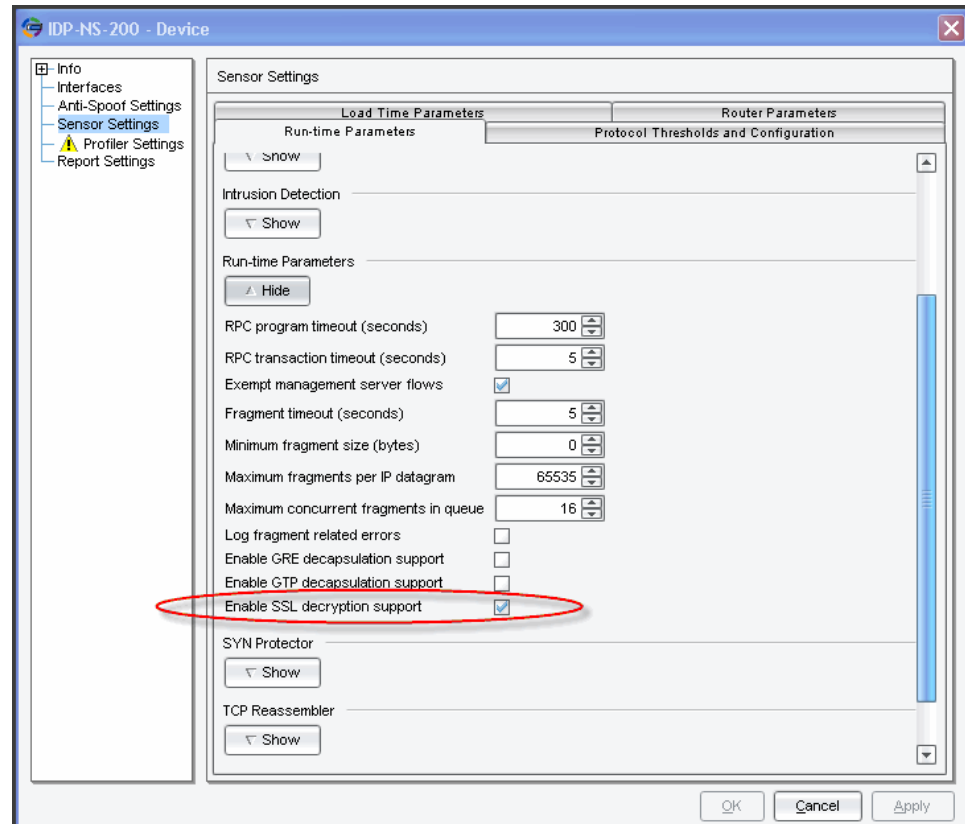
```
[root@default host admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

You can also use the NSM Device Manager to turn on the SSL decryption feature. However, you cannot use NSM to manage the SSL keys.

Figure 123 on page 310 shows the location of the SSL decryption setting in NSM.

Figure 123: NSM Device Manager: SSL Decryption Setting



To enable SSL decryption with NSM:

1. In the NSM Device Manager, double-click the IDP Series device to display the device configuration editor.
2. Click **Sensor Settings**.
3. Click the **Run-time Parameters** tab.
4. Expand the **Run-time Parameters** group.
5. Select **Enable SSL decryption support**.
6. Click **OK**.
7. Push the updated configuration from NSM to the IDP Series device.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic on page 311](#)
- [scio ssl on page 531](#)
- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of SSL Traffic Overview on page 113](#)

## Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic

To inspect the HTTP payload of HTTPS traffic, the IDP Series device must first decrypt it. Your security policy can examine both the SSL session and the decrypted HTTP payload.

The IDP Series solution supports SSL inspection in two ways:

- Using server private keys. Use this method when inspecting traffic to internal servers where you have access to the server private key.
- Using the SSL forward proxy feature. Use this method when the server private key method is not practical (for example, for traffic to servers on the WWW).



**NOTE:** If you enable both methods, the IDP Series device performs SSL inspection using the SSL forward proxy method and does not use the server private keys.

The following procedure provides the basic steps you take to implement the SSL forward proxy feature.

To implement the SSL forward proxy feature:

1. Generate the root certificate authority (CA) that the IDP Series device uses to create and sign new certificates used in SSL proxy operations. The following example creates a root CA:

```
[root@default host admin]# scio ssl ca create US CA Sunnyvale 'Juniper Networks Inc.' 'SSL Inspection policy' 'Juniper IT Services' 'admin@juniper.net' 1024
```

2. Verify the CA was added:

```
[root@default host admin]# scio ssl ca show
serial=8E0012848A2D7CCD
subject= /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks Inc./OU=SSL Inspection policy/CN=Juniper IT Services/emailAddress=admin@juniper.net
issuer= /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks Inc./OU=SSL Inspection policy/CN=Juniper IT Services/emailAddress=admin@juniper.net
notBefore=Jun 25 22:13:23 2009 GMT
notAfter=Jun 23 22:13:23 2019 GMT
```

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Using the SSL Private Server Key to Enable Inspection of SSL Traffic on page 308](#)
- [Troubleshooting SSL Inspection on page 600](#)
- [Disabling SSL Inspection on page 600](#)
- [scio ssl on page 531](#)

- [scio const on page 505](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Example: Implementing Inspection of Outbound SSL Traffic on page 179](#)
- [Inspection of SSL Traffic Overview on page 113](#)

---

## Exempting HTTPS Traffic from Inspection

You can use a whitelist to exempt from inspection traffic to specified HTTPS servers. If traffic matches a whitelist entry, it is passed through (not decrypted or inspected).



**NOTE:** The whitelist applies only to traffic processing based on the SSL forward proxy feature. You would not use a whitelist to exclude inspection of traffic to internal destination servers. If desired, you can use a security policy rule to exempt such traffic from inspection.

The whitelist is a text file you import into the IDP Series device, using the CLI. The following example shows the format of a whitelist file:

```
10.0.0.1
1.0.0.0/8
70.34.21.82
trustedsite.com
landing.trustedsearch.com
```

Each line in the whitelist file specifies the IP address or domain name for a destination server. To whitelist multiple sites with one entry, you can use an IP prefix to match address blocks and a domain suffix to include all subdomains.

The domain name in your whitelist should match the common name entry in the certificate presented by the destination server. For example, suppose the certificate for the E-Trade HTTPS server contains the following subject:

```
C=US, ST=Georgia, L=Alpharetta, O=ETRADE FINANCIAL CORPORATION, OU=Global
Information Security, CN=us.etrade.com
```

You can whitelist this site by adding the string **us.etrade.com** or the string **etrade.com** to your whitelist file.

To create a whitelist:

1. Log into the CLI as **admin** and enter **su -** to switch to root.
2. Use an editor like **vi** to create a whitelist file. For example:

```
[root@default host admin]# vi /tmp/whitelist.txt
10.0.0.1
1.0.0.0/8
70.34.21.82
```

```
etrade.com
bankofamerica.com
```

3. Run the following command to import the whitelist entries:

```
[root@default host admin]# scio ssl whitelist import /tmp/whitelist.txt
```



**NOTE:** The whitelist setting takes effect on sessions that are initiated after your change.



**NOTE:** To update the active whitelist, import an updated whitelist file. To clear the whitelist, import a file that contains only one empty line.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [scio ssl on page 531](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Example: Exempting Outbound SSL Traffic from Inspection on page 181](#)
- [Inspection of SSL Traffic Overview on page 113](#)

## Enabling Inspection of GRE Traffic

You can use the command-line interface (CLI) or Network and Security Manager (NSM) to enable inspection of generic routing encapsulation (GRE) encapsulated traffic. To enable inspection of encapsulated traffic, the IDP engine must first decapsulate it.

To enable and configure decapsulation from the CLI:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to enable decapsulation:

```
[root@default host admin]# scio const -s s0 set sc_gre_decapsulation 1
scio: setting sc_gre_decapsulation to 0x1
```

By default, the IDP engine decapsulates one layer.

3. Optional. Change the maximum decapsulation to two layers by entering the following command:

```
[root@default host admin]# scio const -s s0 set sc_max_decapsulation 2
scio: setting sc_max_decapsulation to 0x2
```

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the constant below the line `user_start_end()`. For example:

```

user_start_end()
{
$SCIO const -s s0 set sc_gre_decapsulation 1

}

```

3. Save the file.

4. Restart the IDP engine:

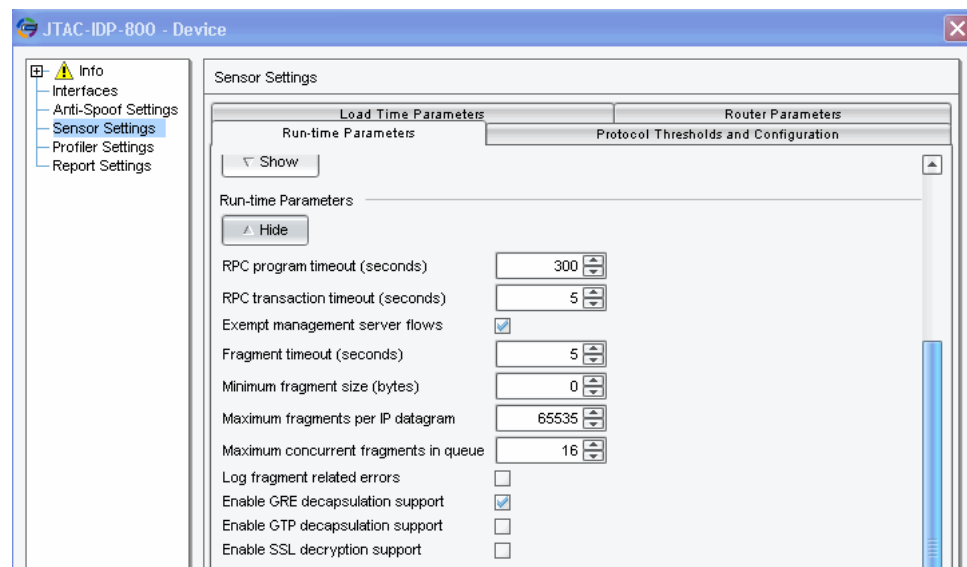
```
[root@defaulthost admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

You can also use Network and Security Manager (NSM) Device Manager to turn on the GRE decapsulation feature. However, you cannot use NSM to change the decapsulation layer setting.

Figure 124 on page 314 shows the location of the GRE support setting in NSM.

**Figure 124: NSM Device Manager: GRE Support Setting**



To enable decapsulation with NSM:

1. In the NSM Device Manager, double-click the IDP Series device to display the device configuration editor.
2. Click **Sensor Settings**.
3. Click the **Run-Time Parameters** tab.
4. Expand the **Run-Time Parameters** group.
5. Select **Enable GRE decapsulation support**.
6. Click **OK**.
7. Push the updated configuration from NSM to the IDP Series device.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of GRE Traffic Overview on page 111](#)

## Enabling Inspection of GTP Traffic

You can use the command-line interface (CLI) or Network and Security Manager (NSM) to enable inspection of GPRS tunnelling protocol (GTP) encapsulated traffic. To enable inspection of encapsulated traffic, the IDP engine must first decapsulate it.

To enable and configure decapsulation from the CLI:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to enable decapsulation:

```
[root@defaultthost admin]# scio const -s s0 set sc_gtp_decapsulation 1
scio: setting sc_gtp_decapsulation to 0x1
```

By default, the IDP engine decapsulates one layer.

3. Optional. Change the maximum decapsulation to two layers by entering the following command:

```
[root@defaultthost admin]# scio const -s s0 set sc_max_decapsulation 2
scio: setting sc_max_decapsulation to 0x2
```

You can also use the **scio const** command to change defaults for the timeout at which the IDP engine closes the GTP tunnel and for the maximum number of concurrent GTP tunnels the IDP engine can handle.

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the constant below the line `user_start_end()`. For example:

```
user_start_end()
{
    $SCIO const -s s0 set sc_gtp_decapsulation 1
}
```

3. Save the file.
4. Restart the IDP engine:

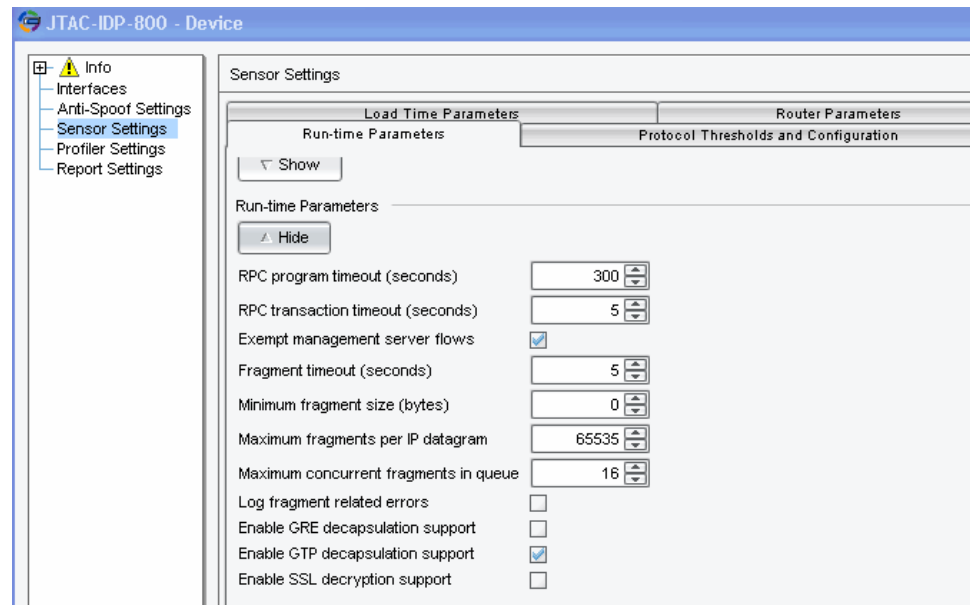
```
[root@defaultthost admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

You can also use Network and Security Manager (NSM) Device Manager to turn on the GTP decapsulation feature. However, you cannot use NSM to change the decapsulation layer setting.

Figure 125 on page 316 shows the location of the GTP support setting in NSM.

**Figure 125: NSM Device Manager: GTP Support Setting**



To enable GTP decapsulation (NSM):

1. In the NSM Device Manager, double-click the IDP Series device to display the device configuration editor.
2. Click **Sensor Settings**.
3. Click the **Run-Time Parameters** tab.
4. Expand the **Run-Time Parameters** group.
5. Select **Enable GTP decapsulation support**.
6. Click **OK**.
7. Push the updated configuration from NSM to the IDP Series device.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of GTP Traffic Overview on page 111](#)



## Enabling Inspection of IPsec VPN Traffic

Internet Protocol Security (IPsec) virtual private networks (VPNs) use the Encapsulating Security Payload (ESP) protocol and the NULL encryption algorithm to ensure the authenticity, integrity, and confidentiality of IP packets. You can use the command-line interface (CLI) to enable decapsulation of IPsec ESP NULL traffic so that the IDP engine can inspect it. You can configure decapsulation for one or two layers.

To enable and configure decapsulation:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to enable decapsulation:

```
[root@defaulthost admin]# scio const -s s0 set sc_null_esp_decapsulation 1
scio: setting sc_null_esp_decapsulation to 0x1
```

By default, the IDP engine decapsulates one layer.

3. Optional. Change the maximum decapsulation to two layers by entering the following commands:

```
[root@defaulthost admin]# scio const -s s0 set sc_max_decapsulation 2
scio: setting sc_max_decapsulation to 0x2
```

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the constant below the line `user_start_end()`. For example:

```
user_start_end()
{
    $SCIO const -s s0 set sc_null_esp_decapsulation 1
}
```

3. Save the file.
4. Restart the IDP engine:

```
[root@defaulthost admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of IPsec VPN Traffic Overview on page 112](#)

## Enabling Inspection of MPLS Traffic

---

Before the IDP engine can inspect the payload of Multiprotocol Label Switching (MPLS) traffic, it must decapsulate it. You can use the command-line interface to enable MPLS decapsulation.

For an overview of MPLS decapsulation support and limitations, see the *IDP Series Concepts and Examples Guide*.

To enable MPLS decapsulation:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to enable decapsulation:

```
[root@defaulthost admin]# scio const -s s0 set sc_mpls_decapsulation 1
scio: sc_mpls_decapsulation = 0x1
```

The value 0x1 indicates MPLS support is enabled.

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the **/usr/idp/device/bin/user\_funcs** file in a text editor, such as **vi**.
2. Locate the MPLS constant below the line **user\_start\_pre\_policy()**:

```
user_start_pre_policy ()
{
    # Disable ARP spoofing detection
    # -----
    # If you are running clusters with virtual MAC addresses, IDP will treat
    # these as spoofed ARP packets since the MAC addresses in the ethernet
    # frame will be different from what is inside the ARP request/response. If
    # you have multiple virtual routers, you need to perform this operation on
    # all defined virtual routers.
    #
    # $SCIO const -v vr0 set sc_arp_spoof_detect 0
    # $SCIO const -s s0 set sc_mpls_decapsulation 1
    return;
}
```

3. Uncomment the **\$SCIO const -s s0 set sc\_mpls\_decapsulation 1** line. For example:

```
user_start_pre_policy ()
{
    # Disable ARP spoofing detection
    # -----
    # If you are running clusters with virtual MAC addresses, IDP will treat
    # these as spoofed ARP packets since the MAC addresses in the ethernet
    # frame will be different from what is inside the ARP request/response. If
```

```
# you have multiple virtual routers, you need to perform this operation on
# all defined virtual routers.
#
# $SCIO const -v vr0 set sc_arp_spoof_detect 0
$SCIO const -s s0 set sc_mpls_decapsulation 1
return;

}
```

4. Save the file.

5. Restart the IDP engine:

```
[root@defaulthost admin]# idp.sh restart
```

Restarting the IDP process engine can take several moments.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying MPLS Decapsulation on page 494](#)
- [Disabling MPLS Decapsulation on page 601](#)
- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of MPLS Traffic Overview on page 112](#)

## Enabling the Flow Bypass Feature

The flow bypass feature prevents the IDP Series device from becoming a point of failure when the network is congested. With flow bypass enabled, when the IDP system packet receive queue reaches a rising threshold that you specify, the IDP engine marks the flow as a bypass flow and passes it through, uninspected. The IDP Series device passes through subsequent flows until the IDP system packet receive queue falls below a reset threshold that you also specify.

The flow bypass feature is not enabled by default.

For an overview of the flow bypass feature, see the *IDP Series Concepts and Examples Guide*.

To enable the flow bypass feature:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to enable flow bypass:

```
[root@defaulthost admin]# scio const -s s0:flow set sc_flow_bypass_enable 1
[root@defaulthost admin]#
```

By default, the system packet queue size utilization rising threshold is 90%; the reset threshold is 80%.

3. Optional. Change the rising threshold with the following command syntax:

```
scio const -s s0:flow set sc_flow_bypass_threshold_hi percent
```

For example:

```
[root@defaulthost admin]# scio const -s s0:flow set sc_flow_bypass_threshold_hi 95
scio: setting sc_flow_bypass_threshold_hi to 0x5f
[root@defaulthost admin]#
```

4. Optional. Change the reset threshold with the following command syntax:

```
scio const -s s0:flow set sc_flow_bypass_threshold_low percent
```

For example:

```
[root@defaulthost admin]# scio const -s s0:flow set sc_flow_bypass_threshold_low 85
scio: setting sc_flow_bypass_threshold_low to 0x55
[root@defaulthost admin]#
```

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the constant below the line `user_start_pre_policy ()`. For example:

```
user_start_pre_policy ()

{

    # Disable ARP spoofing detection
    # -----
    # If you are running clusters with virtual MAC addresses, IDP will treat
    # these as spoofed ARP packets since the MAC addresses in the ethernet
    # frame will be different from what is inside the ARP request/response. If
    # you have multiple virtual routers, you need to perform this operation on
    # all defined virtual routers.
    #
    # $SCIO const -v vr0 set sc_arp_spoof_detect 0
    # $SCIO const -s s0 set sc_mpls_decapsulation 1
    $SCIO const -s s0:flow set sc_flow_bypass_enable 1
    return;

}
```

3. Save the file.
4. Restart the IDP engine:

```
[root@defaulthost admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying the Flow Bypass Feature on page 495](#)
- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)

## Configuring a Default Rate Limit

The “default rate limit” is a limit applied to sessions that do not match APE rules. This option is disabled by default. You can enable it by specifying a rate limit for such sessions.



**NOTE:** If you have enabled per user rate limiting (also called per subscriber rate limiting), the default rate limit is applied per user. If not, the default rate limit is a maximum allocation for all sessions that do not match APE rules.

To assign a default rate limit for sessions that do not match APE rules:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the following command to show the current value:

```
[root@defaultthost admin]# scio const get sc_ape_default_rate_limit
scio: sc_ape_default_rate_limit = 0xffffffff
```

The default is 4,294,967,295 bps (0xffffffff in hexadecimal; 4,096 Mbps or .5 Gbps), which effectively turns off rate limiting for sessions that do not match APE rules.

3. If you want to set a rate limit for sessions that do not match APE rules, use the corresponding **set** command. The following example sets a limit of .25 Gbps:

```
[root@defaultthost admin]# scio const set sc_ape_default_rate_limit 2147483648
scio: setting sc_ape_default_rate_limit to 0x80000000
```

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Enabling Per-User Rate Limiting for User-Role-Based Rules on page 321](#)

## Enabling Per-User Rate Limiting for User-Role-Based Rules

If you implement user-role-based rules, you can apply rate limiting to all users who belongs to the specified role or to each user who belongs to the specified role. By default, rate limiting is applied to all users who belong to the specified role. You can change this setting with the command-line interface.

To enable per-user rate limiting:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to show the current value:

```
[root@defaultthost admin]# scio const -s s0 get sc_per_subscriber_ratelimit
scio: sc_per_subscriber_ratelimit = 0x0
```

0x0 indicates that per-subscriber rate limiting is disabled.

3. Enter the corresponding **set** command to enable per-subscriber rate limiting:

```
[root@defaultthost admin]# scio const -s s0 set sc_per_subscriber_ratelimit 1
```

scio: setting `sc_per_subscriber_ratelimit` to `0x1`

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Configuring a Default Rate Limit on page 321](#)

---

## Configuring Advanced Settings for the User-Role-Based Policy Feature

In most cases, we recommend you retain the defaults for the user role-based policy feature. These settings have been made configurable to support varying requirements for different deployment challenges.

By default:

- The IDP Series device sends a maximum of five logs per second to Juniper Networks IC Series Unified Access Control appliances. You can modify this value.
- User role-based rules are not processed if the IDP Series device loses connectivity with the IC Series for 30 seconds. You can modify this value.
- The user session table that is populated by the IC Series appliance and maintained on the IDP Series device contains a maximum of 50,000 users. You can change the maximum.

To change the threshold where lost connectivity stops processing of user role-based rules:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to show the current value:

```
[root@defaultthost admin]# scio const -s s0 get sc_ic_reconcile_timeout
scio: sc_ic_reconcile_timeout = 0x1E
```

The default is 30 seconds (0x1E).

3. Enter the following command to change this setting:

```
[root@defaultthost admin]# scio const -s s0 set sc_ic_reconcile_timeout 180
scio: sc_ic_reconcile_timeout = 0xB4
```

To change the maximum number of logs per second the IDP Series device sends to the IC Series appliance:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to show the current value:

```
[root@defaultthost admin]# scio user logs throttle show
5 Log(s)/Second.
[root@defaultthost admin]#
```

3. Enter the following command to change the value:

```
[root@defaultthost admin]# scio user logs throttle set 10
```

```
IC-Log Throttle limit set to '10'.  
[root@defaulthost admin]#
```

To change the maximum number of users in the user session table:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Open the **/usr/idp/device/bin/user\_funcs** file in a text editor, such as **vi**.
3. Locate the following line:  

```
export max_ic_users=50000
```
4. Edit the value for **max\_ic\_users**. Valid values are 1000 to 100,000.
5. Save the file and exit the editor.
6. Restart the IDP engine:

```
[root@defaulthost admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying Integration with an IC Series Unified Access Control Appliance on page 493](#)
- [scio const on page 505](#)
- [scio user on page 544](#)





## PART 4

# Administration

- [Managing the Profiler on page 327](#)
- [Logging on page 331](#)
- [Managing Security Policies on page 335](#)
- [Managing the IDP Device Configuration with NSM on page 343](#)
- [Managing IDP Processes on page 383](#)
- [Updating IDP Software on page 389](#)
- [Installing Traffic Interface I/O Modules on page 397](#)
- [Enabling Bypass and Peer Port Modulation on page 401](#)
- [Configuring the Management Interface on page 405](#)



## CHAPTER 30

# Managing the Profiler

- [Profiler Task Summary on page 327](#)
- [Starting and Stopping the Profiler \(NSM Procedure\) on page 328](#)
- [Managing the Profiler Database \(NSM Procedure\) on page 328](#)

### Profiler Task Summary

---

You use NSM Profiler to learn about your internal network so you can create effective security policies and minimize unnecessary log records. The Profiler queries and correlates attack logs and application usage logs from multiple IDP Series devices.

The Profiler feature is available only through Network and Security Manager (NSM).

IDP Series administration includes the following tasks related to the Profiler feature:

- Configuring Profiler options and preferences
- Starting and stopping the Profiler
- Viewing Profiler reports
- Managing the Profiler database

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring Profiler Options \(NSM Procedure\) on page 204](#)
- [Modifying Profiler Settings on page 210](#)
- [Starting and Stopping the Profiler \(NSM Procedure\) on page 328](#)
- [Using Profiler Viewer \(NSM Procedure\) on page 463](#)
- [Managing the Profiler Database \(NSM Procedure\) on page 328](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

## Starting and Stopping the Profiler (NSM Procedure)

---

You use Network and Security Manager (NSM) to start and stop the Profiler.

The Profiler is a service, located on the IDP Series device at `/usr/idp/device/bin/profiler.sh`.

To start the Profiler:

1. From the NSM main menu, select **Devices > IDP Profiler > Start Profiler**.
2. Select the devices on which you want to start the Profiler.
3. Click **OK**.

To stop the Profiler:

1. From the NSM main menu, select **Devices > IDP Profiler > Stop Profiler**.
2. Select the devices on which you want to stop the Profiler.
3. Click **OK**.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

## Managing the Profiler Database (NSM Procedure)

---

The following topics provide procedures for managing the Profiler database:

- [Displaying Profiler Database Information on page 328](#)
- [Querying the Profiler Database on page 328](#)
- [Purging the Profiler Database on page 329](#)

### Displaying Profiler Database Information

**Purpose** Data discovered by Profiler is stored in a database located on the NSM GUI server. Use the steps in this procedure to display information about the Profiler database.

**Action** To display Profiler database information:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Show DB Information** icon in the upper right corner to view specific details about the Profiler database, including the database size.

### Querying the Profiler Database

**Purpose** Data discovered by Profiler is stored in a database located on the NSM GUI server. Use the steps in this procedure to query the Profiler database.

**Action** To query records in the database:

1. Log into the NSM GUI server as the Postgres SQL user. By default, the Postgres SQL user is netscreen.
2. Navigate to the directory where the Profiler DB is located: `/usr/local/nsmpsql/bin`.
3. Run any Postgres SQL command. For example, you can type the following command:  
`./psql -d profilerDb`

## Purging the Profiler Database

Data discovered by Profiler is stored in a database located on the NSM GUI server. When the database reaches a maximum size (4 GB by default), it begins purging records (oldest first) automatically. The Profiler stops purging records when it reaches a certain set minimum size (3 GB by default).

Use the steps in this procedure to purge the Profiler database, if needed.

To change automatic purge settings, from the NSM main menu, select **Tools > Preferences** and modify the Profiler database settings.

To purge the database immediately:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Clear All DB** icon in the upper right corner.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)



## CHAPTER 31

# Logging

- [Developing a Logging Strategy on page 331](#)
- [Developing a Log Storage Strategy on page 332](#)

### Developing a Logging Strategy

---

Intrusion prevention systems can generate hundreds of logs per hour. In order to make the best use of the security logs, you should develop strategic approaches to the following administrative tasks:

- Fine-tuning the security policy rules.

Security policy rules determine the amount of logging performed by the IDP Series device, as well as automatic actions to take on offending traffic, such as dropping the session, sending a TCP reset, blocking the IP address from future connections, and so forth. See [“Example: Fine-Tuning a Security Policy” on page 48](#).

- Analyzing log event summaries and packet capture data.

By viewing log summaries, attack reference information, and packet data, you can verify whether the severity and actions associated with a security event are appropriate, whether refinements to your security policy are required, and whether further response actions are warranted. See [“Example: Using NSM Log Viewer Features” on page 139](#).

- Managing log and packet storage.

Your business log management and log storage policies determine where you store IDP Series device logs and security event logs. Your IDP Series device supports local logging, central collection by NSM, and forwarding to a syslog server. See [“Developing a Log Storage Strategy” on page 332](#).

#### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## Developing a Log Storage Strategy

This topic summarizes IDP log storage and log forwarding options so you can develop a log storage strategy suitable for your business. It includes the following sections:

- [Log Management Considerations on page 332](#)
- [Local Log Files and Directories on page 332](#)
- [NSM Log Collection on page 334](#)

### Log Management Considerations

An IDP Series device might generate hundreds of logs per day. Your log storage strategy depends on a number of factors:

- The nature of your business. Compliance with regulations or business agreements might determine where you collect logs or how often you retain them.
- Existing log management infrastructure. We recommend you become familiar with an use Network and Security Manager (NSM) as a central location for log analysis, but your previous investments in technology and training are also strong considerations.
- Distribution to the appropriate personnel for analysis is also a key consideration.

If your organization has not formalized a log management policy, consult the National Institute of Standards and Technology (NIST) publication, [Guide to Computer Security Log Management](#), for a treatment of the myriad considerations.

### Local Log Files and Directories

Logs are stored locally on the device in subdirectories of `/usr/ldp/device/var`. Log pruning occurs when a disk partition reaches 90% capacity.

**Table 91: IDP Local Log Directories**

Directory	Content
<code>/usr/ldp/device/var/logs</code>	Local storage for device and security event logs before they are forwarded to NSM.
<code>/usr/ldp/device/var/pktlogs</code>	Local storage for packet capture logs before they are forwarded to NSM.
<code>/usr/ldp/device/var/profile</code>	Local storage for Profiler database logs before they are forwarded to NSM.
<code>/usr/ldp/device/var/sysinfo/logs</code>	Location where system messages are written.
<code>/usr/ldp/device/var/stat/</code>	Local storage for application volume tracking logs before they are forwarded to NSM, IDP Reporter, or Application Usage Manager.





**NOTE:** Although `/usr/idp/device/var` is a symbolic link to `/var/idp/device/var`, user scripts or programs created to manage files should reference the `/usr/idp/device/var` path.

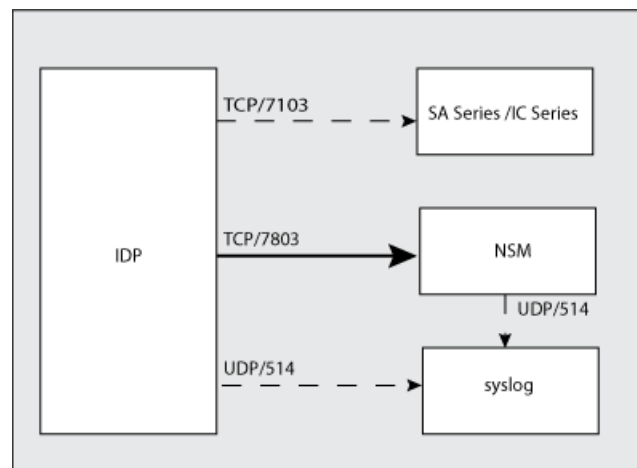
By default, logs are forwarded to NSM, which is the primary user interface for the IDP Series device.

Optionally, you can configure the IDP Series device to send copies of logs to external devices, such as:

- A syslog server, including a Juniper Networks Security Threat Response Manager (STRM) device, which reads the IDP syslog format.
- A Juniper Networks Secure Access Series or Infranet Controller Series device to inform access policies.

Figure 126 on page 333 provides a visual summary of your log forwarding options. The solid line indicates default behavior. The dashed lines indicate options you must configure to use.

**Figure 126: IDP Log Storage and Log Forwarding**



**NOTE:** In IDP OS Release 5.1, syslog protocol port are configurable. However, we recommend you use the standard protocol and port whenever feasible.

## NSM Log Collection

By default, the IDP Series device sends logs to NSM where they can be displayed and analyzed with the NSM user interface. We recommend you become familiar with an use NSM as a central location for log analysis. Logs are stored on the NSM Device Server in subdirectories of `/usr/netscreen/DevSvr/var/logs`. NSM supports the following log management features:

- Command-line utilities to archive, copy, and purge logs.
- Configurable time retention policies that trigger pruning.
- Automated log management jobs based on criteria you configure, including severity, category, and so forth.
- Support for log field filters in export operations to XML, CSV, syslog, SNMP, e-mail, or script.

For complete information on NSM log management features, see Chapter 19 of the [NSM Administration Guide](#).

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)
- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)

## CHAPTER 32

# Managing Security Policies

- [Managing Security Policies Task Summary on page 335](#)
- [Assigning a Security Policy to a Device \(NSM Procedure\) on page 335](#)
- [Validating a Security Policy \(NSM Procedure\) on page 336](#)
- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)
- [Pushing Security Policy Updates to an IDP Series Device \(NSM Procedure\) on page 340](#)
- [Disabling Rules \(NSM Procedure\) on page 342](#)
- [Exporting Security Policies \(NSM Procedure\) on page 342](#)

## Managing Security Policies Task Summary

---

Routine administration can include the following tasks related to managing IDP security policies:

- Assigning a security policy to a device if you have not done so already
- Validating a security policy
- Pushing a security policy update
- Exporting a security policy to a file
- Disabling rules that trigger false positives or degrade performance

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Assigning a Security Policy to a Device \(NSM Procedure\) on page 335](#)
- [Validating a Security Policy \(NSM Procedure\) on page 336](#)
- [Pushing Security Policy Updates to an IDP Series Device \(NSM Procedure\) on page 340](#)
- [Exporting Security Policies \(NSM Procedure\) on page 342](#)
- [Disabling Rules \(NSM Procedure\) on page 342](#)

## Assigning a Security Policy to a Device (NSM Procedure)

---

When you create a security policy with the new security policy wizard, you can designate an IDP Series device target. You can also specify IDP Series devices as targets for particular

rules. Follow this procedure if you did not complete the assignment when you created the security policy, or if you want to change the assignment.

To assign a security policy to a device:

1. In the NSM navigation tree, select **Policy Manager > Security Policies**.
2. Right-click the security policy you want to assign and select **Assign Policy** to display the Assign Policies to Devices dialog box, where you can select IDP Series devices to which the policy should be assigned.
3. Click **OK**.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Using the New Policy Wizard \(NSM Procedure\) on page 198](#)
- [Specifying Rule Targets \(NSM Procedure\) on page 223](#)
- [Managing Security Policies Task Summary on page 335](#)

---

## Validating a Security Policy (NSM Procedure)

We recommend you validate the integrity of a security policy before pushing the security policy to a device.

To validate a security policy:

1. From the NSM main menu, select **Devices > Policy > Validate IDP Policy** to display the Validate IDP Policy dialog box.
2. Select IDP Series devices to which the validation job applies.
3. Click **OK**.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Troubleshooting Security Policy Validation Errors \(NSM Procedure\) on page 594](#)
- [Managing Security Policies Task Summary on page 335](#)

---

## Loading J-Security Center Updates (NSM Procedure)

The Juniper Networks Security Center (J-Security Center) routinely makes important updates available to IDP security policy components:

- **Detector engine.** The IDP detector engine is a dynamic protocol decoder that includes support for decoding more than 60 protocols and more than 500 service contexts. You should update IDP detector engine when you first install IDP, whenever you upgrade, and whenever alerted to do so by Juniper Networks. You can view release notes for detector engine updates at <http://www.juniper.net/techpubs/software/management/idp/de/>.

- Attack database. The [attack signature database](#) stores data definitions for attack objects. Attack objects are patterns comprising stateful signatures and traffic anomalies. You specify attack objects in IDP rulebase rules.
- Application signature database. The [application signature database](#) stores data definitions for application objects. Application objects are patterns used to identify applications and match APE rulebase rules.

J-Security Center updates are packaged and released separately from the IDP operating system and software code base to ensure IDP products protect your network against recently discovered vulnerabilities. We recommend you schedule automatic updates for the attack database and application database. For IDP Series devices, both databases are distributed in “signature database updates”.

After you have completed the update, any new attack objects and application objects are available in the security policy editor. If you use dynamic groups in IDP rulebase rules and a new attack object belongs to the dynamic group, the rule automatically inherits the new attacks.



**NOTE:** We recommend you subscribe to the IDP Signature Updates technical bulletin to be notified when J-Security Center releases IDP detector engine updates. Go to <https://www.juniper.net/alerts/>.

[Table 92 on page 337](#) provides procedures for updating the IDP detector engine and the NSM attack database.

**Table 92: IDP Detector Engine and NSM Attack Database Update Procedures**

Task	Procedure
To view version information for the installed IDP detector engine	In the NSM Device Manager, double-click the IDP Series device to display the IDP Series device configuration editor. The Info node displays version information, including the IDP detector engine version.

Table 92: IDP Detector Engine and NSM Attack Database Update Procedures (*continued*)

Task	Procedure
To update the IDP detector engine	<p>Updating the IDP detector engine is a three part process.</p> <p>To update IDP detector engine:</p> <ol style="list-style-type: none"> <li>Download IDP detector engine and NSM attack database updates to the NSM GUI server: In NSM, select <b>Tools &gt; View/Update NSM attack database</b> and complete the wizard steps.</li> <li>Push the updated IDP detector engine to IDP Series devices: In NSM, select <b>Devices &gt; IDP Detector Engine &gt; Load IDP Detector Engine</b> and complete the wizard steps.</li> </ol> <p><b>NOTE:</b> Updating the IDP detector engine on a device does not require a reboot of the device.</p> <ol style="list-style-type: none"> <li>Run a security policy update job to initialize the IDP detector engine update: <ol style="list-style-type: none"> <li>In NSM, select <b>Devices &gt; Configuration &gt; Update Device Config</b>.</li> <li>Select devices to which to push the updates and set update job options.</li> <li>Click <b>OK</b>.</li> </ol> </li> </ol>
To update predefined attack objects and application objects	<p>Updating attack objects is a two-part process.</p> <p>To update predefined attack objects:</p> <ol style="list-style-type: none"> <li>Download NSM attack database updates to the NSM GUI server: From the NSM main menu, select <b>Tools &gt; View/Update NSM attack database</b> and complete the wizard steps.</li> <li>Push the updates to IDP Series devices: <ol style="list-style-type: none"> <li>From the NSM main menu, select <b>Devices &gt; Configuration &gt; Update Device Config</b>.</li> <li>Select devices to receive pushed updates and set update job options.</li> <li>Click <b>OK</b>.</li> </ol> </li> </ol> <p><b>NOTE:</b> Only the attack objects that are used in IDP rules for the device are pushed from the GUI server to the device.</p>

Table 92: IDP Detector Engine and NSM Attack Database Update Procedures (*continued*)

Task	Procedure
To schedule regular updates	<ol style="list-style-type: none"> <li>1. Log in to the NSM GUI server command line.</li> <li>2. Change directory to <code>/usr/netscreen/GuiSvr/utls</code>.</li> <li>3. Create a shell script called <b>attackupdates.sh</b> with the following contents: <ul style="list-style-type: none"> <li>• Set the NSMUSER environment variable with an NSM domain/user pair. The command for setting environment variables depends on your OS. For example: <pre>export NSMUSER=domain/user</pre> </li> <li>• Set the NSMPASSWD environment variable with an NSM password. The command for setting environment variables depends on your OS and shell. For example: <pre>export NSMPASSWD=password</pre> </li> <li>• Specify a <b>guiSvrCli.sh</b> command string. For example: <pre>/usr/netscreen/GuiSvr/utls/guiSvrCli.sh --update-attacks --post-action --update-devices --skip</pre> </li> </ul> </li> <li>4. Make the script executable by the user associated with the cron job: <pre>chmod 700 attackupdates.sh</pre> </li> <li>5. Run the crontab editor: <pre>crontab -e</pre> </li> <li>6. Add an entry for the shell script: <pre>minutes_after_hour hour * * * /usr/netscreen/GuiSvr/utls/attackupdates.sh</pre> </li> </ol> <p>During the update, the <b>guiSvrCli</b> utility updates the attack object database, then performs the post actions. After updating and executing actions, the system generates an exit status code of 0 (no errors) or 1 (errors).</p> <p><b>NOTE:</b> For information on connecting to the NSM command line, see the NSM documentation.</p>

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)
- [Application Objects Task Summary on page 286](#)
- [Pushing Security Policy Updates to an IDP Series Device \(NSM Procedure\) on page 340](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)
- [Using Application Objects on page 73](#)

## Pushing Security Policy Updates to an IDP Series Device (NSM Procedure)

You must run a device configuration update job (also called *pushing* an update) in the following cases:

- After you have revised the security policy assigned to an IDP Series device. The configuration changes you make in NSM do not affect the IDP Series device until you have successfully pushed the configuration to the IDP Series device.
- If you have deleted the device from NSM and subsequently re-add it. In these cases, the IDP Series device does not retain the previous security policy assignment.
- If you use the NSM Device Manager to change IDP Series device settings.

To push configuration updates to multiple IDP Series devices:

1. From the NSM main menu, select **Devices > Configuration > Update Device Config** to display the Update Devices dialog box.
2. Select the devices to receive the pushed configuration updates.
3. Set update job options as described in [Table 93 on page 340](#).
4. Click **OK**.

**Table 93: Devices Update Job Options**

Tab	Description
General	<b>Run Summarize Delta Config</b> —Displays a summary of the delta config. The delta config is the difference between the device running configuration and the NSM configuration object.
Netconf	<b>Lock configuration during update</b> —Not applicable.
	<b>Update to candidate config first before commit to running config</b> —Not applicable.
	<b>Use confirmed commit</b> —Not applicable.
	<b>Rollback candidate config to running config in error</b> —Not applicable.
	<b>Discard uncommitted changes when exclusive lock is available</b> —Not applicable.



Table 93: Devices Update Job Options (*continued*)

Tab	Description
ScreenOS and IDP	<b>Show unconnected devices</b> —Displays devices that are not connected to NSM in the Update Devices dialog box
	<b>Update when device connects</b> —Attempts to update a previously unconnected device with pending changes stored in NSM.
	<b>Firewall Device Options</b> —Not applicable.
	<b>Standalone IDP Series device options</b> —includes the following option: <ul style="list-style-type: none"> <li>• <b>Restart IDP Profiler after Device Update</b>—Restarts the Profiler after the update.</li> </ul>
	<b>ISG Device Options</b> —Not applicable.

To push an update to a specific, single device:

1. In Device Manager, right-click the device to receive the pushed configuration update select **Update Device** to display the Update Device dialog box.
2. Set update job options. [Table 94 on page 341](#) describes these update job options.
3. Click **OK**.

Table 94: Device Update Job Options

Option	Description
Update When Device Connects	Attempts to update a previously unconnected device with pending changes stored in NSM.
Restart IDP Profiler After Device Update	Restarts the Profiler after the update.
Update IDP Rulebase Only	Updates only the IDP rulebase, Exempt rulebase, and Backdoor rulebase. Select this option if you are updating only the rulebases or attack objects.
Don't Show This Dialog	Does not display this dialog box in the future.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Troubleshooting Configuration Push Errors \(NSM Procedure\) on page 593](#)
- [Managing Security Policies Task Summary on page 335](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Understanding the Number of Available and Installed Policies on page 43](#)

## Disabling Rules (NSM Procedure)

---

You can disable a rule without deleting it in cases where you run tuning experiments, troubleshoot an issue, or otherwise need to make a quick or temporary modification.

To disable a rule, right-click the rule number and select **Disable**.



**NOTE:** You cannot disable an entire security policy or a rulebase.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Managing Security Policies Task Summary on page 335](#)

## Exporting Security Policies (NSM Procedure)

---

You can export a security policy rulebase to an HTML file.

To export a security policy to an HTML file:

1. From the NSM main menu, select **File > Export Policy** to display the Export Policy dialog box.
2. Select the rulebases you want to export.
3. Select a directory in which to save the exported file.
4. Click **Export** to complete the operation.

Each export creates a new directory. The default directory name is *policyname\_YYMMDD\_HHMMSS*. The export process puts each rulebase in a separate HTML file in that directory.

Use an HTML browser to view the exported file.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Managing Security Policies Task Summary on page 335](#)

## CHAPTER 33

# Managing the IDP Device Configuration with NSM

- [NSM Device Configuration Management Task Summary on page 343](#)
- [Adding IDP Series Devices to NSM Device Manager on page 344](#)
- [Activating Devices \(NSM Procedure\) on page 348](#)
- [Pulling or Pushing Configuration Updates on page 350](#)
- [Modifying the IDP Series Device Configuration on page 351](#)
- [Deleting an IDP Series Device Configuration from NSM Device Manager \(NSM Procedure\) on page 382](#)

### NSM Device Configuration Management Task Summary

---

Juniper Networks Network and Security Manager (NSM) is a central management server capable of managing hundreds of IDP Series devices and other Juniper Networks devices, such as Juniper Networks ScreenOS firewalls, Juniper Networks SA Series SSL VPN appliances, and Juniper Networks IC Series Unified Access Control appliances. You typically deploy NSM in a management subnet accessible to NSM-managed devices.

You use NSM to perform the following tasks related to IDP Series device configuration management:

- Adding the IDP Series device to the NSM Device Manager
- Pushing for pulling configuration updates
- Modifying the IDP Series device configuration

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Adding IDP Series Devices to NSM Device Manager on page 344](#)
- [Activating Devices \(NSM Procedure\) on page 348](#)
- [Pulling or Pushing Configuration Updates on page 350](#)
- [Modifying the IDP Series Device Configuration on page 351](#)
- [Deleting an IDP Series Device Configuration from NSM Device Manager \(NSM Procedure\) on page 382](#)

## Adding IDP Series Devices to NSM Device Manager

---

Before you can use Network and Security Manager (NSM) to manage an IDP Series device, you must add the IDP Series device to NSM Device Manager. Use one of the following workflows to add the IDP Series device to the NSM Device Manager:

- [Adding a Reachable Device on page 344](#)
- [Adding an Unreachable Device on page 345](#)
- [Modeling an IDP Series Device Configuration on page 346](#)
- [Adding Device Clusters on page 347](#)

### Adding a Reachable Device

A reachable device is a device you have installed and initialized, including configuring an IP address for the management interface and connecting the management interface to the network. You complete the reachable device workflow in cases where you set up the IDP Series device first and the NSM device object second.

To import an IDP Series device with a known IP address:

1. In the NSM navigation tree, select **Device Manager > Devices**.
2. Click the + icon and select **Device** to display the Add Device wizard. Configure the following properties:
  - **Name**—Specify a string to identify the IDP Series device. The string may contain letters, numbers, spaces, dashes, and underscores.
  - **Color**—Select a color. Some administrators use colors to distinguish devices by type, region, software version, and so forth.
  - Select **Device Is Reachable** (default).
3. Click **Next**.
4. In the Specify Connection Settings dialog box, enter the following connection information:
  - Enter the IP address of the IDP Series device.
  - Enter the username of the device admin user.
  - Enter the password for the device admin user.
  - Enter the password for the device root user.



**NOTE:** In NSM, passwords are case-sensitive.

---

- Select **SSH Version 2** and port 22.

Click **Next**.

5. On the Verify Device Authenticity page, use an out-of-band method to verify the RSA key fingerprint information to prevent man-in-the-middle attacks.

Click **Next**.

In response, NSM connects to the IDP Series device to retrieve device information. This process takes a moment.

6. Verify that the device type, OS version, device serial number, and device mode are correct.
7. Click **Next** to add the device to NSM.
8. Click **Next** to import the configuration from the IDP Series device.
9. Click **Finish**.

For IDP OS Release 4.1 and later devices, NSM next runs a job to update the IDP Series device with the Recommended IDP security policy. The Job Information dialog box shows the status of the Update Device job.

10. After the job is complete, double-click the device in Device Manager to view the imported configuration.

To check the device configuration status, mouse over the device and verify that the device status displays **Managed**.

## Adding an Unreachable Device

An unreachable device is a device that has not been set up and so does not have an IP address for the management interface. You complete the unreachable device workflow in cases where you set up the NSM device object first and the IDP Series device second.

To add an IDP Series device with an unknown IP address:

1. In the NSM navigation tree, select **Device Manager > Devices**.
2. Click the + icon and select **Device** to display the Add Device wizard.
3. Configure the following properties:
  - **Name**—Specify a string to identify the IDP Series device. The string may contain letters, numbers, spaces, dashes, and underscores.
  - **Color**—Select a color. Some administrators use colors to distinguish devices by type, region, software version, and so forth.
  - Select **Device Is Not Reachable**.
4. Click **Next**.
5. On the Specify One Time Password page:
  - Make a note of the unique external ID for the device. The device administrator will need it to connect the device to NSM. This ID number represents the device within the management system. The wizard automatically provides this value.
  - Specify the first connection one-time password (OTP) that authenticates the device.

- Click **Show Device Commands** to display the list of CLI commands that must be executed on the device to connect to NSM. The commands enable management, set the IP address for NSM, set the unique external ID, and set the device OTP.

Copy these commands to a text file.

Click **Finish** to complete the Add Device wizard and include the new device in the Device Manager list.

6. Log into the CLI as **admin** and enter **su -** to switch to **root**.
7. Run the CLI commands you copied in Step 5.
8. In the NSM Device Manager, mouse over the device to track its configuration status. The first status message is **Waiting for 1st connect**. After the connection has been established, the status displays **Import Needed**.
9. Right-click the device and select **Import Device**.

The Job Information box displays the job type and status for the import; when the job status displays successful completion, click **Close**.

For IDP OS Release 4.1 and later devices, NSM next runs a job to update the IDP Series device with the Recommended IDP security policy. The Job Information dialog box shows the status of the Update Device job.

10. After the job is complete, double-click the device in Device Manager to view the imported configuration.

To check the device configuration status, mouse over the device and verify that the device status displays **Managed**.

## Modeling an IDP Series Device Configuration

You model an IDP Series device configuration when the device is not online, and you intend to push the configuration to the device when it is ready to be put online and configured.

To model an IDP Series device

1. In the NSM navigation tree, select **Device Manager > Devices**.
2. Click the + icon and then select **Device** to display the Add Device wizard.
3. Configure the following properties:
  - **Name**—Specify a string to identify the IDP Series device. The string may contain letters, numbers, spaces, dashes, and underscores.
  - **Color**—Select a color. Some administrators use colors to distinguish devices by type, region, software version, and so forth.
  - Select **Model Device**.

4. In the Specify OS Name, Version, and Platform page, enter the following connection information:
  - In the OS Name list, select the device family that the modeled device belongs to.
  - In the platform list, select the device platform name.
  - In the OS version list, select the version of the operating system or firmware that runs on the device.
5. Click **Finish**.
6. Double-click the device to display the device configuration editor.
7. When you have completed the model configuration, check the device configuration status. Mouse over the device and verify that the device status displays **Modeled**.

## Adding Device Clusters

In a high-availability (HA) deployment, an IDP Series device cluster is a set of two IDP Series devices deployed for the same purpose—to provide intrusion detection and prevention for a particular network segment. You use Appliance Configuration Manager (ACM) to configure HA. You use Network and Security Manager (NSM) to create a cluster object that will help you ensure the nodes (IDP Series devices) maintain the same feature configuration, which is a requirement of HA deployments.

To configure clusters in NSM:

1. Add the cluster object.
2. Add cluster members to the cluster object.

To add a cluster object:

1. In the NSM navigation tree, select **Device Manager > Devices**.
2. Click the + icon and then select **Cluster** to display the New Cluster wizard.
3. Configure the following properties:
  - **Cluster Name**—Specify a string to identify the IDP Series device. The string may contain letters, numbers, spaces, dashes, and underscores.
  - **Color**—Select a color. Some administrators use colors to distinguish devices by type, region, software version, and so forth.
  - In the OS Name list, select **ScreenOS/IDP**.
  - In the platform list, select the device model number.
  - In the Managed OS version list, select the IDP OS version.
4. Click **OK**.

To add cluster members:

1. In the NSM navigation tree, select **Device Manager > Devices**.
2. Right-click the cluster object and then select **New > Cluster Member** to display the Add Cluster Member wizard.
3. Complete the wizard steps.
4. Repeat to add the second cluster member.



**NOTE:** An IDP Series cluster contains exactly two members.

#### **Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [NSM Device Configuration Management Task Summary on page 343](#)

---

## **Activating Devices (NSM Procedure)**

This section includes the following topics:

- [Activating a Reachable IDP Series Device on page 348](#)
- [Activating an Unreachable IDP Series Device on page 349](#)

### **Activating a Reachable IDP Series Device**

To activate a device:

1. In the NSM Device Manager, right-click the device and select **Activate Device** to display the Activate Device wizard.
2. Select **Device deployed and IP is reachable**.
3. In the Specify Connection Settings dialog box, enter the following connection information:
  - Enter the IP address of the IDP Series device.
  - Enter the username of the device admin user.
  - Enter the password for the device admin user.
  - Enter the password for the device root user.



**NOTE:** In NSM, passwords are case-sensitive.

- Select **SSH Version 2** as the connection method and port 22.

Click **Next**.



4. On the Verify Device Authenticity page, use an out-of-band method to verify the RSA key fingerprint information to prevent man-in-the-middle attacks. For example:
  - a. Log into the CLI as **admin** and enter **su -** to switch to **root**.
  - b. Enter **cd /etc/ssh**.
  - c. Enter **ssh-keygen -l -f ssh\_host\_dsa\_key**.
  - d. After you have verified the key, click **Next**.

In response, NSM connects to the IDP Series device to retrieve device information. This process takes a moment.

5. Verify that the device type, OS version, device serial number, and device mode are correct.
6. Click **Next** to add the device to NSM.
7. Click **Next** to import the configuration from the IDP Series device.
8. Click **Finish**.

For IDP OS Release 4.1 and later devices, NSM next runs a job to update the device with the Recommended IDP security policy. The Job Information dialog box shows the status of the Update Device job.

9. After the job is complete, double-click the device in Device Manager to view the imported configuration.

To check the device configuration status, mouse over the device and verify that the device status displays **Managed**.

## Activating an Unreachable IDP Series Device

To activate an unreachable device:

1. In the NSM Device Manager, right-click the device and select **Activate Device** to display the Activate Device wizard.
2. Select **Device deployed, but IP is not reachable**.
3. On the Specify One Time Password page:
  - Make a note of the unique external ID for the device. The device administrator will need it to connect the device to NSM. This ID number represents the device within the management system. The wizard automatically provides this value.
  - Specify the first connection one-time password (OTP) that authenticates the device.
  - Click **Show Device Commands** to display the list of CLI commands that must be executed on the device to connect to NSM. The commands enable management, set the unique external ID, set the management IP address to the device server IP address, and set the device OTP.

Copy these commands to a text file.

Click **Finish** to complete the Add Device wizard and include the new device in the Device Manager list.

4. Log into the CLI as **admin** and enter **su -** to switch to **root**.
5. Run the CLI commands you copied in Step 3.
6. In the NSM Device Manager, mouse over the device to track its configuration status. The first status message is **Waiting for 1st connect**. After the connection has been established, the status displays **Import Needed**.
7. Right-click the device and select **Import Device**.

The Job Information box displays the job type and status for the import; when the job status displays successful completion, click **Close**.

For IDP OS Release 4.1 and later devices, NSM next runs a job to update the device with the Recommended IDP security policy. The Job Information dialog box shows the status of the Update Device job.

8. After the job is complete, double-click the device in Device Manager to view the imported configuration.

To check the device configuration status, mouse over the device and verify that the device status displays **Managed**.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [NSM Device Configuration Management Task Summary on page 343](#)

## Pulling or Pushing Configuration Updates

The IDP Series device configuration stored in NSM is not an active configuration. The active configuration is the one running on the IDP Series device. In some cases, the two configurations can be out of sync. To synchronize them, you can pull the running configuration from the IDP Series device into NSM or push the NSM device configuration onto the IDP Series device.

[Table 95 on page 350](#) describes the cases when you pull and the cases where you push.

**Table 95: Pulling and Pushing Configuration Updates**

Pull a Configuration	Push a Configuration
<ul style="list-style-type: none"> <li>• When you add an IDP Series device to NSM.</li> <li>• After you use ACM to change the deployment mode.</li> </ul>	<ul style="list-style-type: none"> <li>• After you model an IDP Series device in NSM.</li> <li>• After you update the IDP detector engine, NSM attack database, or security policy.</li> <li>• After you use the NSM device editor to enable features or change feature settings.</li> </ul>

To import the IDP Series device configuration:

1. In the NSM Device Manager, right-click the device and select **Import Device**.

2. Select the **Run Delta Config** check box.

A delta configuration summary displays the differences between the current IDP Series device configuration and the running configuration (on the IDP Series device).

3. Click **OK**.

To push an update from the NSM Device Manager to an IDP Series device:

1. In the NSM Device Manager, right-click the device you want to push the update to and select **Update Device** to display the Update Device dialog box.
2. Set update job options as described in [Table 96 on page 351](#).
3. Click **OK**.

**Table 96: Device Update Job Options**

Option	Description
Update When Device Connects	If the IDP Series device is not currently connected to NSM, then NSM queues the update so that the update happens when the IDP Series device reconnects to NSM.
Restart IDP Profiler After Device Update	Restarts the Profiler after the update.
Update IDP Rulebase Only	Updates only the IDP rulebase. Select this option if you are updating only the IDP rulebase or attack objects.
Don't Show This Dialog	Does not display this dialog box in the future.

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [NSM Device Configuration Management Task Summary on page 343](#)

## Modifying the IDP Series Device Configuration

You do not need to modify the IDP Series device configuration to get started with your deployment. The default settings are appropriate for most deployments. As you learn how the IDP Series device handles your network traffic, you can use Network and Security Manager (NSM) to modify the IDP Series device properties described in this section to optimize performance and reduce false positives.

This section includes the following topics:

- [Modifying NSM Informational Properties on page 352](#)
- [Modifying Antispoof Settings on page 353](#)
- [Modifying Runtime Parameters on page 355](#)
- [Modifying Load-Time Parameters on page 365](#)
- [Modifying Protocol Anomaly Thresholds on page 367](#)

## Modifying NSM Informational Properties

NSM informational properties are management object parameters you created when you added the device to the NSM Device Manager, as well as inventory data, including the installed software and firmware versions. [Figure 127 on page 352](#) shows the Info page, where you can modify these properties.

Figure 127: NSM Device Configuration Editor: Info Page

The screenshot shows the 'idp-75 - Device' configuration window. On the left is a tree view with 'Info' selected. The main area displays the following settings:

- Name:** idp-75
- Color:** green
- Platform:** NS-IDP-75
- Managed OS Version:** IDP5.0
- Running OS Version:** IDP5.0.124008
- IP Address:** 10.65.23.213
- Serial Number:** 0230012008000007
- IDP Detector Version:** 5.0.110090213
- IDP Mode:** sniff
- Secondary Management Server IP:** . . .
- License:**
  - Software License Type:** Evaluation
  - Software License Expiration date:** 4/25/2009
- Version:**
- Policy for Device:**
  - Security Policy Name:** Recommended\_Policy

Buttons at the bottom right: OK, Cancel, Apply.

To modify NSM informational properties:

1. In NSM Device Manager, double-click the IDP Series device you want to modify to display the device configuration editor, which opens by default to the Info page.
2. Configure the informational settings described in [Table 97 on page 352](#).
3. Click **Apply**.
4. Click **OK**.

Table 97: IDP Series Device Configuration: Info Settings

Setting	Description
Name	The name of the IDP Series device in NSM. Editable.

Table 97: IDP Series Device Configuration: Info Settings (*continued*)

Setting	Description
Color	The color of the IDP Series device icon in NSM. Selectable.
Platform	The IDP Series device hardware model number.
Managed OS Version	The major OS version.
Running OS Version	The precise OS version installed on the device.
IP Address	The IDP Series device management port IP address. <b>NOTE:</b> Can only be changed with ACM.
Serial Number	The product serial number.
IDP Detector Version	The version of the IDP detector engine installed on the device.
IDP Mode	Deployment mode: sniffer, transparent, mixed. <b>NOTE:</b> Can only be changed with ACM.
Secondary Management Server IP	The IP address that the IDP Series device contacts if it cannot reach the current NSM server.
Software License Type	The type of license currently loaded on the IDP Series device. An evaluation license is good for one year.
Software License Expiration Date	The expiration date of the license currently loaded on the IDP Series device.
Security Police Name	The security policy assigned to the device. Selectable.

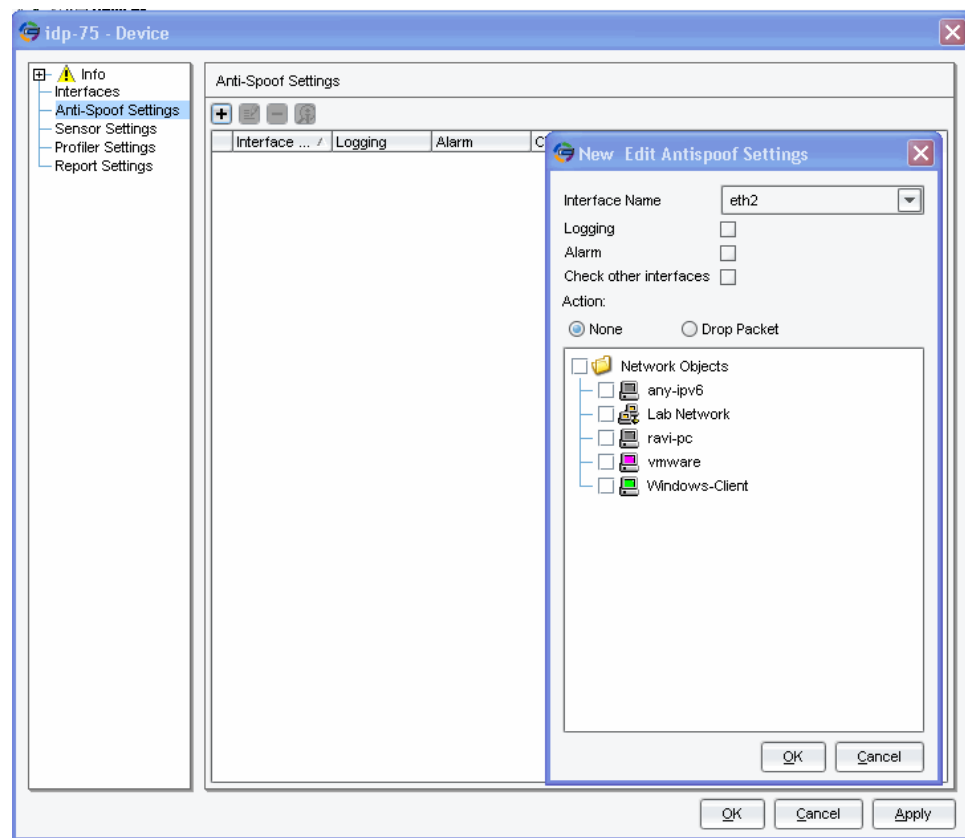
## Modifying Antispoof Settings

You detect attacks that attempt to spoof the addresses of hosts in your protected network by associating IDP Series traffic interfaces with the addresses of hosts in your protected network. The IDP Series appliance then detects an IP spoof attack if:

- An incoming packet uses an IP address that belongs to a network object on your internal network.
- An outgoing packet uses an IP address that does not belong to a network object on your internal network.

[Figure 128 on page 354](#) shows the Anti-Spoof Settings page, where you can configure IP spoof detection.

Figure 128: NSM Device Configuration Editor: Anti-Spoof Settings Page



To modify antispoof settings:

1. In NSM Device Manager, double-click the IDP Series device you want to modify to display the device configuration editor.
2. Click **Anti-Spoof Settings**.
3. Click the + icon to display the Anti-Spoof Settings dialog box.
4. Configure the antispoof settings described in [Table 98 on page 354](#).
5. Click **Apply**.
6. Click **OK**.

Table 98: IDP Series Device Configuration: Antispoof Settings

Setting	Function
Interface Name	Selects a forwarding interface to configure.
Logging	Enables logging for spoofed IP addresses.
Alarm	Enables alerts for spoofed IP addresses.

Table 98: IDP Series Device Configuration: Antispoof Settings (*continued*)

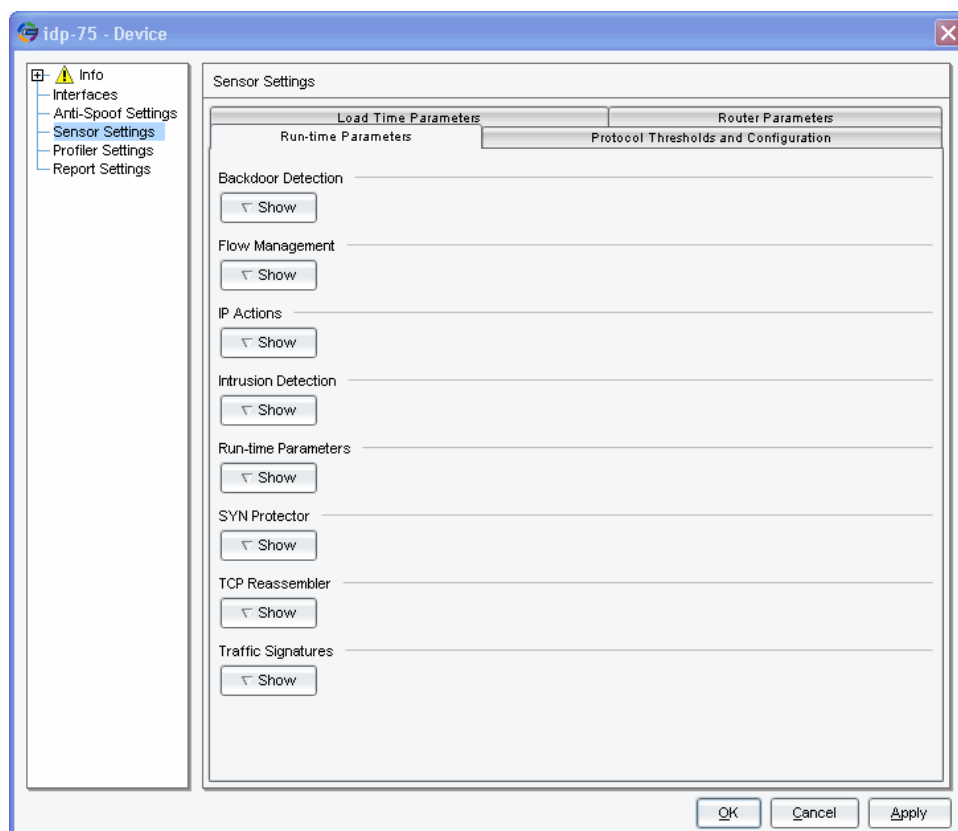
Setting	Function
Check Other Interfaces	Indicates whether the device should check the status of other interfaces when determining spoofing.
Action	Specifies the action for the IDP Series device to take: <b>None</b> or <b>Drop Packet</b> .
Network Objects	Specifies the address objects you associate with the selected interface.

## Modifying Runtime Parameters

Runtime parameters include options for tuning IDP Series detection methods. In general, you modify these settings only if you encounter false positives or performance issues.

Figure 129 on page 355 shows the Run-time Parameters tab, where you can configure these settings.

Figure 129: NSM Device Configuration Editor: Run-time Parameters Tab



To modify runtime parameters:

1. In NSM Device Manager, double-click the IDP Series device you want to modify to display the device configuration editor.
2. Click **Sensor Settings**.

3. Click the **Run-time Parameters** tab.
4. Modify the runtime settings described in [Table 99 on page 356](#).
5. Click **Apply**.
6. Click **OK**.

**Table 99: IDP Series Device Configuration: Runtime Parameters**

Setting	Description
Backdoor Detection	<p><b>Minimum interval between consecutive small packets / Maximum interval between consecutive small packets</b>—Controls the minimum and maximum intervals (in microseconds) between the arrival of two consecutive small packets in suspected interactive traffic. If the IDP engine sees small packets arrive in less than the minimum or more than the maximum number of microseconds, it does not consider the traffic to be interactive.</p> <p>The defaults are 20,000 and 20,000,000. This means that consecutive small packets must arrive within 20,000 to 20,000,000 microseconds to be considered interactive.</p> <hr/> <p><b>Byte threshold for packet sizes in a backdoor connection</b>—Controls the maximum number of bytes a TCP packet must contain before the IDP engine uses the packet for backdoor detection heuristics. The default is 20 bytes.</p> <hr/> <p><b>Minimum number of data carrying TCP packets</b>—Controls the minimum number of data-carrying TCP packets in suspected interactive traffic. The default is 20 packets.</p> <hr/> <p><b>Minimum percentage of back-to-back small packets</b>—Controls the minimum percentage of consecutive small packets in suspected interactive traffic. If the IDP engine sees less than this percentage, it does not report a backdoor event. The default is 20%.</p> <hr/> <p><b>Ratio of small packets to the total packets</b>—Controls the minimum percentage of small packets that the IDP engine uses for backdoor detection heuristics. If the IDP engine sees less than this minimum, it does not report a backdoor event. The default is 20%.</p>



Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description																																		
Flow Management	<b>Timeout for non-UDP/TCP/ICMP flows</b> —Controls idle flow. Each connection through the security module typically has two flows, one in each direction. If the IDP engine does not see flow activity for the specified timeout, it removes the idle flow from the flow table. The default is 30 seconds.																																		
	<b>Timeout for UDP flows</b> —Controls idle flow. Each connection through the security module typically has two flows, one in each direction. If the IDP engine does not see flow activity for the specified timeout, it removes the idle flow from the flow table. The default is 30 seconds.																																		
	<b>Timeout for TCP flows</b> —Controls idle flow. Each connection through the security module typically has two flows, one in each direction. If the IDP engine does not see flow activity for the specified timeout, it removes the idle flow from the flow table. The default is 30 seconds.																																		
	<b>Timeout for ICMP flows</b> —Controls idle flow. Each connection through the security module typically has two flows, one in each direction. If the IDP engine does not see flow activity for the specified timeout, it removes the idle flow from the flow table. The default is 30 seconds.																																		
	<b>Maximum TCP Sessions</b> —Controls the maximum number of TCP sessions. If the IDP system reaches the maximum, it drops all new sessions and writes a SESSION_LIMIT_EXCEEDED log. Defaults vary according to model, as shown in the following table.																																		
<table><tr><th>Model</th><th>Default</th><th>Minimum</th><th>Maximum</th></tr><tr><td>IDP75</td><td>40,000</td><td>0</td><td>100,000</td></tr><tr><td>IDP250</td><td>125,000</td><td>0</td><td>300,000</td></tr><tr><td>IDP800</td><td>400,000</td><td>0</td><td>1,000,000</td></tr><tr><td>IDP8200<sup>2</sup></td><td>400,000</td><td>0</td><td>500,000</td></tr><tr><td>IDP200</td><td>50,000</td><td>0</td><td>70,000</td></tr><tr><td>IDP600</td><td>100,000</td><td>0</td><td>220,000</td></tr><tr><td>IDP1100</td><td>200,000</td><td>0</td><td>500,000</td></tr></table>				Model	Default	Minimum	Maximum	IDP75	40,000	0	100,000	IDP250	125,000	0	300,000	IDP800	400,000	0	1,000,000	IDP8200 <sup>2</sup>	400,000	0	500,000	IDP200	50,000	0	70,000	IDP600	100,000	0	220,000	IDP1100	200,000	0	500,000
Model	Default	Minimum	Maximum																																
IDP75	40,000	0	100,000																																
IDP250	125,000	0	300,000																																
IDP800	400,000	0	1,000,000																																
IDP8200 <sup>2</sup>	400,000	0	500,000																																
IDP200	50,000	0	70,000																																
IDP600	100,000	0	220,000																																
IDP1100	200,000	0	500,000																																
<p><sup>1</sup>For all entries except IDP8200, the maximum session limit shown is also the device session limit. We recommend that the sum of “max sessions” you configure for TCP, UDP, ICMP, and Other not exceed the device session limit. The user interface does not enforce this, so do a quick calculation whenever you change these settings. If you increase TCP, decrease UDP in proportion. Otherwise, you might encounter a traffic load that leads to undesirable results, such as drops for all traffic when the device limit is reached.</p> <p><sup>2</sup>For IDP8200, the limits shown are configured for each core IDP engine. There are six core IDP engines. The sum of “max sessions” you configure for IDP8200 can be 1,000,000 per core IDP engine. The IDP8200 device session limit rating is 6,000,000.</p>																																			

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
---------	-------------

**Maximum UDP Sessions**—Controls the maximum number of UDP sessions. If the IDP system reaches the maximum, it drops all new sessions and writes a SESSION\_LIMIT\_EXCEEDED log.

Model	Default	Minimum	Maximum <sup>1</sup>
IDP75	40,000	0	100,000
IDP250	125,000	0	300,000
IDP800	400,000	0	1,000,000
IDP8200 <sup>2</sup>	400,000	0	500,000
IDP200	10,000	0	70,000
IDP600	100,000	0	220,000
IDP1100	200,000	0	500,000

<sup>1</sup>For all entries except IDP8200, the maximum session limit shown is also the device session limit. We recommend that the sum of “max sessions” you configure for TCP, UDP, ICMP, and Other not exceed the device session limit. The user interface does not enforce this, so do a quick calculation whenever you change these settings. If you increase TCP, decrease UDP in proportion. Otherwise, you might encounter a traffic load that leads to undesirable results, such as drops for all traffic when the device limit is reached.

<sup>2</sup>For IDP8200, the limits shown are configured for each core IDP engine. There are six core IDP engines. The sum of “max sessions” you configure for IDP8200 can be 1,000,000 per core IDP engine. The IDP8200 device session limit rating is 6,000,000.

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
---------	-------------

**Maximum ICMP Sessions**—Controls the maximum number of ICMP sessions. If the IDP system reaches the maximum, it drops all new sessions and writes a `SESSION_LIMIT_EXCEEDED` log. Defaults vary according to model, as shown in the following table.

Model	Default	Minimum	Maximum <sup>1</sup>
IDP75	10,000	0	100,000
IDP250	25,000	0	300,000
IDP800	100,000	0	1,000,000
IDP8200 <sup>2</sup>	100,000	0	500,000
IDP200	5,000	0	70,000
IDP600	10,000	0	220,000
IDP1100	50,000	0	500,000

<sup>1</sup>For all entries except IDP8200, the maximum session limit shown is also the device session limit. We recommend that the sum of “max sessions” you configure for TCP, UDP, ICMP, and Other not exceed the device session limit. The user interface does not enforce this, so do a quick calculation whenever you change these settings. If you increase TCP, decrease UDP in proportion. Otherwise, you might encounter a traffic load that leads to undesirable results, such as drops for all traffic when the device limit is reached.

<sup>2</sup>For IDP8200, the limits shown are configured for each core IDP engine. There are six core IDP engines. The sum of “max sessions” you configure for IDP8200 can be 1,000,000 per core IDP engine. The IDP8200 device session limit rating is 6,000,000.

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
---------	-------------

**Maximum IP Sessions (non-UDP/TCP/ICMP)**—Controls the maximum number of other IP-based sessions. If the IDP system reaches the maximum, it drops all new sessions and writes a SESSION\_LIMIT\_EXCEEDED log. Defaults vary according to model, as shown in the following table.

Model	Default	Minimum	Maximum <sup>1</sup>
IDP75	10,000	0	100,000
IDP250	25,000	0	300,000
IDP800	100,000	0	1,000,000
IDP8200 <sup>2</sup>	100,000	0	500,000
IDP200	5,000	0	70,000
IDP600	10,000	0	220,000
IDP1100	50,000	0	500,000

<sup>1</sup>For all entries except IDP8200, the maximum session limit shown is also the device session limit. We recommend that the sum of “max sessions” you configure for TCP, UDP, ICMP, and Other not exceed the device session limit. The user interface does not enforce this, so do a quick calculation whenever you change these settings. If you increase TCP, decrease UDP in proportion. Otherwise, you might encounter a traffic load that leads to undesirable results, such as drops for all traffic when the device limit is reached.

<sup>2</sup>For IDP8200, the limits shown are configured for each core IDP engine. There are six core IDP engines. The sum of “max sessions” you configure for IDP8200 can be 1,000,000 per core IDP engine. The IDP8200 device session limit rating is 6,000,000.

**Reset flow table with policy load/unload**—Resets the flow table each time you load or unload a security policy. If you do not enable this option, the IDP system maintains the flow table until all flows referencing that security policy have completed. This setting is enabled by default. We recommend that you keep this setting enabled to preserve memory.

With this setting enabled, IDP system resets the flow table when you install a new policy. When the flow table is reset, existing sessions are passed through uninspected. For IDP75 and IDP200, you cannot override the default.

For high-end devices, you can unset this default to avoid passing through sessions uninspected. If you unset this default, when you load a new policy, the IDP system flow table maintains sessions belonging to the previously installed policy as well as the newly installed policy. The IDP engine continues to use the previously installed security policy to inspect previous sessions; and use the newly installed security policy to inspect new sessions. When the previously installed policy is no longer in use, it is unloaded and all traffic is inspected using the newly installed policy. For IDP8200 and IDP250, the IDP system can maintain flows for as many as two security policies. For IDP1100, IDP800, and IDP600, the IDP system can maintain flows for as many as four security policies.

**Log flow related errors**—Enables logging for flow-related errors. This setting is not enabled by default.

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
IP Actions	<p><b>Reset block table with policy load/unload</b>—Resets the IP action block table each time a security policy is loaded or unloaded. This table maintains IP addresses for connections to which the IP action block has been applied. This setting is enabled by default.</p>
Intrusion Detection	<p><b>Buffer flow emulator</b>—Turns on buffer overflow emulation.</p> <p><b>Attack matches per packet when Signature Hierarchy take effect</b>—Sets the threshold for activating signature hierarchy calculations.</p> <p>Common attack can be composed of several known vulnerabilities. Each vulnerability has an attack object, and each would generate a separate log entry if the signature hierarchy feature were disabled.</p> <p>For example, for a policy with critical, high, medium, low, and info attacks and logging enabled, a single detection of HTTP:IIS:COMMAND-EXEC attack generates the following logs:</p> <ul style="list-style-type: none"> <li>• HTTP:IIS:COMMAND-EXEC [wininnt/system32/cmd.exe] (medium)</li> <li>• HTTP:WIN-CMD:WIN-CMD-EXE [cmd.exe] (medium)</li> <li>• HTTP:REQERR:REQ-MALFORMED-URL [anomaly for %xx] (medium)</li> <li>• HTTP:DIR:TRAVERSE-DIRECTORY (anomaly for ../) (medium)</li> <li>• HTTP:REQERR:REQ-LONG-UTF8CODE (anomaly for oe) (medium)</li> <li>• TCP:AUDIT:BAD-SYN-NONSYN (info)</li> <li>• HTTP:AUDIT:URL (info)</li> <li>• TCP:AUDIT:BAD-SYN-NONSYN (info)</li> </ul> <p>If the number of attacks in a packet exceeds the set value, the IDP engine examines its signature hierarchy to see if some attacks are actually part of a larger attack. If so, only the parent attack is displayed in the logs. In this example, if the value was set to 9 or lower, only a log for HTTP:IIS:COMMAND-EXEC would be generated.</p> <p>An attack in the signature hierarchy may have multiple parents or multiple children. If a child attack is part of two discovered parents, the IDP system takes action based on the parent with the highest severity.</p> <p>Specify 0 to disable.</p>

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
Run-Time Parameters	<b>RPC program timeout</b> —Controls how long the IDP system maintains information about an RPC server. The IDP engine performs a stateful inspection of all RPC messages on port 111, then builds a table of program-to-port mapping for each RPC server that it finds on the network. The default is 300 seconds.
	<b>RPC transaction timeout</b> —Controls RPC timeout. All RPC messages (port 111) are based on a request/response protocol. When the IDP engine receives a request, it adds the request to a request table. If the IDP engine does not receive an RPC reply in the specified timeout, the RPC entry times out. The default is 5 seconds.
	<b>Exempt management server flows</b> —Exempts NSM connections from processing. This setting is enabled by default.
	<b>Fragment timeout</b> —Controls when the IDP system drops an incomplete fragment chain because one or more fragments did not arrive. If the IDP system does not receive missing fragments in the specified timeout, it generates a log (FRAGMENT_TIME_EXCEEDED). The default is 5 seconds.
	<b>Minimum fragment size</b> —Drops all IP fragments less than the specified size (bytes). The default is 0 bytes (no fragments are dropped).
	<b>Maximum fragments per IP datagram</b> —Controls size of the IP fragment chain. An IP datagram can be broken into many fragments which, when assembled, should not exceed 64 K. IP fragment processing is CPU and memory intensive. If the number of fragments in a chain exceeds this number, the IDP system drops the entire fragment chain. The default is 65,535 bytes.
	<b>Maximum concurrent fragments in queue</b> —Controls the maximum number of reassembled fragment chains. The IDP engine can perform pseudo reassembly of IP fragment chains. Once this limit is reached, the IDP system drops all new IP fragment chains and generates a log (TOO_MANY_FRAGMENTS). If your network produces a large number of IP fragments, such as those produced by Network File System (NFS), increase the number of fragments per chain to eliminate unnecessary logs. The default is 16 fragments.
	<b>Log fragment related errors</b> —Logs fragment related errors. This setting is not enabled by default.
	<b>Enable GRE decapsulation support</b> —Enables decapsulation and inspection of generic routing encapsulation (GRE) traffic. IDP Series devices support inspection of IP-in-GRE or PPP-in-GRE encapsulated traffic. GRE decapsulation is not enabled by default.
	<b>Enable GTP decapsulation support</b> —Enables decapsulation and inspection of GPRS Tunneling Protocol (GTP) traffic. IDP Series devices support decapsulation of UDP GTPv0 and GTPv1 only. GTP decapsulation is not enabled by default.
	<b>Enable SSL decryption support</b> —Enables SSL decryption and inspection. SSL decryption is not enabled by default.

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
SYN-Protector	<p><b>Timeout for half-open SYN protected flows</b>—Determines the number of seconds before the IDP system closes a half-open SYN protected flow when the SYN Protector rulebase is in passive mode. The default is 5 seconds.</p> <p>A half-open SYN flow occurs during the TCP three-way handshake, after the client has sent a SYN/ACK packet to the server. The half-open connection is now in the SYN_RECV state, and is placed into a connection queue while it waits for an ACK or RST packet. The connection remains in the queue until the connection-establishment timeout expires and the half-open connection is deleted.</p> <hr/> <p><b>Lower SYNs-per-second threshold below which SYN Protector will be deactivated / Upper SYNs-per-second threshold above which SYN Protector will be activated</b>—Determines when the SYN Protector rulebase is activated and deactivated.</p> <p>In relay mode, the SYN Protector rulebase is activated when the number of SYN packets per second is greater than the lower threshold. Relay mode does not use the upper threshold.</p> <p>In passive mode, the SYN Protector rulebase is activated when the number of SYN packets per second is greater than the sum of the lower and upper thresholds and deactivated when the number of SYNs-per-second falls below the lower threshold. The defaults are 1000 and 20. Using the defaults, the SYN Protector is activated when SYNs-per-second reach 1020 and deactivated when SYNs-per-second fall below 1000.</p>

Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

Setting	Description
TCP Reassembler	<p><b>Ignore packets in TCP flows where a SYN hasn't been seen</b>—Ignores the absence of SYN flags in TCP flows. This is enabled by default.</p> <hr/> <p><b>Timeout for connected, idle TCP flows</b>—Controls the number of seconds that the IDP system maintains connected (but idle) TCP flows. The default is 3600 seconds.</p> <hr/> <p><b>Timeout for closed TCP flows</b>—Controls the number of seconds that closed TCP flows are maintained in the flow table.</p> <p>When the IDP engine sees a RST packet or FIN/FIN+ACK packets on a TCP connection, it closes the connection flows. It drops any further packets for the closed flow, but does not delete existing, closed flows from the flow table.</p> <p>The default is 5 seconds.</p> <hr/> <p><b>Timeout for CLOSE-WAIT/LAST-ACK TCP flows</b>—Controls the number of seconds a connection is maintained while waiting for the final ACK.</p> <p>When a TCP connection closes, the IDP engine sees a FIN packet from each side of the connection followed by an ACK packet from each side of the connection. However, TCP does not guarantee delivery of the final ACK.</p> <p>To improve performance during heavy loads, decrease the timeout. Decreasing the timeout reduces the size of the flow table by closing connections sooner. The default is 120 seconds.</p> <hr/> <p><b>Close flows as soon as a FIN is seen</b>—Enables the IDP system to quickly close a TCP connection after receiving a FIN packet.</p> <p>When a TCP connection closes, the IDP engine sees a FIN packet from each side of the connection followed by an ACK packet from each side of the connection. However, TCP does not guarantee delivery of the final ACK.</p> <p>When enabled, the IDP system maintains a connection waiting for a final ACK for 5 seconds, then closes the connection. This is enabled by default and recommended.</p>



Table 99: IDP Series Device Configuration: Runtime Parameters (*continued*)

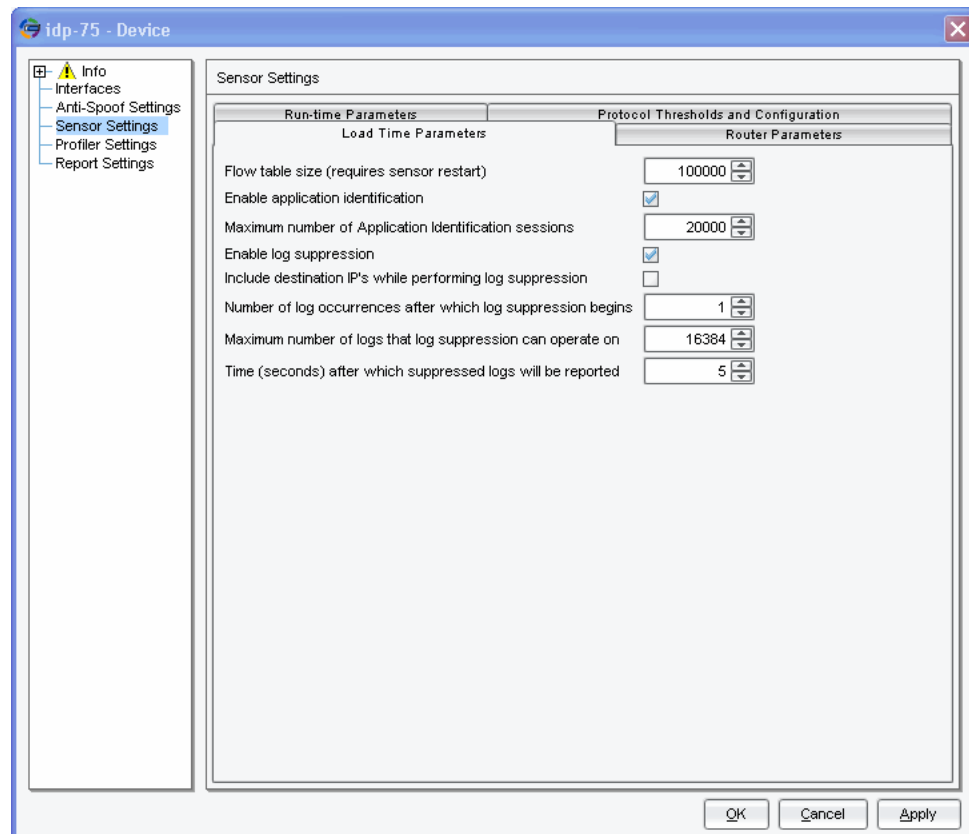
Setting	Description
Traffic Signatures	<p><b>Byte threshold for suspicious flows</b>—Specifies a threshold for what the IDP engine considers a small packet.</p> <p>A scan typically uses small packets to access its targets. You can exclude suspicious flows that contain large packets to prevent false positives when detecting scans.</p> <p>If the IDP engine sees more than this maximum, it does not consider the connection to be a scan. The default is 20 bytes.</p> <hr/> <p><b>Reporting frequency when scan is in progress</b>—Controls how often the IDP system generates "in progress" logs for a stealthy scan.</p> <p>Attackers can perform blatant scans very quickly, mapping your network in just a few seconds, but these scans typically trigger intrusion detection systems and leave evidence behind. Stealthy scans are performed over much longer time periods, lasting hours, days, or even weeks, making them more difficult to detect. The default is 30 seconds.</p> <hr/> <p><b>The number of IP tracked for session rate</b>—Controls the number of IP addresses tracked by the session rate counter. If the IDP engine sees more addresses than the maximum, it does not track the additional IP addresses. The default is 32,767 IP addresses.</p>

## Modifying Load-Time Parameters

Load-time parameters include options for tuning performance. In general, you modify these settings only if you encounter performance issues.

[Figure 130 on page 366](#) shows the Load Time Parameters tab, where you can configure these settings.

Figure 130: NSM Device Configuration Editor: Load Time Parameters Tab



To modify parameters:

1. In NSM Device Manager, double-click the IDP Series device you want to modify to display the device configuration editor.
2. Click **Sensor Settings**.
3. Click the **Load Time Parameters** tab.
4. Configure parameters as described in [Table 100 on page 366](#).
5. Click **Apply**.
6. Click **OK**.

Table 100: IDP Series Device Configuration: Load Time Parameters

Setting	Guideline
Flow table size	For improved performance, modify the flow table size to limit the size of the connection table. This setting should reflect the maximum number of concurrent flows you expect to have at any one time. A TCP connection has about two flows per session, and a UDP connection has about three flows per session. The default setting is 100,000 concurrent flows. If you change this value, you have to restart the IDP Series device.

Table 100: IDP Series Device Configuration: Load Time Parameters (*continued*)

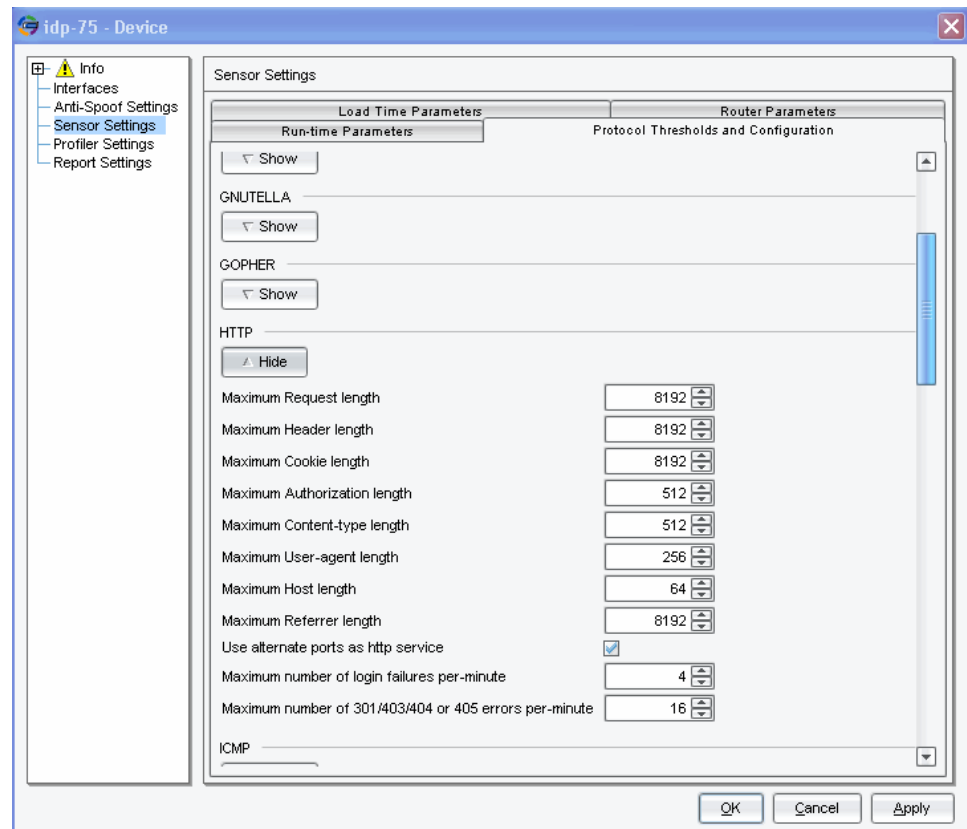
Setting	Guideline
Enable application identification	The application identification feature is used to detect the session application regardless of port. We recommend you disable this feature only when troubleshooting.
Maximum number of Application Identification sessions	Specifies the maximum number of sessions where application identification is in use. The default is 50,000. Valid values are 0 - 200,000. We recommend you tune this setting only if you encounter issues.
Enable log suppression	Log suppression reduces the number of logs displayed in the Log Viewer by displaying a single record for multiple occurrences of the same event.
Include destination IPs while performing log suppression	When log suppression is enabled, multiple occurrences of events with the same source IP, Service, and matching attack object generate a single log record with a count of occurrences. If you enable this option, log suppression combines log records for events with the same destination IP.
Number of log occurrences after which log suppression begins	The number of identical log records received before suppression starts. The default is 1 (meaning log suppression begins with the first redundancy).
Maximum number of logs that log suppression can operate on	When log suppression is enabled, the IDP system must cache log records so that it can identify when multiple occurrences of the same event occur. This number represents the number of log records cached for this purpose. The default is 16,384 log records.
Time (seconds) after which suppressed logs will be reported	When log suppression is enabled, the IDP system maintains a count of multiple occurrences of the same event. This number represents the number of seconds that pass before reporting a single log entry that contains the count of occurrences. The default is 10 seconds.  <b>NOTE:</b> If the reporting interval is set too high, log suppression can negatively impact performance.

## Modifying Protocol Anomaly Thresholds

The protocol anomaly detection methods identify traffic that deviates from RFC specifications. In general, you modify protocol thresholds and configuration settings only if you encounter false positives or performance issues.

[Figure 131 on page 368](#) shows the Protocol Thresholds and Configuration tab, where you can configure these settings.

Figure 131: NSM Device Configuration Editor: Protocol Thresholds and Configuration Tab



To tune protocol anomaly detection thresholds:

1. In NSM Device Manager, double-click the IDP Series device you want to modify to display the device configuration editor.
2. Click **Sensor Settings**.
3. Click the **Protocol Thresholds and Configuration** tab.
4. Complete the configuration for protocol thresholds as described in [Table 101 on page 369](#).
5. Click **Apply**.
6. Click **OK**.

Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings

Setting	Description
AIM	<b>Maximum header length</b> —Detects a header containing more bytes than the specified maximum. The default is 10,000 bytes.
	<b>Maximum type-length-value length</b> —Detects an AIM/ICQ type-length-value (TLV) containing more bytes than the specified maximum. A TLV is a tuple used for passing typed information to the protocol. The default is 8,000 bytes.
	<b>Maximum inter-client-message-block length</b> —Detects an AIM/ICQ inter-client-message-block (ICMB) containing more bytes than the specified maximum. The default is 2,000 bytes.
	<b>Maximum filename length</b> —Detects an AIM/ICQ filename containing more bytes than the specified maximum. The default is 10,000 bytes.
DHCP	<b>Check to see if the source port of client's packets is 68</b> —Detects DHCP traffic that originates from a port other than 68. This setting is not enabled by default.
DNS	<b>Report unknown DNS parameters (high noise)</b> —Detects and reports unknown DNS parameters.  You must also configure an IDP rulebase rule to detect DNS anomalies. This setting is not enabled by default.
	<b>Report unexpected DNS parameters (high noise)</b> —Detects and reports unexpected DNS parameters. This setting is not enabled by default.  You must also configure an IDP rulebase rule to detect DNS anomalies.
	<b>Maximum length of a DNS UDP packet</b> —Detects a DNS UDP packet containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum number of pointer loops for name compression</b> —The default is 8.
	<b>Maximum size of an NXT resource record</b> —Detects an NXT resource record in a DNS request or response message that is larger than the specified maximum size. The default is 4,096 bytes.  This setting tunes the <a href="#">DNS_BIND_NXT_OVERFLOW</a> protocol anomaly.
	<b>Maximum time of a DNS cache</b> —Controls the maximum amount of time for a DNS query and reply. The default is 60 seconds.
	<b>Maximum size of a DNS cache</b> —Controls the maximum number of DNS queries kept to match a reply. The default is 100 queries.
	<b>Maximum number of logs in a session</b> —The default is 128.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
FTP	<b>Maximum Line length</b> —Detects an FTP line containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum Username length</b> —Detects an FTP username containing more bytes than the specified maximum. The default is 32 bytes.
	<b>Maximum Password length</b> —Detects an FTP password containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum Pathname length</b> —Detects an FTP pathname containing more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum Sitestring length</b> —Detects an FTP site string containing more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum number of login failures per minute</b> —Detects more FTP login failures in one minute than the specified maximum. The default is 4 FTP login failures per minute.
GNUTELLA	<b>Maximum TTL hops</b> —Detects a number of TTL hops that is higher than the specified maximum. The default is 8 TTL hops.
	<b>Maximum line length</b> —Detects, in a Gnutella connection, a line that contains more bytes than the specified maximum. The default is 2,048 bytes.
	<b>Maximum query size</b> —Detects a Gnutella client query that contains more bytes than the specified maximum. The default is 256 bytes.
GOPHER	<b>Maximum line length</b> —Detects, in a Gopher server-to-client connection, a line sent by a Gopher server to a client that contains more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum hostname length</b> —Detects, in a Gopher server-to-client connection, a hostname that contains more bytes than the specified maximum. The default is 64 bytes.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
HTTP	<b>Maximum Request length</b> —Detects an HTTP request that contains more bytes than the specified maximum. The default is 8,192 bytes.
	<b>Maximum Header length</b> —Detects an HTTP header that contains more bytes than the specified maximum. The default is 8,192 bytes.
	<b>Maximum Cookie length</b> —Detects a cookie that contains more bytes than the specified maximum. The default is 8,192 bytes.  Cookies that exceed the cookie length setting can match the <a href="#">HTTP: Cookie Overflow</a> protocol anomaly and produce unnecessary log records. If you are getting too many log records for the <a href="#">HTTP: Cookie Overflow</a> protocol anomaly, increase the maximum cookie length.
	<b>Maximum Authorization length</b> —Detects an HTTP header authorization line that contains more bytes than the specified maximum. The default is 512 bytes.  Use this setting to tune results matching the <a href="#">HTTP:OVERFLOW:AUTH-OVFLW</a> protocol anomaly.
	<b>Maximum Content-type length</b> —Detects an HTTP header content-type that contains more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum User-agent length</b> —Detects an HTTP header user-agent that contains more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum Host length</b> —Detects an HTTP header host that contains more bytes than the specified maximum. The default is 256 bytes.
	<b>Maximum Referrer length</b> —Detects an HTTP header referrer that contains more bytes than the specified maximum. The default is 8,192 bytes.
	<b>Use alternate ports as http service</b> —Detects HTTP traffic on the following ports in addition to tcp/80: 7001; 8000; 8001; 8100; 8200; 8080; 8888; 9080. This setting is enabled by default.  <b>NOTE:</b> In IDP OS Release 5.0 and later, this setting is no longer functional. The IDP engine now automatically detects HTTP traffic over any port.
	<b>Maximum number of login failures per-minute</b> —Detects login failures more frequent than the specified maximum. The default is 5 HTTP authentication failures per minute.  This setting tunes the <a href="#">HTTP: Brute Force Login Attempt</a> protocol anomaly.
	<b>Maximum number of 301/403/404 or 405 errors per-minute</b> —Detects 301/403/404/405 errors that occur more frequently than the specified maximum. The default is 16 HTTP errors per minute.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
ICMP	<b>Maximum Packets per second to trigger a flood</b> —Raises a protocol anomaly if the IDP engine detects more ICMP packets than the specified maximum. The default is 250 packets per second.
	<b>Minimum time interval (in seconds) between packets</b> —Detects ICMP packets that have less than the specified minimum time interval between them. The default is 1 second.
	Use this setting to tune the <a href="#">ICMP:EXPLOIT:FLOOD</a> protocol anomaly.
IDENT	<b>Maximum requests per session</b> —Detects more IDENT (identification protocol) requests than the specified maximum. The default is 1 request per session.
	This setting tunes the <a href="#">IDENT: Too Many Requests</a> protocol anomaly.
	<b>Maximum Request length</b> —Detects an IDENT request containing more bytes than the specified maximum. The default is 15 bytes.
	This setting tunes <a href="#">IDENT: Request Too Long</a> protocol anomaly.
IKE	<b>Maximum Reply length</b> —Detects an IDENT reply containing more bytes than the specified maximum. The default is 128 bytes.
	This setting tunes the <a href="#">IDENT: Reply Too Long</a> protocol anomaly.
	<b>Maximum number of payloads in an IKE message</b> —Detects an IKE message with a number of payloads larger than the specified maximum. The default is 57 payloads.
	This setting tunes the <a href="#">IKE: Too Many Payloads</a> protocol anomaly.



Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings (*continued*)

Setting	Description
IMAP	<b>Maximum line length</b> —Detects an IMAP line containing more bytes than the maximum. The default is 2,048 bytes.
	<b>Maximum Username length</b> —Detects an IMAP username containing more bytes than the maximum. The default is 64 bytes.
	<b>Maximum Password length</b> —Detects an IMAP password containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum Mailbox length</b> —Detects an IMAP mailbox containing more than the maximum. The default is 64 bytes.
	<b>Maximum Reference length</b> —Detects an IMAP reference containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum Flag length</b> —Detects an IMAP flag containing more bytes than the specified maximum. The default is 64 bytes.
	<p><b>Maximum Literal length</b>—Detects a literal with more octets than the specified maximum. In IMAP4 protocol, a string can be in one of two forms: literal and quoted. As defined in RFC 2060 4.3, a literal is a sequence of zero or more octets (including CR and LF), prefix-quoted with an octet count in the form of an open brace ("{"), the number of octets, close brace ("}"), and CRLF. Valid range is 1 to 16,777,215. The default is 1,048,576 bytes.</p> <p>This setting tunes the <a href="#">IMAP: Literal Length Overflow</a> protocol anomaly.</p>
IRC	<b>Maximum number of login failures per minute</b> —Detects a brute force protocol anomaly if the IDP engine detects more login failures than the maximum. The default is 4 IMAP login failures per minute.
	This setting tunes the <a href="#">IMAP: Brute Force Login Attempt</a> protocol anomaly.
	<b>Maximum Password length</b> —Detects an Internet Relay Chat (IRC) password containing more bytes than the specified maximum. The default is 16 bytes.
	<b>Maximum Username length</b> —Detects an IRC username containing more bytes than the specified maximum. The default is 16 bytes.
	<b>Maximum Channel length</b> —Detects an IRC channel name containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum Nickname length</b> —Detects an IRC nickname containing more bytes than the specified maximum. The default is 16 bytes.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
LDAP	<b>Maximum length of integer representation in BER encoding</b> —Detects an integer field of the LDAP BER containing more bytes than the specified maximum. The default is 4 bytes.
	<b>Maximum number of left zeros for tag in BER encoding</b> —Detects more left zeros in any tag in LDAP BER encoding than the specified maximum. The default is 4 left zeros.
	<b>Maximum value of any LDAP tag in BER encoding</b> —Detects a value for a tag that can be seen in the LDAP BER encoding that is greater than the specified maximum. LDAP tags are represented using 1 byte, with the top 3 bits reserved. The default is 31.
	<b>Maximum number of left zeros for length in BER encoding</b> —Detects more left zeros in any length field in LDAP BER encoding than the specified maximum. The default is 64 left zeros.
	<b>Maximum number of search results requested by LDAP client</b> —Detects an LDAP client request for more matching entries than the specified maximum. The default is 0 (indicating no limit).
	<b>Maximum timelimit for search result requested by LDAP client</b> —Detects a time limit greater than the specified maximum. The time limit is the number of seconds before a client request times out waiting for a response from the server. The default is 0 (indicating no limit).
	<b>Maximum length of an LDAP Attribute Descriptor</b> —Detects a length of an attribute descriptor field in an LDAP message containing more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum length of an LDAP Distinguished Name</b> —Detects a length of a distinguished name field in the LDAP message containing more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum value of Message id in any LDAP Message</b> —Detects a message ID greater than the specified maximum. The default is 2,147,483,647.
	<b>Maximum length of an LDAP message</b> —Detects an LDAP message that will be processed by the LDAP subsystem larger than the specified maximum. The default is 8,100 bytes.
	This setting tunes the <a href="#">LDAP: Message Too Long</a> protocol anomaly.
	<b>Maximum number of nested operators in an LDAP search request</b> —Detects a number of nested levels allowed in an LDAP search request filter argument greater than the specified maximum. The default is 8 nested operators.
	<b>Maximum Number of Login Failures Per Minute</b> —Detects a brute force protocol anomaly if the IDP engine detects more login failures than the maximum. The default is 4 LDAP login failures per minute.
	This setting tunes the <a href="#">LDAP: Brute Force Login Attempt</a> protocol anomaly.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
LPR	<p><b>Maximum Sub-command length in RECEIVE-JOB Command</b>—Detects in a Line Printer Protocol (LPR) control file a subcommand line containing more bytes than the specified maximum. LPR is a TCP-based print server protocol used by line printer daemons (client and server) to communicate over networks. An LPR client uses the LPR protocol to send a print command to an LPR server (a line printer) at TCP/515. After the print command is received by the server, the client can issue subcommands to the server and send control and data files. Control files tell the line printer which functions to perform when printing the file; data files carry the payload. The default is 256 bytes.</p> <p><b>Maximum Reply length from server</b>—Detects an LPR control filename containing more bytes than the specified maximum. The default is 256 bytes.</p> <p><b>Maximum Control filename length</b>—Detects an LPR control filename containing more bytes than the specified maximum. The default is 64 bytes.</p> <p><b>Maximum Data filename length</b>—Detects a data filename containing more bytes than the specified maximum. The default is 64 bytes.</p> <p><b>Maximum Control file size</b>—Detects an LPR control file size greater than the specified maximum. The default is 1,024 bytes.</p> <p><b>Maximum Data file size</b>—Detects an LPR data file size greater than the specified maximum. The default is 65,535 bytes.</p> <p><b>Maximum Banner string length</b>—Detects an LPR banner string containing more bytes than the specified maximum. A banner string is typically the filename of the print job. The default is 32 bytes.</p> <p><b>Maximum E-mail length</b>—Detects an LPR control file e-mail address containing more bytes than the specified maximum. After the file has printed, it is sent to the e-mail address specified in the control file. The default is 32 bytes.</p> <p><b>Maximum Symbolic link length</b>—Detects in an LPR control file a symbolic link containing more bytes than the specified maximum. A symbolic link is a file that points to another file (entry) in a UNIX file system, but does not contain the data in the target file. When the LPR protocol receives a symbolic link command in a control file, it records the symbolic link data for the print job filename to prevent directory entry changes from reprinting the file. The default maximum is 128 bytes.</p> <p><b>Maximum font length</b>—Detects in an LPR control file a font name containing more bytes than the specified maximum. The default is 64 bytes.</p> <p><b>Maximum filename length for format related sub commands</b>—Detects in an LPR control file a format-related filename containing more bytes than the specified maximum. The default is 32 bytes.</p>

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
MSN	<b>Maximum Username length</b> —Detects an MSN (Microsoft Instant Messaging) username containing more bytes than the specified maximum. The default is 84 bytes.
	<b>Maximum Display name length</b> —Detects an MSN display name containing more bytes than the specified maximum. The default is 128 bytes.
	<b>Maximum Group name length</b> —Detects an MSN group name containing more bytes than the specified maximum. The default is 84 bytes.
	<b>Maximum User state length</b> —Detects an MSN user state containing more bytes than the specified maximum. A user state is a three-letter code that indicates the status of the user's connection (online, offline, idle, and the like). The default is 10 bytes.
	<b>Maximum Phone number length</b> —Detects a phone number containing more bytes than the specified maximum. The default is 20 bytes.
	<b>Maximum Length of IP:port</b> —Detects an IP:port parameter containing more bytes than the specified maximum. An IP:port parameter indicates the IP address and port number of the MSN server for a switchboard session. The default is 30 bytes.
	<b>Maximum URL length</b> —Detects a URL containing more bytes than the specified maximum. The default is 1,024 bytes.
MSRPC	<b>Maximum fragment length in MSRPC message</b> —Detects an MSRPC (Microsoft Remote Procedure Call) message with a fragment length greater than the specified maximum. The default is 8,192.
	<b>Maximum tower data length in endpoint mapper messages</b> —Detects an endpoint mapper message with a tower data length greater than the specified maximum. The default is 8,192.
	<b>Maximum number of entries in an insert message</b> —Detects an MSRPC insert message with more entries than the specified maximum. The default is 100 entries.
NFS	<b>Maximum name length</b> —Detects an NFS packet name containing more bytes than the specified maximum. The default is 256 bytes.
	<b>Maximum path length</b> —Detects an NFS packet pathname containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum buffer length for read/write</b> —Detects an NFS read/writer buffer larger than the specified maximum. The default is 32,768 bytes.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
NTP	<b>Minimum time (in seconds) between two requests</b> —Detects that the time between two client-to-server NTP requests is greater than the specified maximum. Valid values range from 64 to 1024 seconds. The default is 0 seconds (which turns the feature off).
	<b>Maximum length for NTPv3 message</b> —Detects an NTPv3 message containing more bytes than the specified maximum. The default is 68 bytes.
	<b>Maximum length for NTPv4 message</b> —Detects an NTPv4 message containing more bytes than the specified maximum. The default is 68 bytes.
	<b>Maximum stratum value for any NTP peer</b> —Detects a stratum value larger than the specified maximum. The default is 15 bytes.
	<b>Maximum time since last update of Reference clock</b> —Detects that the NTP reference clock has not been updated in more time than the specified maximum. The default is 86,400 seconds.
	<b>Match timestamps on NTP request and response</b> —Enables the IDP engine to perform timestamp matching on client requests and server responses. With this setting enabled, the IDP engine expects the server response original timestamp to match the client request transmit timestamp; otherwise it considers the packet a possible protocol anomaly. This setting is enabled by default.
	<b>Maximum Authorization field length in NTP control message</b> —Detects that the length of the Authentication field in an NTP control message is larger than the specified maximum. The default is 20 bytes.
	<b>Maximum length of any NTP control variable</b> —Detects that the length of the NTP control data variable name is larger than the specified maximum. The default is 128 bytes.
	<b>Maximum length of any NTP variable value</b> —Detects that the length of any NTP control data variable value is larger than the specified maximum. The default is 255 bytes.
	<b>Maximum length of buffer to store between control packets</b> —Detects that the buffer used to store NTP control messages is greater than the specified maximum. NTP control messages can be split across multiple UDP packets. The default is 255 bytes.
	<b>Maximum time for an NTP Symmetric passive association to dissolve</b> —Specifies the duration in seconds after which the IDP engine considers an NTP symmetric passive association as expired. A symmetric passive association between two NTP peers must be dissolved after sending one reply. The default is 900 seconds.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
POP3	<b>Maximum Line length</b> —Detects a POP3 line containing more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum Username length</b> —Detects a POP3 username containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum Password length</b> —Detects a POP3 password containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum APOP length</b> —Detects an APOP containing more bytes than the specified maximum. The default is 100 bytes.
	<b>Maximum message number</b> —Detects a POP3 message number that is higher than the specified maximum. The default is 1,000,000.
	<b>Maximum Number of Login Failures Per Minute</b> —Raises a brute force protocol anomaly if the IDP engine detects more login failures than the specified maximum. The default is 4 POP3 login failures per minute.  This setting tunes the <a href="#">POP3: Brute Force Login Attempt</a> protocol anomaly.
RADIUS	<b>Maximum Number of Authenticated Failures Per Minute</b> —Raises a brute force protocol anomaly if the IDP engine detects more login failures than the specified maximum. The default is 4 RADIUS login failures per minute.  This setting tunes the <a href="#">RADIUS: Brute Force</a> protocol anomaly.
SIP	<b>Max Forwards Threshold</b> —Detects if the value in the Max-Forwards header field is greater than the specified value. The default is 70.
SMB	<b>Maximum registry key length</b> —Detects an SMB registry key containing more bytes than the specified maximum. The default is 8,192 bytes.
	<b>Maximum Number of Login Failures Per Minute</b> —Raises a brute force protocol anomaly if the IDP engine detects more login failures than the specified maximum. The default is 4 SMB login failures per minute.
	This setting tunes the <a href="#">SMB: Brute Force Login Attempt</a> protocol anomaly.

Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings (*continued*)

Setting	Description
SMTP	<b>Maximum Number of mail recipients</b> —Detects an SMTP message containing more recipients than the specified maximum. The default is 100 recipients.
	<b>Maximum Username length in RCPT and MAIL</b> —Detects an SMTP message with a username containing more bytes than the specified maximum. The default is 256 bytes.
	<b>Maximum Domain name length in RCPT and MAIL</b> —Detects an SMTP message with a domain name containing more bytes than the specified maximum. The default is 64 bytes.
	<b>Maximum Path length in RCPT and MAIL</b> —Detects an SMTP message with a pathname containing more bytes than the specified maximum. The default is 256 bytes.
	<b>Maximum Command line length (before DATA)</b> —Detects an SMTP message with a command-line entry containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum Reply line length from server (default)</b> —Detects an SMTP message with a reply line from the server containing more bytes than the specified maximum. The default is 512 bytes.
	<b>Maximum Text line length (after DATA)</b> —Detects an SMTP text line containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum number of nested mime multi-part attachments</b> —Detects more nested attachments than the specified maximum. The default is 4 nested mime multi-part attachments.
	<b>Maximum number of base-64 bytes to decode</b> —Detects more bytes of encoded mime data than the specified maximum. The default is 64 bytes.
	<b>Maximum length of the value for content-type's name attribute</b> —Detects a name attribute in the content-type header containing more bytes than the specified maximum. The default is 128 bytes.
SYSLOG	<b>Maximum length of the value for the content-disposition's filename attribute</b> —Detects a filename attribute in the content-disposition header containing more bytes than the specified maximum. The default is 128 bytes.
	<b>Look for email headers in message data</b> —Controls whether the IDP engine looks for e-mail headers in the message data, which can occur when a bounced e-mail contains an attachment. This setting is not enabled by default.
SYNLOG	<b>Validate RFC-3164 compliant timestamp format</b> —Raises a protocol anomaly if the timestamp in syslog traffic is not compliant with RFC 3164. This setting is not enabled by default.
TELNET	<p><b>Maximum Number of Login Failures Per Minute</b>—Raises a brute force protocol anomaly if the IDP engine detects more login failures than the specified maximum. The default is 4 Telnet login failures per minute.</p> <p>This setting tunes the <a href="#">Telnet:: Brute Force Login Attempt</a> protocol anomaly.</p>
TFTP	<b>Maximum filename length</b> —Detects a filename containing more bytes than the specified maximum. The default is 128 bytes.

**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
VNC	<b>Maximum Reason string length</b> —Detects a VNC (Virtual Network Computing) reason string length greater than the specified maximum. A reason string contains the text that describes why a connection between a VNC server and client failed. The default is 512 bytes.
	<b>Maximum Display name length</b> —Detects a VNC display name containing more bytes than the specified maximum. The default is 128 bytes.
	<b>Maximum cut text length</b> —Detects a VNC cut text buffer containing more bytes than the specified maximum. The default is 4,096 bytes.
	<b>Verify message after the initial handshake</b> —Enables the IDP engine to verify VNC connections after the initial handshake. This setting is not enabled by default.
	<b>Maximum Number of Login Failures Per Minute</b> —Raises a brute force protocol anomaly if the IDP engine detects more login failures than the specified maximum. The default is 4 VNC login failures per minute.  This setting tunes the <a href="#">VNC: Brute Force Login Attempt</a> protocol anomaly.
WHOIS	<b>Maximum Request length</b> —Detects a WHOIS request containing more bytes than the specified maximum. The default is 128 bytes.



**Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings** (*continued*)

Setting	Description
YMSG	<b>Maximum Message length</b> —Detects a Yahoo! Messenger message with a header that indicates more bytes for the total message than the specified maximum. The default is 8,192 bytes.
	<b>Maximum Username length</b> —Detects a Yahoo! Messenger username containing more bytes than the specified maximum. The default is 84 bytes.
	<b>Maximum Groupname length</b> —Detects a Yahoo! Messenger group name containing more bytes than the specified maximum. The default is 84 bytes.
	<b>Maximum Crypt length</b> —Detects a Yahoo! Messenger encrypted password containing more bytes than the specified maximum. The default is 124 bytes.
	<b>Maximum Instant message length</b> —Detects a Yahoo! Messenger message containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum Activity string length</b> —Detects a Yahoo! Messenger activity data type containing more bytes than the specified maximum. The default is 8,000 bytes.
	<b>Maximum Challenge length</b> —Detects a Yahoo! Messenger challenge containing more bytes than the specified maximum. The default is 15 bytes.
	<b>Maximum Cookie length</b> —Detects a Yahoo! Messenger cookie containing more bytes than the specified maximum. The default is 84 bytes.
	<b>Maximum URL length</b> —Detects a Yahoo! Messenger Web Name containing more bytes than the specified maximum. The default is 400 bytes.
	<b>Maximum Conference message length</b> —Detects a Yahoo! Messenger join conference message containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum Conference name length</b> —Detects a Yahoo! Messenger conference name containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum E-mail length</b> —Detects a Yahoo! Messenger new e-mail alert containing an e-mail that has more bytes than the specified maximum. The default is 84 bytes.
	<b>Maximum E-mail subject length</b> —Detects an Yahoo Messenger e-mail subject line containing more bytes than the specified maximum. The default is 128 bytes.
	<b>Maximum Filename length</b> —Detects a Yahoo! Messenger file transfer containing a filename that has more bytes than the specified maximum. The default is 1,000 bytes.
	<b>Maximum Chatroom name length</b> —Detects a Yahoo! Messenger chat room name containing more bytes than the specified maximum. The default is 1,024 bytes.
	<b>Maximum Chatroom message length</b> —Detects a Yahoo! Messenger chat room message containing more bytes than the specified maximum. The default is 2,000 bytes.

Table 101: IDP Series Device Configuration: Protocol Thresholds and Configuration Settings (*continued*)

Setting	Description
	<b>Maximum buddy list length</b> —Detects a Yahoo! Messenger buddy list containing more bytes than the specified maximum. The default is 8,000 bytes.
	<b>Maximum webcam key length</b> —Detects a Yahoo! Messenger Webcam key containing more bytes than the specified maximum. The default is 124 bytes.

- Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:
- [NSM Device Configuration Management Task Summary on page 343](#)

## Deleting an IDP Series Device Configuration from NSM Device Manager (NSM Procedure)

To delete a device from NSM Device Manager:

1. In the NSM navigation tree, navigate to **Device Manager > Devices**.
2. Right-click the device you want to delete and select **Delete**.

The Delete dialog box appears. NSM analyzes administration objects for references to the device marked for deletion. If there are any references, you can click on the links that appear to view or edit the objects that reference the device.

3. To delete the device, click **Next**.

The Delete dialog box displays the progress of the deletion.

4. Click **Finish** to close the dialog box.

- Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:
- [NSM Device Configuration Management Task Summary on page 343](#)
  - [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## CHAPTER 34

# Managing IDP Processes

- [Restarting the IDP Engine on page 383](#)
- [Rebooting and Shutting Down the IDP Series Appliance on page 384](#)

### Restarting the IDP Engine

You use the **idp.sh** command to start, stop, or restart the main IDP process.

The **idp.sh** utility is located in **/usr/idp/device/bin**.

[Table 102 on page 383](#) identifies operations you perform with IDP Series user interfaces.

**Table 102: Operations Requiring Use of Particular IDP Series User Interfaces**

Task	CLI	ACM	NSM
Start the IDP main process.	idp.sh start	No	No
Stop the IDP main process.	idp.sh stop	No	No
Restart the IDP main process.	idp.sh restart	No	No
Reload all IDP processes	idp.sh reload	No	No
Reboot the IDP Series device.	reboot	Yes	Yes
Shutdown the IDP Series device.	shutdown	No	No
Start the Profiler.	No	No	Yes

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)
- [idp.sh Command Reference on page 385](#)
- [Rebooting and Shutting Down the IDP Series Appliance on page 384](#)

## Rebooting and Shutting Down the IDP Series Appliance

Table 103 on page 384 describes the commands you use to reboot and shut down the IDP Series appliance.

**Table 103: IDP Series Appliance Reboot and Shutdown Commands**

Command	Usage
<b>reboot</b>	<p>Reboots the IDP Series appliance.</p> <p>You reboot under the following circumstances:</p> <ul style="list-style-type: none"> <li>• After installing software updates</li> <li>• After changing interface settings</li> </ul> <p>You do not need to reboot under the following circumstances:</p> <ul style="list-style-type: none"> <li>• Installing customer service patches</li> <li>• Pushing IDP detector engine updates</li> <li>• Pushing attack object updates</li> <li>• Pushing configuration updates</li> </ul> <p><b>TIP:</b> If you are not sure whether the IDP Series appliance requires a reboot, log in to ACM. The ACM home page indicates whether the IDP Series appliance is in a state that requires or does not require a reboot.</p> <p><b>NOTE:</b> The <b>reboot</b> command is different from <b>idp.sh restart</b>, which restarts the IDP processes.</p>
<b>shutdown</b>	<p>Shuts down the IDP Series appliance.</p> <p>You shut down the IDP Series appliance under the following circumstances:</p> <ul style="list-style-type: none"> <li>• Before replacing cold-swappable components, such as I/O modules on high-end models and power supplies on low-end models.</li> <li>• To take the IDP Series appliance out of use.</li> </ul> <p><b>NOTE:</b> You do not need to shut down to replace hot-swappable components.</p> <p><b>NOTE:</b> If you have enabled internal bypass, traffic passes through the IDP Series device uninspected when the device is shut down.</p>

- Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:
- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)
  - [Connecting to ACM on page 191](#)
  - [Restarting the IDP Engine on page 383](#)

## idp.sh Command Reference

**Syntax** `idp.sh option`

**Description** Starts, stops, and restarts the IDP engine.

**Options** [Table 104 on page 385](#) describes **idp.sh** options.

**Table 104: Command Reference: idp.sh**

Option	Usage and Examples
start	<p>Starts the main IDP process.</p> <pre>[root@idp ~]# idp.sh start Starting apps... Freezing security settings on /dev/hda...Done 10:48:24 Setting NICs to defined settings.....Done 10:48:25 Start idpinit.....ok 10:48:26 Starting idpengine.....ok 10:48:28 Enabling layer 2 bypass.....ok 10:48:28 Attaching to NICs.....ok 10:48:28 Setting vr modes.....ok Trying to load last policy [policy-test]...done Starting zero copy driver.....OK Starting idpLogReader.....OK Starting agent.....OK Starting idpHMD.....OK Starting scioid.....OK Starting pkid.....OK</pre>
stop	<p>Stops the main IDP process.</p> <pre>[root@idp ~]# idp.sh stop  Stopping apps... sctop: no process killed          Stopped scioid (pid 26245) No pidfile for pkid Cleaning up remaining instances of pkid... Stoppingok 10:47:07 Stopping idpinit.....ok 10:47:09 Stopping idpengine.....ok</pre>

Table 104: Command Reference: idp.sh (*continued*)

Option	Usage and Examples
reload	<p>Reloads all IDP processes.</p> <pre>[root@idp ~]# idp.sh reload Stopping apps... sctop: no process killed wc: write error: Broken pipe          Stopped scioid (pid 8774) No pidfile for pkid Cleaning up remaining instances of pkid... Stoppinok 10:50:01 Stopping idpinit.....ok 10:50:03 Stopping idpengines.....ok Starting apps... Freezing security settings on /dev/hda...Done 10:50:08 Setting NICs to defined settings.....Done 10:50:09 Start idpinit.....ok 10:50:10 Starting idpengines.....ok 10:50:11 Enabling layer 2 bypass.....ok 10:50:12 Attaching to NICs.....ok 10:50:12 Setting vr modes.....ok Trying to load last policy [policy-test]...done Starting zero copy driver.....OK Starting idpLogReader.....OK Starting agent.....OK Starting idpHMD.....OK Starting scioid.....OK Starting pkid.....OK</pre>
restart	<p>Restarts the main IDP process.</p> <pre>[root@idp ~]# idp.sh restart  Stopping apps... sctop: no process killed          Stopped scioid (pid 20702) No pidfile for pkid Cleaning up remaining instances of pkid... Stoppingok 10:43:44 Stopping idpinit.....ok 10:43:46 Stopping idpengines.....ok Starting apps... Freezing security settings on /dev/hda...Done 10:43:51 Setting NICs to defined settings.....Done 10:43:52 Start idpinit.....ok 10:43:53 Starting idpengines.....ok 10:43:55 Enabling layer 2 bypass.....ok 10:43:55 Attaching to NICs.....ok 10:43:55 Setting vr modes.....ok Trying to load last policy [policy-test]...done Starting zero copy driver.....OK Starting idpLogReader.....OK Starting agent.....OK Starting idpHMD.....OK Starting scioid.....OK Starting pkid.....OK</pre>

Table 104: Command Reference: `idp.sh` (*continued*)

Option	Usage and Examples
status	<p>Displays status information for IDP processes.</p> <p>The following example shows the output of the <code>idp.sh status</code> command:</p> <pre>[root@default host admin]# idp.sh status Retrieving status... idpinit (pid 6618).....ON idpengine_0 (pid 7217).....ON idpengine_1 (pid 7207).....ON idpengine_2 (pid 7211).....ON idpengine_3 (pid 7209).....ON idpengine_4 (pid 7219).....ON idpengine_5 (pid 7257).....ON idpLogReader (pid 12524).....ON agent (pid 12450).....ON idpHMD (pid 12510).....ON sciiod (pid 12517).....ON pkid (pids 12566).....ON</pre>
version	<p>Displays version information for IDP processes.</p> <p>The following example shows the output of the <code>idp.sh version</code> command:</p> <pre>[root@default host admin]# idp.sh version Retrieving version information... scio 5.1.136718 kernel 5.1.136718 idpLogReader 5.1.136718 agent 5.1.136718 idpHMD 5.1.136718 sciiod 5.1.136718 pkid 5.1.136718.</pre>





# Updating IDP Software

- Upgrading Software (CLI Procedure) on page 389
- Updating IDP OS Software (NSM Procedure) on page 390
- Loading J-Security Center Updates (NSM Procedure) on page 392

## Upgrading Software (CLI Procedure)

---

This topic provides the basic procedure for upgrading IDP OS software. Check the [release notes](#) to see the exact procedure tested for a particular release.

When you upgrade IDP OS software, you perform the following basic steps:

1. Upgrade IDP OS software (CLI or NSM).
2. Update IDP detector engine (NSM).
3. Update the NSM attack object database (NSM).
4. Update the security policy installed on the IDP Series device (NSM).

To upgrade IDP OS software from the CLI:

1. Download the software image to a host that runs an FTP server. Follow these steps:
  - a. Go to <https://www.juniper.net/customers/csc/software/> and log in with your customer username and password.
  - b. Enter the IDP Series device serial number to display a view of applicable software releases available for download.
  - c. Click the applicable link to display the software download page.
  - d. Save the **sensor\_version.sh** file (where version is the number that identifies the software release version).
2. Connect to the IDP OS command-line interface in one of the following ways:
  - Use SSH to connect to the IP address or hostname for the management interface. Log in as **admin** and enter **su -** to switch to **root**.
  - If you prefer, make a connection through the serial port and log in as **root**.



**NOTE:** To make an SSH connection, you must have enabled SSH for the management port (eth0). For details, see the ACM online Help.

3. Use SCP or FTP to copy the software image to the IDP Series device. The IDP Series device does not run an FTP server, so you have to initiate the FTP session from the IDP Series device.
4. Run the upgrade script by entering **sh sensor\_version.sh**, where *version* is the number that identifies the software release version. When the script has finished, enter **reboot**.
5. In the NSM Device Manager, right-click the device, select **Adjust OS Version**, and complete the wizard steps.
6. Download the IDP detector engine and NSM attack database updates to the NSM GUI server:

From the NSM main menu, select **Tools > View/Update NSM attack database** and complete the wizard steps.

7. Push the updated IDP detector engine to IDP Series devices:

From the NSM main menu, select **Devices > IDP Detector Engine > Load IDP Detector Engine for ScreenOS** and complete the wizard steps.



**NOTE:** Updating the IDP detector engine on a device does not require a reboot of the device.

8. Push a security policy update job to update attack objects in use in your security policy:
  - a. In NSM, select **Devices > Configuration > Update Device Config**.
  - b. Select devices to which to push the updates and set update job options.
  - c. Click **OK**.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)
- [Pushing Security Policy Updates to an IDP Series Device \(NSM Procedure\) on page 340](#)

---

## Updating IDP OS Software (NSM Procedure)

This topic provides the basic procedure for upgrading IDP OS software. Check the [release notes](#) to see the exact procedure tested for a particular release.

To update IDP OS software:

1. Add the IDP OS software to the NSM GUI server.
2. Push the IDP OS software from the NSM GUI server to one or more IDP Series devices.

To add an IDP OS software image to the NSM GUI server:

1. Download the software image:
  - a. Go to <https://www.juniper.net/customers/csc/software/> and log in with your customer username and password.
  - b. Enter the IDP Series device serial number to display a view of applicable software releases available for download.
  - c. Click the applicable link to display the software download page.
  - d. Download the software to a location you can access from your NSM client.
2. From the NSM main menu, select **Tools > Software Manager** to display the Software Manager dialog box.
3. Click the + button to display the Open dialog box.
4. Select the IDP OS software image you just downloaded and click **Open** to add the software image to the NSM GUI server.
5. Click **OK**.

To push the software image from the NSM GUI server to IDP Series devices:

1. From the NSM main menu, select **Devices > Software > Install Device Software** to display the Install Device Software dialog box.
2. From the Select OS Name list, select **ScreenOS/IDP**.
3. From the Select Software Image list, select the image file you just added to the NSM GUI server.
4. In the Select Devices list, select the IDP Series devices on which to install the software update.
5. Click **Next** and complete the wizard steps.
6. Select **Automate ADM Transformation** to automatically update the Abstract Data Model (ADM) for the device after NSM installs the update.



**NOTE:** If you clear this setting, the update is installed onto the device, but you cannot manage the device from NSM until the device ADM is updated.

7. Click **Finish** to display upgrade status in the Job Information dialog box.
8. When the upgrade finishes, click **Close** to exit the Job Information dialog box.
9. Download the IDP detector engine and NSM attack database updates to the NSM GUI server:
 

In NSM, select **Tools > View/Update NSM attack database** and complete the wizard steps.
10. Push the updated IDP detector engine to IDP Series devices:

In NSM, select **Devices > IDP Detector Engine > Load IDP Detector Engine** and complete the wizard steps.



**NOTE:** Updating the IDP detector engine on a device does not require a reboot of the device.

11. Push a security policy update job to update attack objects in use in your security policy:
  - a. In NSM, select **Devices > Configuration > Update Device Config**.
  - b. Select devices to which to push the updates and set update job options.
  - c. Click **OK**.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Upgrading Software \(CLI Procedure\) on page 389](#)
- [Loading J-Security Center Updates \(NSM Procedure\) on page 336](#)

---

## Loading J-Security Center Updates (NSM Procedure)

The Juniper Networks Security Center (J-Security Center) routinely makes important updates available to IDP security policy components:

- **Detector engine.** The IDP detector engine is a dynamic protocol decoder that includes support for decoding more than 60 protocols and more than 500 service contexts. You should update IDP detector engine when you first install IDP, whenever you upgrade, and whenever alerted to do so by Juniper Networks. You can view release notes for detector engine updates at <http://www.juniper.net/techpubs/software/management/idp/de/>.
- **Attack database.** The [attack signature database](#) stores data definitions for attack objects. Attack objects are patterns comprising stateful signatures and traffic anomalies. You specify attack objects in IDP rulebase rules.
- **Application signature database.** The [application signature database](#) stores data definitions for application objects. Application objects are patterns used to identify applications and match APE rulebase rules.

J-Security Center updates are packaged and released separately from the IDP operating system and software code base to ensure IDP products protect your network against recently discovered vulnerabilities. We recommend you schedule automatic updates for the attack database and application database. For IDP Series devices, both databases are distributed in "signature database updates".

After you have completed the update, any new attack objects and application objects are available in the security policy editor. If you use dynamic groups in IDP rulebase rules and a new attack object belongs to the dynamic group, the rule automatically inherits the new attacks.



**NOTE:** We recommend you subscribe to the IDP Signature Updates technical bulletin to be notified when J-Security Center releases IDP detector engine updates. Go to <https://www.juniper.net/alerts/>.

Table 92 on page 337 provides procedures for updating the IDP detector engine and the NSM attack database.

**Table 105: IDP Detector Engine and NSM Attack Database Update Procedures**

Task	Procedure
To view version information for the installed IDP detector engine	In the NSM Device Manager, double-click the IDP Series device to display the IDP Series device configuration editor. The Info node displays version information, including the IDP detector engine version.
To update the IDP detector engine	<p>Updating the IDP detector engine is a three part process.</p> <p>To update IDP detector engine:</p> <ol style="list-style-type: none"> <li>Download IDP detector engine and NSM attack database updates to the NSM GUI server: In NSM, select <b>Tools &gt; View/Update NSM attack database</b> and complete the wizard steps.</li> <li>Push the updated IDP detector engine to IDP Series devices: In NSM, select <b>Devices &gt; IDP Detector Engine &gt; Load IDP Detector Engine</b> and complete the wizard steps.</li> </ol> <p><b>NOTE:</b> Updating the IDP detector engine on a device does not require a reboot of the device.</p> <ol style="list-style-type: none"> <li>Run a security policy update job to initialize the IDP detector engine update: <ol style="list-style-type: none"> <li>In NSM, select <b>Devices &gt; Configuration &gt; Update Device Config</b>.</li> <li>Select devices to which to push the updates and set update job options.</li> <li>Click <b>OK</b>.</li> </ol> </li> </ol>

**Table 105: IDP Detector Engine and NSM Attack Database Update Procedures (*continued*)**

Task	Procedure
To update predefined attack objects and application objects	<p>Updating attack objects is a two-part process.</p> <p>To update predefined attack objects:</p> <ol style="list-style-type: none"><li>1. Download NSM attack database updates to the NSM GUI server: From the NSM main menu, select <b>Tools &gt; View/Update NSM attack database</b> and complete the wizard steps.</li><li>2. Push the updates to IDP Series devices:<ol style="list-style-type: none"><li>a. From the NSM main menu, select <b>Devices &gt; Configuration &gt; Update Device Config</b>.</li><li>b. Select devices to receive pushed updates and set update job options.</li><li>c. Click <b>OK</b>.</li></ol></li></ol> <p><b>NOTE:</b> Only the attack objects that are used in IDP rules for the device are pushed from the GUI server to the device.</p>

---

Table 105: IDP Detector Engine and NSM Attack Database Update Procedures (*continued*)

Task	Procedure
To schedule regular updates	<ol style="list-style-type: none"> <li>1. Log in to the NSM GUI server command line.</li> <li>2. Change directory to <code>/usr/netscreen/GuiSvr/utls</code>.</li> <li>3. Create a shell script called <b>attackupdates.sh</b> with the following contents: <ul style="list-style-type: none"> <li>• Set the NSMUSER environment variable with an NSM domain/user pair. The command for setting environment variables depends on your OS. For example: <pre>export NSMUSER=domain/user</pre> </li> <li>• Set the NSMPASSWD environment variable with an NSM password. The command for setting environment variables depends on your OS and shell. For example: <pre>export NSMPASSWD=password</pre> </li> <li>• Specify a <b>guiSvrCli.sh</b> command string. For example: <pre>/usr/netscreen/GuiSvr/utls/guiSvrCli.sh --update-attacks --post-action --update-devices --skip</pre> </li> </ul> </li> <li>4. Make the script executable by the user associated with the cron job: <pre>chmod 700 attackupdates.sh</pre> </li> <li>5. Run the crontab editor: <pre>crontab -e</pre> </li> <li>6. Add an entry for the shell script: <pre>minutes_after_hour hour * * * /usr/netscreen/GuiSvr/utls/attackupdates.sh</pre> </li> </ol> <p>During the update, the <b>guiSvrCli</b> utility updates the attack object database, then performs the post actions. After updating and executing actions, the system generates an exit status code of 0 (no errors) or 1 (errors).</p> <p><b>NOTE:</b> For information on connecting to the NSM command line, see the NSM documentation.</p>

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Attack Objects Task Summary on page 246](#)
- [Application Objects Task Summary on page 286](#)
- [Pushing Security Policy Updates to an IDP Series Device \(NSM Procedure\) on page 340](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Attack Objects on page 60](#)
- [Using Application Objects on page 73](#)





## Installing Traffic Interface I/O Modules

- [Installing an I/O Module on page 397](#)

### Installing an I/O Module

---

I/O modules contain traffic interfaces of the type and capacity required for your specific deployment.

The following figures show the faceplate of an I/O module blank and the faceplates of IDP Series I/O modules. The procedure following the figures describes how to install an I/O module.

Figure 132: I/O Module Blank



Figure 133: IDP-1GE-4COP-BYP



Figure 134: IDP-1GE-4SFP\*



\* requires a transceiver (sold separately)

Figure 135: IDP-1GE-4SX-BYP



Figure 136: IDP-10GE-2XFP\*



\* requires a transceiver (sold separately)

Figure 137: IDP-10GE-2SR-BYP



The following procedure assumes you have already installed the IDP Series appliance in your equipment rack.



**CAUTION:** Carefully observe safety guidelines provided in the *Juniper Networks Safety Guide*.

To install the I/O module hardware:

1. Power off the device. Verify that the power LED is off.
2. Unscrew the blank or replaceable module tray and remove it from the chassis.
3. Carefully replace the blank or replaceable module tray with the new I/O module tray. Slide the I/O module tray into the I/O module slot.

**Figure 138: Replacing a Blank Tray with an I/O Module Tray**



4. Tighten the screws on each side of the faceplate.
5. Power on the device. Verify that the power LED is a steady green light.
6. Use ACM to reconfigure device settings to incorporate the new traffic interfaces. You need to configure speed/duplex settings and assign the new interfaces virtual routers.
7. Connect the traffic interfaces to your network.
8. Verify traffic flow.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Connecting to ACM on page 191](#)
- [Configuring Virtual Routers \(ACM Procedure\) on page 192](#)
- [Configuring Interface Aliasing \(ACM Procedure\) on page 296](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Network Interfaces Overview on page 11](#)



# Enabling Bypass and Peer Port Modulation

- [Configuring Virtual Routers \(ACM Procedure\)](#) on page 401
- [Enabling the Flow Bypass Feature](#) on page 402

## Configuring Virtual Routers (ACM Procedure)

A virtual router is a logical pair of traffic interfaces that handles traffic into and out of the IDP Series device. You use the ACM Configure Virtual Routers page to configure the features of the virtual routers, including deployment mode and bypass options. For background information on these features, see the *IDP Series Concepts and Examples Guide*. [Figure 75 on page 193](#) shows the ACM Configure Virtual Routers page.

Figure 139: ACM Configure Virtual Routers Page

**Configure Virtual Routers**

---

In this step, you must configure the interfaces that the IDP Sensor will use to handle traffic.

For each pair of interfaces, select the mode you want each pair to run in.

Active?	Interfaces	Virtual Router	Mode	NIC State (after system unavailability)	NIC State (after graceful shutdown)
<input checked="" type="checkbox"/>	eth2,eth3	vr0	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth4,eth5	vr1	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth6,eth7	vr2	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth8,eth9	vr3	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>
<input checked="" type="checkbox"/>	eth10,eth11	vr4	<input checked="" type="radio"/> Transparent <input type="radio"/> Sniffer	NICs off <input type="button" value="v"/>	NICs off <input type="button" value="v"/>

☒ Enable layer2 bypass  
☐ Enable Peer Port Modulator

Fallover timeout value:

To configure virtual routers:

1. Connect to ACM.
2. From the main menu, click **Reconfigure Virtual Routers**.

3. On the Configure Virtual Routers page:

- Select the box in the Active? column to enable the virtual routers you plan to connect to your network.
- Specify a deployment mode: transparent or sniffer.
- For transparent mode virtual routers, specify how you want to handle failure or shutdown: internal bypass, external bypass, or NICs off.
- Specify whether you want to enable Layer 2 bypass.
- Specify whether you want to enable peer port modulation.
- Specify a failover timeout value.

For details, see the ACM online Help.

4. Click **Next** to advance the wizard until you reach the Brief Configuration Report page.
5. Review, save, and apply your configuration changes.

**Related  
Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Connecting to ACM on page 191](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Network Interfaces Overview on page 11](#)

---

## Enabling the Flow Bypass Feature

The flow bypass feature prevents the IDP Series device from becoming a point of failure when the network is congested. With flow bypass enabled, when the IDP system packet receive queue reaches a rising threshold that you specify, the IDP engine marks the flow as a bypass flow and passes it through, uninspected. The IDP Series device passes through subsequent flows until the IDP system packet receive queue falls below a reset threshold that you also specify.

The flow bypass feature is not enabled by default.

For an overview of the flow bypass feature, see the *IDP Series Concepts and Examples Guide*.

To enable the flow bypass feature:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to enable flow bypass:

```
[root@default host admin]# scio const -s s0:flow set sc_flow_bypass_enable 1
[root@default host admin]#
```

By default, the system packet queue size utilization rising threshold is 90%; the reset threshold is 80%.

- Optional. Change the rising threshold with the following command syntax:

```
scio const -s s0:flow set sc_flow_bypass_threshold_hi percent
```

For example:

```
[root@defaulthost admin]# scio const -s s0:flow set sc_flow_bypass_threshold_hi 95
scio: setting sc_flow_bypass_threshold_hi to 0x5f
[root@defaulthost admin]#
```

- Optional. Change the reset threshold with the following command syntax:

```
scio const -s s0:flow set sc_flow_bypass_threshold_low percent
```

For example:

```
[root@defaulthost admin]# scio const -s s0:flow set sc_flow_bypass_threshold_low 85
scio: setting sc_flow_bypass_threshold_low to 0x55
[root@defaulthost admin]#
```

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

- Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
- Add the constant below the line `user_start_pre_policy ()`. For example:

```
user_start_pre_policy ()

{

    # Disable ARP spoofing detection
    # -----
    # If you are running clusters with virtual MAC addresses, IDP will treat
    # these as spoofed ARP packets since the MAC addresses in the ethernet
    # frame will be different from what is inside the ARP request/response. If
    # you have multiple virtual routers, you need to perform this operation on
    # all defined virtual routers.
    #
    # $SCIO const -v vr0 set sc_arp_spoof_detect 0
    # $SCIO const -s s0 set sc_mpls_decapsulation 1
    $SCIO const -s s0:flow set sc_flow_bypass_enable 1
    return;

}
```

- Save the file.
- Restart the IDP engine:

```
[root@defaulthost admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying the Flow Bypass Feature on page 495](#)
- [scio const on page 505](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)



# Configuring the Management Interface

- [Changing the Management Interface IP Address on page 405](#)

## Changing the Management Interface IP Address

---

The IDP system uses the management interface (eth0) to communicate with Network and Security Manager (NSM) (all deployments), SNMP managers (if applicable), and Juniper IVE devices (Access Control Service or Secure Access Service in coordinated threat control deployments).

The management interface IP address is the only IP address for the IDP Series device. It is the IP address used to make ssh connections to the command-line interface (CLI) and to make browser-based connections to the Appliance Configuration Manager (ACM).

You configure the management interface IP address when you install and deploy the device, as described in the [installation documentation](#) for your IDP Series device. If your network IP address space changes, and you are required to reassign the IP address for management interface, you can use ACM to change it.

Before you begin, prepare for the impact that changing the management interface IP address has on any systems that might be configured to listen to or communicate with the IDP Series device through the IDP Service device management interface IP address, such as:

- NSM
- SNMP
- Juniper IVE systems



**NOTE:** The IP address change operation restarts the IDP engine. Be sure to use the same precautions you use when upgrading software, such as high-availability (HA) features, NIC bypass features, and choosing an opportune time to perform the operation (for example, a period when traffic is light).



**TIP:** Be sure to inform others who might act as administrators for the device that the IP address has changed. Administrators use the IP address when connecting to the device through ssh or ACM.

To change the management interface IP address:

1. Reconfigure the IDP Series device using ACM:
  - a. Log into ACM.
  - b. Use the Configure Management Interface page to configure settings for the management interface. The IP address must be reachable from the NSM Device Server.

After the management IP address is changed:

- The ACM Web server is rebooted. You can access ACM again using the new IP address.
  - The IDP engine is restarted, and traffic flow is processed according to NIC bypass settings during the restart cycle.
  - The device reconnects to NSM and logs are forwarded to NSM without disruption.
  - In NSM Device Monitor, the status displayed in the Connection Status column goes down and comes up. There are no changes to the Configuration Status column.
  - In NSM Log Viewer, NSM displays the same IDP device name with the previous IP address.
2. Update the NSM configuration for the IDP Series device:
    - a. In NSM Device Manager, right-click the IDP Series device and select RMA to change the device status to RMA.
    - b. Right-click the device and select **Activate Device** to display the Activate Device dialog box.
    - c. Select the **Device deployed and IP is reachable** option, and complete the configuration by specifying the IDP Series device IP address, admin username and password, and root password.

After the device has been reactivated:

- In NSM Device Monitor, you might see the connection status go down and up again as the connection is re-established.
- In NSM Log Viewer, the logs collected before the RMA/Activate Device operation show the same device name but the changed IP address.

**Related  
Documentation**

- [IDP Series Network Interfaces Overview on page 11](#)

- [Connecting to ACM on page 191](#)
- [Activating Devices \(NSM Procedure\) on page 348](#)
- [Supported Tools for Management Tasks on page 189](#)



## PART 5

# Monitoring

- [Overview on page 411](#)
- [Using SNMP on page 417](#)
- [Using NSM Logs and Reports on page 447](#)
- [Packet Logging on page 475](#)
- [Using the bypassStatus Utility to Monitor the Internal Bypass Daemon on page 483](#)
- [Using the sctop Utility to Monitor Session Flow on page 487](#)
- [Using the scio Utility to Verify Feature Implementation on page 491](#)
- [scio Commands on page 499](#)
- [IDP MIB Object ID Reference on page 553](#)



## CHAPTER 39

# Overview

- [Supported Tools for Monitoring Tasks on page 411](#)
- [Developing a Logging Strategy on page 412](#)
- [Developing a Log Storage Strategy on page 412](#)

### Supported Tools for Monitoring Tasks

[Table 106 on page 411](#) identifies supported tools for IDP Series monitoring tasks.

**Table 106: Supported Tools for Monitoring Tasks**

Task	CLI	SNMP	IDP Reporter	NSM
Monitoring the status of the operating system.	Yes	Yes	No	Yes
Monitoring the status of network interfaces.	Yes	Yes	No	Yes
Monitoring the status of IDP engines.	Yes	Yes	No	No
Monitoring actual CPU usage of IDP engines.	Yes	Yes	No	No
Analyzing attack logs and reports.	No	No	Yes	Yes
Analyzing application usage logs and reports.	No	No	Yes	Yes
Viewing flow statistics.	Yes	Yes	No	No
Using debugging counters.	Yes	No	No	No
Packet capture	jnetTcpdump, tcpdump	No	No	Policy-based

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)
- [IDP Reporter Task Summary](#)
- [scio Monitoring Commands Task Summary on page 491](#)

- [bypassStatus Utility Task Summary on page 483](#)
- [sctop Task Summary on page 487](#)

---

## Developing a Logging Strategy

Intrusion prevention systems can generate hundreds of logs per hour. In order to make the best use of the security logs, you should develop strategic approaches to the following administrative tasks:

- Fine-tuning the security policy rules.

Security policy rules determine the amount of logging performed by the IDP Series device, as well as automatic actions to take on offending traffic, such as dropping the session, sending a TCP reset, blocking the IP address from future connections, and so forth. See [“Example: Fine-Tuning a Security Policy” on page 48](#).

- Analyzing log event summaries and packet capture data.

By viewing log summaries, attack reference information, and packet data, you can verify whether the severity and actions associated with a security event are appropriate, whether refinements to your security policy are required, and whether further response actions are warranted. See [“Example: Using NSM Log Viewer Features” on page 139](#).

- Managing log and packet storage.

Your business log management and log storage policies determine where you store IDP Series device logs and security event logs. Your IDP Series device supports local logging, central collection by NSM, and forwarding to a syslog server. See [“Developing a Log Storage Strategy” on page 332](#).

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

---

## Developing a Log Storage Strategy

This topic summarizes IDP log storage and log forwarding options so you can develop a log storage strategy suitable for your business. It includes the following sections:

- [Log Management Considerations on page 413](#)
- [Local Log Files and Directories on page 413](#)
- [NSM Log Collection on page 414](#)



## Log Management Considerations

An IDP Series device might generate hundreds of logs per day. Your log storage strategy depends on a number of factors:

- The nature of your business. Compliance with regulations or business agreements might determine where you collect logs or how often you retain them.
- Existing log management infrastructure. We recommend you become familiar with an use Network and Security Manager (NSM) as a central location for log analysis, but your previous investments in technology and training are also strong considerations.
- Distribution to the appropriate personnel for analysis is also a key consideration.

If your organization has not formalized a log management policy, consult the National Institute of Standards and Technology (NIST) publication, [Guide to Computer Security Log Management](#), for a treatment of the myriad considerations.

## Local Log Files and Directories

Logs are stored locally on the device in subdirectories of `/usr/idp/device/var`. Log pruning occurs when a disk partition reaches 90% capacity.

Table 107: IDP Local Log Directories

Directory	Content
<code>/usr/idp/device/var/logs</code>	Local storage for device and security event logs before they are forwarded to NSM.
<code>/usr/idp/device/var/pktlogs</code>	Local storage for packet capture logs before they are forwarded to NSM.
<code>/usr/idp/device/var/profile</code>	Local storage for Profiler database logs before they are forwarded to NSM.
<code>/usr/idp/device/var/sysinfo/logs</code>	Location where system messages are written.
<code>/usr/idp/device/var/stat/</code>	Local storage for application volume tracking logs before they are forwarded to NSM, IDP Reporter, or Application Usage Manager.



**NOTE:** Although `/usr/idp/device/var` is a symbolic link to `/var/idp/device/var`, user scripts or programs created to manage files should reference the `/usr/idp/device/var` path.

By default, logs are forwarded to NSM, which is the primary user interface for the IDP Series device.

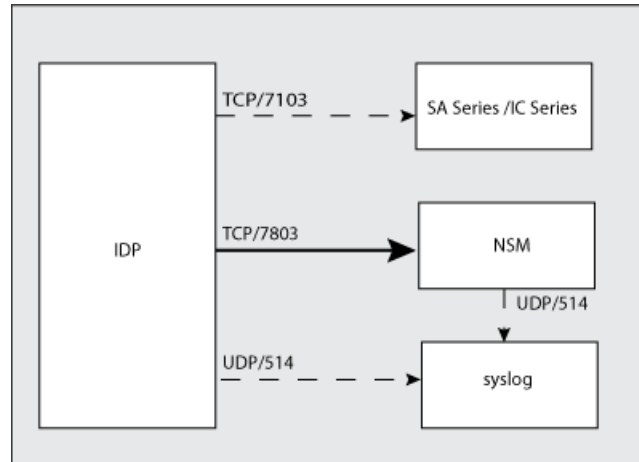
Optionally, you can configure the IDP Series device to send copies of logs to external devices, such as:

- A syslog server, including a Juniper Networks Security Threat Response Manager (STRM) device, which reads the IDP syslog format.

- A Juniper Networks Secure Access Series or Infranet Controller Series device to inform access policies.

Figure 126 on page 333 provides a visual summary of your log forwarding options. The solid line indicates default behavior. The dashed lines indicate options you must configure to use.

Figure 140: IDP Log Storage and Log Forwarding



**NOTE:** In IDP OS Release 5.1, syslog protocol port are configurable. However, we recommend you use the standard protocol and port whenever feasible.

## NSM Log Collection

By default, the IDP Series device sends logs to NSM where they can be displayed and analyzed with the NSM user interface. We recommend you become familiar with an use NSM as a central location for log analysis. Logs are stored on the NSM Device Server in subdirectories of `/usr/netscreen/DevSvr/var/logs`. NSM supports the following log management features:

- Command-line utilities to archive, copy, and purge logs.
- Configurable time retention policies that trigger pruning.
- Automated log management jobs based on criteria you configure, including severity, category, and so forth.
- Support for log field filters in export operations to XML, CSV, syslog, SNMP, e-mail, or script.

For complete information on NSM log management features, see Chapter 19 of the [NSM Administration Guide](#).

### Related Documentation

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)
- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)



## CHAPTER 40

# Using SNMP

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)
- [Configuring SNMP Reporting for the Interface Category of Statistics on page 418](#)
- [Configuring SNMP Reporting for the Resource Category of Statistics on page 428](#)
- [Configuring SNMP Reporting for the Rule Category of Statistics on page 433](#)
- [Configuring SNMP Reporting for the Sensor Category of Statistics on page 435](#)
- [Configuring SNMP Reporting for the Traffic Category of Statistics on page 438](#)

### SNMP Statistic Reporting and Traps Task Summary

IDP OS Release 5.1 supports extensive system resource instrumentation, so you can use SNMP utilities to monitor device health and load. [Table 108 on page 417](#) summarizes the categories of statistics you can monitor.

**Table 108: System Resource Instrumentation Categories**

Category	Guidance	SNMP MIB	SNMP Traps
Sensor: control plane CPU, memory, and disk usage statistics	Monitor the health of the device in broad terms.	Enabled (default)	Enabled (default)
Resource: IDP engine CPU utilization, session count per protocol, and session create rate	Monitor resource utilization to understand the extent to which new attack objects or applications affect performance.	Enabled (default)	Enabled (default)
Traffic: count and rate reporting for throughput and packet drops; count and rate of logs	Monitor traffic to understand traffic volume and causes of packet drops.	Enabled (default)	-
Rule: count for security policy rules and attack object matches	We recommend you use NSM to analyze security statistics. We support SNMP reporting for cases where access to NSM is not immediately available.	Enabled (default)	-
Interface: receive, transmit, and drop rates for interfaces; free packet buffer space; NIC status	Monitor interface statistics and free packet buffer space to understand load at receive and transmit queues.	Enabled (default)	Enabled (default)

Administration with system resource instrumentation features includes the following tasks:

- Enabling SNMP and configuring communication with your SNMP manager.
- Enabling or disabling statistic reporting by category. Reporting consumes resources. Disable reporting for statistics you do not use or if necessary to work around performance issues.
- Enabling or disabling trap reporting by category. Disable traps if you do not use them.
- Tuning trap thresholds. If desired, change the default values to thresholds.
- Using an SNMP manager or utility to view MIB data and trap notifications.

#### **Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring an SNMP Agent \(NSM Procedure\) on page 299](#)
- [Configuring SNMP Reporting for the Sensor Category of Statistics on page 435](#)
- [Configuring SNMP Reporting for the Resource Category of Statistics on page 428](#)
- [Configuring SNMP Reporting for the Interface Category of Statistics on page 418](#)
- [Configuring SNMP Reporting for the Traffic Category of Statistics on page 438](#)
- [Configuring SNMP Reporting for the Rule Category of Statistics on page 433](#)
- [Troubleshooting SNMP Statistic Reporting on page 585](#)
- [scio sri on page 529](#)
- [IDP Series MIB Object ID Reference on page 553](#)
- [Example: Querying the IDP Series Device MIB on page 151](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

---

## **Configuring SNMP Reporting for the Interface Category of Statistics**

IDP OS Release 5.1 supports extensive system resource instrumentation so you can use SNMP to monitor device health. The system resource instrumentation is classified in five categories: sensor, resource, traffic, rule, and interface. The interface category includes detailed statistics about traffic received, transmitted, and dropped by the IDP Series device traffic interfaces. You use the interface category of statistics to understand load at receive and transmit queues. You can turn interface statistics on or off and configure thresholds that trigger SNMP traps.

To configure the interface statistics reporting and traps:

1. Log in to the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the **list** command to display the current settings:

```
[root@defaulthost ~]# scio sri interface list
SRI interface :
sc_enable_interface_stats           = 1      [ 0 - 1 ]
sc_enable_interface_traps           = 1      [ 0 - 1 ]
thrshld_rx_pkt_drop_rate_per_if     = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_per_if          = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_all_if          = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_rate_all_if     = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_ovflow          = 80     [ 0 - 100 ]
thrshld_tx_pkt_drop_per_if          = 80     [ 0 - 100 ]
thrshld_tx_pkt_drop_all_if          = 80     [ 0 - 100 ]
thrshld_free_pkt_bufs               = 5      [ 0 - 100 ]
```

By default, statistic reporting and traps are enabled (value 1). The default threshold that triggers traps on drop statistics is 80%. The trap for free packet-buffer space is triggered when free space is less than 5%.

- (Optional) Use the **set** command to change the defaults. For example, the following commands show how to change the threshold for sending traps when the free packet-buffer space is less than 10%.

```
[root@defaulthost ~]# scio sri interface set thrshld_rx_pkt_drop_rate_per_if 10
Setting variable thrshld_idp_cpu successful
```

```
[root@defaulthost ~]# scio sri interface list
SRI interface :
sc_enable_interface_stats           = 1      [ 0 - 1 ]
sc_enable_interface_traps           = 1      [ 0 - 1 ]
thrshld_rx_pkt_drop_rate_per_if     = 10     [ 0 - 100 ]
thrshld_rx_pkt_drop_per_if          = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_all_if          = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_rate_all_if     = 80     [ 0 - 100 ]
thrshld_rx_pkt_drop_ovflow          = 80     [ 0 - 100 ]
thrshld_tx_pkt_drop_per_if          = 80     [ 0 - 100 ]
thrshld_tx_pkt_drop_all_if          = 80     [ 0 - 100 ]
thrshld_free_pkt_bufs               = 10     [ 0 - 100 ]
```



**TIP:** If you know you want to enable statistics for all system resource instrumentation categories, use the following command: **scio sri all set sc\_enable\_all\_stats 1**. If you know you want to enable traps for all categories, use the following command: **scio sri all set \_sc\_enable\_all\_traps 1**.

The following tables describe the device statistics reported to the IDP Series device MIB when interface statistics are enabled.



**NOTE:** Interface statistics use the same counters as **ifconfig** commands. Counters are reset when the IDP Series device is rebooted.

#### Packets Received Per Interface (Count)

Name	jnxIdpSensorPktsRxdPerIntfcTable
------	----------------------------------

OID	1.3.6.1.4.1.2636.3.9.1.49
-----	---------------------------

## Packets Received Per Interface (Count)

**Description** Packets received per interface (count).

**Example** [host]# **snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsRxdPerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorPktsRxdPerIntfcTable

jnxIdpSensorIFTable1Index	jnxIdpSensorIntfcName	jnxIdpSensorNoOfPkts
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth11"	0
7	"eth10"	0
8	"eth3"	27207
9	"eth2"	44484
10	"eth5"	0
11	"eth4"	0

## Packets Received Per Interface (Rate)

**Name** jnxIdpSensorPktsRxRatePerIntfcTable

**OID** 1.3.6.1.4.1.2636.3.9.1.50

**Description** Packets received per interface (rate).

**Example** [host]# **snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsRxRatePerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorPktsRxRatePerIntfcTable

jnxIdpSensorIFTable2Index	jnxIdpSensorPktsRxRateIntfcName	jnxIdpSensorPktsRxdPerSec
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth11"	0
7	"eth10"	0
8	"eth3"	1
9	"eth2"	0
10	"eth5"	0
11	"eth4"	0

## Packets Dropped Per Receiving Interface (Count)

**Name** jnxIdpSensorRxPktsDropPerIntfcTable

**OID** 1.3.6.1.4.1.2636.3.9.1.51

**Description** Packets dropped at the interface (count).



## Packets Dropped Per Receiving Interface (Count)

**Example** [host]# **snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorRxPktsDropPerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorRxPktsDropPerIntfcTable

jnxIdpSensorIFTable3Index	jnxIdpSensorRxIntfcName	jnxIdpSensorRxPktsDropCount
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth11"	0
7	"eth10"	0
8	"eth3"	0
9	"eth2"	0
10	"eth5"	0
11	"eth4"	0

## Packets Dropped Per Receiving Interface (Rate)

**Name** jnxIdpSensorRxPktsDropRatePerIntfcTable

**OID** 1.3.6.1.4.1.2636.3.9.1.52

**Description** Packets dropped at the interface (rate).

**Example** [host]# **snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorRxPktsDropRatePerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorRxPktsDropRatePerIntfcTable

jnxIdpSensorIFTable4Index	jnxIdpSensorRxPktsDropRateIntfcName	jnxIdpSensorRxPktsDropRate
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth11"	0
7	"eth10"	0
8	"eth3"	0
9	"eth2"	0
10	"eth5"	0
11	"eth4"	0

## Total Packets Received (Count)

**Name** jnxIdpSensorPktsRxdOnAllIntfc

## Total Packets Received (Count)

**OID** 1.3.6.1.4.1.2636.3.9.1.53.0

**Description** Sum of packets received on all interfaces (count).

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsRxdOnAllIntfc.0**  
JUNIPER-IDP-MIB::jnxIdpSensorPktsRxdOnAllIntfc.0 = INTEGER: 71753

## Total Packets Dropped (Count)

**Name** jnxIdpSensorPktsDropOnAllIntfc

**OID** 1.3.6.1.4.1.2636.3.9.1.54.0

**Description** Sum of packets dropped on all interfaces (count).

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDropOnAllIntfc.0**  
JUNIPER-IDP-MIB::jnxIdpSensorPktsDropOnAllIntfc.0 = INTEGER: 0

## Total Packets Dropped (Rate)

**Name** jnxIdpSensorPktsDropRateOnAllIntfc

**OID** 1.3.6.1.4.1.2636.3.9.1.55.0

**Description** Rate of packets per second (pps) dropped.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDropRateOnAllIntfc.0**  
JUNIPER-IDP-MIB::jnxIdpSensorPktsDropRateOnAllIntfc.0 = INTEGER: 0

## Packets Dropped Due Overflow at Receiving Interface (Count)

**Name** jnxIdpSensorPktsDropDueToRxOverflowTable

**OID** 1.3.6.1.4.1.2636.3.9.1.56

**Description** Packets dropped because of overflow at the receiving interface (count).

**Example** [host]# **snmptable -v2c -c public 10.209.96.78**  
**JUNIPER-IDP-MIB::jnxIdpSensorPktsDropDueToRxOverflowTable**  
JUNIPER-IDP-MIB::jnxIdpSensorPktsDropDueToRxOverflowTable: No entries

## Packets Transmitted Per Interface (Count)

**Name** jnxIdpSensorPktsTxdPerIntfcTable

**OID** 1.3.6.1.4.1.2636.3.9.1.57

## Packets Transmitted Per Interface (Count)

**Description**   Packets transmitted per interface (count).

**Example**       [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsTxdPerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorPktsTxdPerIntfcTable

jnxIdpSensorIFTable5Index	jnxIdpSensorTxIntfcName	jnxIdpSensorNoOfPktsTxd
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth11"	0
7	"eth10"	0
8	"eth3"	44555
9	"eth2"	27236
10	"eth5"	0
11	"eth4"	0

## Packets Transmitted Per Interface (Rate)

**Name**       jnxIdpSensorPktsTxRatePerIntfcTable

**OID**       1.3.6.1.4.1.2636.3.9.1.58

**Description**   Packets transmitted per interface (rate).

**Example**       [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsTxRatePerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorPktsTxRatePerIntfcTable

jnxIdpSensorIFTable8Index	jnxIdpSensorPktsTxRateIntfcName	jnxIdpSensorPktsTxdPerSec
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth11"	0
7	"eth10"	0
8	"eth3"	0
9	"eth2"	0
10	"eth5"	0
11	"eth4"	0

## Packets Dropped at Transmitting Interface (Count)

**Name**       jnxIdpSensorTxPktsDropPerIntfcTable

**OID**       1.3.6.1.4.1.2636.3.9.1.59

**Description**   Packets dropped at the transmitting interface (count).

## Packets Dropped at Transmitting Interface (Count)

**Example** [host]# **snmptable -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropPerIntfcTable**  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropPerIntfcTable

jnxIdpSensorIFTable6Index	jnxIdpSensorTxdIntfcName	jnxIdpSensorTxPktsDropped
1	"eth1"	0
2	"eth7"	0
3	"eth6"	0
4	"eth9"	0
5	"eth8"	0
6	"eth3"	0
7	"eth2"	0
8	"eth5"	0
9	"eth4"	0

## Total Packets Transmitted (Count)

**Name** jnxIdpSensorTxPktsOnAllIntfc

**OID** 1.3.6.1.4.1.2636.3.9.1.60.0

**Description** Sum of packets transmitted by all transit interfaces (count).

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTxPktsOnAllIntfc.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorTxPktsOnAllIntfc.0 = INTEGER: 72195

## Total Packets Dropped at the Transmitting Interface (Count)

**Name** jnxIdpSensorTxPktsDropOnAllIntfc

**OID** 1.3.6.1.4.1.2636.3.9.1.61.0

**Description** Sum of packets transmitted by all transit interfaces (count).

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropOnAllIntfc.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorTxPktsDropOnAllIntfc.0 = INTEGER: 0

## Network Interface Status (Up or Down)

**Name** jnxIdpSensorNICStatusTable

**OID** 1.3.6.1.4.1.2636.3.9.1.62

**Description** Network interface status (up or down).

## Network Interface Status (Up or Down)

**Example** [host]# `snmptable -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorNICStatusTable`  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorNICStatusTable

jnxIdpSensorIFTable7Index	jnxIdpSensorNICIntfcName	jnxIdpSensorNICStatus
1	"eth1"	"Down"
2	"eth7"	"Down"
3	"eth6"	"Down"
4	"eth9"	"Down"
5	"eth8"	"Down"
6	"eth3"	"Up"
7	"eth2"	"Up"
8	"eth5"	"Down"
9	"eth4"	"Down"

Table 109 on page 425 describes the device statistics reported to SNMP traps when interface traps are enabled.

**Table 109: System Resource Instrumentation: Network Interface Traps**

Object Name	Object ID	Description
jnxIdpFreePktLastFiveSecNotify	jnxIdpTrapsPrefix 17	Alarm is triggered when the count falls below the configured threshold.  Objects referenced: jnxIdpSensorFreePktBuffersFiveSec, jnxIdpSensorFreePktThreshold
jnxIdpFreePktLastFiveSecRestored	jnxIdpTrapsPrefix 18	Notification is triggered when the count is restored to above the alarm threshold.  Objects referenced: jnxIdpSensorFreePktBuffersFiveSec
jnxIdpFreePktLastOneMinNotify	jnxIdpTrapsPrefix 19	Alarm is triggered when the count falls below the configured threshold.  Objects referenced: jnxIdpSensorFreePktBuffersOneMin, jnxIdpSensorFreePktThreshold
jnxIdpFreePktLastOneMinRestored	jnxIdpTrapsPrefix 20	Notification is triggered when the count is restored to above the alarm threshold.  Objects referenced: jnxIdpSensorFreePktBuffersOneMin
jnxIdpTotalRxPktsDropNotify	jnxIdpTrapsPrefix 21	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktsDropOnAllIntfc, jnxIdpSensorPktsDropOnAllIntfcThreshold

Table 109: System Resource Instrumentation: Network Interface Traps (*continued*)

Object Name	Object ID	Description
jnxIdpTotalRxPktsDropRestored	jnxIdpTrapsPrefix 22	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktsDropOnAllIntfc
jnxIdpTotalRxPktsDropRateNotify	jnxIdpTrapsPrefix 23	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktsDropRateOnAllIntfc, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpTotalRxPktsDropRateRestored	jnxIdpTrapsPrefix 24	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktsDropRateOnAllIntfce
jnxIdpPerIfRxOverflowNotify	jnxIdpTrapsPrefix 25	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpPerIfRxOverflowRestored	jnxIdpTrapsPrefix 26	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount
jnxIdpPerIfRxDropNotify	jnxIdpTrapsPrefix 27	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpPerIfRxDropRestored	jnxIdpTrapsPrefix 28	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount
jnxIdpPerIfRxDropRateNotify	jnxIdpTrapsPrefix 29	Alarm is triggered when the rate exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold

Table 109: System Resource Instrumentation: Network Interface Traps (*continued*)

Object Name	Object ID	Description
jnxldpPerIfRxDropRateRestored	jnxldpTrapsPrefix 30	Notification is triggered when the rate falls below the alarm threshold.  Objects referenced: jnxldpSensorPktDropCount
jnxldpPerIfTxDropNotify	jnxldpTrapsPrefix 31	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxldpSensorPktDropCount, jnxldpSensorIfName, jnxldpSensorPktsDropRateOnAllIntfcThreshold
jnxldpPerIfTxDropRestored	jnxldpTrapsPrefix 32	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxldpSensorPktDropCount
jnxldpTotalTxDropNotify	jnxldpTrapsPrefix 33	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxldpSensorPktDropCount, jnxldpSensorIfName, jnxldpSensorPktsDropRateOnAllIntfcThreshold
jnxldpTotalTxDropRestored	jnxldpTrapsPrefix 34	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxldpSensorPktDropCount
jnxldpIntfcDownNotify	jnxldpTrapsPrefix 35	Alarm is triggered when status is down.  Objects referenced: jnxldpSensorIfStatus, jnxldpSensorIfName
jnxldpIntfcStatusRestored	jnxldpTrapsPrefix 36	Notification is triggered when status of a down interface is restored (up).  Objects referenced: jnxldpSensorIfStatus, jnxldpSensorIfName



**NOTE:** After a down trap is sent, the alarm is suspended for 1 minute. After this period, alarms can again be triggered.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)

- [IDP Series MIB Object ID Reference on page 553](#)

## Configuring SNMP Reporting for the Resource Category of Statistics

IDP OS Release 5.1 supports extensive system resource instrumentation, so you can use SNMP to monitor device health. The system resource instrumentation is classified in five categories: sensor, resource, traffic, rule, and interface. The resource category includes detailed CPU utilization, session rate, and session count statistics. You use the resource category statistics to understand the extent to which new attack objects or applications affect performance.

To configure the resource category of statistics reporting and traps:

1. Log in to the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the **list** command to display the current settings:

```
[root@defaulthost ~]# scio sri resource list
SRI resource :
  sc_enable_resource_stats          = 1      [ 0 - 1 ]
  sc_enable_resource_traps          = 1      [ 0 - 1 ]
  thrshld_idp_cpu                   = 80     [ 0 - 100 ]
  thrshld_session_rate              = 80     [ 0 - 100 ]
  thrshld_active_sessions           = 80     [ 0 - 100 ]
```

By default, statistic reporting and traps are enabled (value 1), and the threshold to trigger traps for IDP engine CPU utilization, session rate, and active session count is 80% (value 80).

3. (Optional) Use the **set** command to change the defaults. For example, the following commands show how to change the threshold for sending IDP engine CPU traps to 85% and verify your change:

```
[root@defaulthost ~]# scio sri resource set thrshld_idp_cpu 85
Setting variable thrshld_idp_cpu successful

[root@defaulthost ~]# scio sri resource list
SRI resource :
  sc_enable_resource_stats          = 1      [ 0 - 1 ]
  sc_enable_resource_traps          = 1      [ 0 - 1 ]
  thrshld_idp_cpu                   = 85     [ 0 - 100 ]
  thrshld_session_rate              = 80     [ 0 - 100 ]
  thrshld_active_sessions           = 80     [ 0 - 100 ]
```



**TIP:** If you know you want to enable statistics for all system resource instrumentation categories, use the following command: **scio sri all set sc\_enable\_all\_stats 1**. If you know you want to enable traps for all categories, use the following command: **scio sri all set \_sc\_enable\_all\_traps 1**.



The following tables describe the device statistics reported to the IDP Series device MIB when resource statistics are enabled.

#### IDP Engine CPU Utilization – Last 5 Seconds

<b>Name</b>	jnxIdpSensorCpuUtilFiveSecTable
<b>OID</b>	1.3.6.1.4.1.2636.3.9.1.12
<b>Description</b>	CPU utilization per IDP engine in the last 5 seconds (percent).
<b>Example</b>	<pre>[host]# snmptable -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorCpuUtilFiveSecTable SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorCpuUtilFiveSecTable  jnxIdpSensorFiveSecCpuID jnxIdpSensorFiveSecCpuUtilPercent 0 60</pre>

#### IDP Engine CPU Utilization – Last 1 Minute

<b>Name</b>	jnxIdpSensorCpuUtilOneMinTable
<b>OID</b>	1.3.6.1.4.1.2636.3.9.1.13
<b>Description</b>	CPU utilization per IDP engine in the last 1 minute (percent).
<b>Example</b>	<pre>[host]# snmptable -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorCpuUtilOneMinTable SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorCpuUtilOneMinTable  jnxIdpSensorOneMinCpuID jnxIdpSensorOneMinCpuUtilPercent 0 53</pre>

#### IDP Engine CPU Utilization – Last 5 Minutes

<b>Name</b>	jnxIdpSensorCpuUtilFiveMinTable
<b>OID</b>	1.3.6.1.4.1.2636.3.9.1.14
<b>Description</b>	CPU utilization per IDP engine in the last 5 minutes (percent).
<b>Example</b>	<pre>[host]# snmptable -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorCpuUtilFiveMinTable SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorCpuUtilFiveMinTable  jnxIdpSensorFiveMinCpuID jnxIdpSensorFiveMinCpuUtilPercent 0 53</pre>

#### Current Sessions

<b>Name</b>	jnxIdpSensorSessAllocated
<b>OID</b>	1.3.6.1.4.1.2636.3.9.1.3.0

### Current Sessions

**Description** Sessions currently allocated (count).

**Example** [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorSessAllocated.0**  
JUNIPER-IDP-MIB::jnxIdpSensorSessAllocated.0 = Gauge32: 0

### Session Capacity

**Name** jnxIdpSensorSessMaximum

**OID** 1.3.6.1.4.1.2636.3.9.1.4.0

**Description** Maximum sessions supported (static number).

**Example** [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorSessMaximum.0**  
JUNIPER-IDP-MIB::jnxIdpSensorSessMaximum.0 = INTEGER: 70000

### Session Create Rate

**Name** jnxIdpSensorSessnCreateRateFiveSec

**OID** 1.3.6.1.4.1.2636.3.9.1.15.0

**Description** Rate of connections created per second (cps). The value reported is an average per second over the last 5 seconds.

**Example** [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorSessnCreateRateFiveSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorSessnCreateRateFiveSec.0 = INTEGER: 0

### TCP Session Count

**Name** jnxIdpSensorTCPSessions

**OID** 1.3.6.1.4.1.2636.3.9.1.9.16.0

**Description** Active TCP sessions (count).

**Example** [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorTCPSessions.0**  
JUNIPER-IDP-MIB::jnxIdpSensorTCPSessions.0 = INTEGER: 0

### UDP Session Count

**Name** jnxIdpSensorUDPSessions

**OID** 1.3.6.1.4.1.2636.3.9.1.9.17.0

**Description** Active UDP sessions (count).

### UDP Session Count

**Example**      [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorUDPSessions.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorUDPSessions.0 = INTEGER: 0

### ICMP Session Count

**Name**          jnxIdpSensorICMPSessions

**OID**            1.3.6.1.4.1.2636.3.9.1.9.18.0

**Description**   Active ICMP sessions (count).

**Example**      [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorICMPSessions.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorICMPSessions.0 = INTEGER: 0

### Other Session Count

**Name**          jnxIdpSensorOtherSessions

**OID**            1.3.6.1.4.1.2636.3.9.1.9.19.0

**Description**   Active sessions that are not TCP, UDP, or ICMP (count).

**Example**      [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorOtherSessions.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorOtherSessions.0 = INTEGER: 0

[Table 110 on page 431](#) describes the device statistics reported to SNMP traps when resource traps are enabled.

**Table 110: System Resource Instrumentation: Resource SNMP Traps**

Object Name	Object ID	Description
jnxIdpSessionCountNotify	jnxIdpTrapsPrefix 1	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorSessAllocated, jnxIdpSensorSessThreshold
jnxIdpSessionCountLimitRestored	jnxIdpTrapsPrefix 2	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorSessAllocated

Table 110: System Resource Instrumentation: Resource SNMP Traps (*continued*)

Object Name	Object ID	Description
jnxIdpEngineCpuUsgOneMinNotify	jnxIdpTrapsPrefix 11	Alarm is triggered when IDP engine CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId, jnxIdpSensorCpuThreshold
jnxIdpEngineCpuUsgOneMinRestored	jnxIdpTrapsPrefix 12	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId
jnxIdpEngineCpuUsgFiveMinNotify	jnxIdpTrapsPrefix 13	Alarm is triggered when IDP engine CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId, jnxIdpSensorCpuThreshold
jnxIdpEngineCpuUsgFiveMinRestored	jnxIdpTrapsPrefix 14	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId
jnxIdpEngineSessnCreateRateNotify	jnxIdpTrapsPrefix 15	Alarm is triggered when the rate exceeds the configured threshold.  Objects referenced: jnxIdpSensorSessnCreateRateFiveSec, jnxIdpSensorSessnCreateRateThreshold
jnxIdpEngineSessnCreateRateRestored	jnxIdpTrapsPrefix 16	Notification is triggered when the rate falls below the alarm threshold.  Objects referenced: jnxIdpSensorSessnCreateRateFiveSec



**NOTE:** After a down trap is sent, the alarm is suspended for 1 minute. After this period, alarms can again be triggered.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)
- [IDP Series MIB Object ID Reference on page 553](#)

## Configuring SNMP Reporting for the Rule Category of Statistics

IDP OS Release 5.1 supports extensive system resource instrumentation so you can use SNMP to monitor device health. The system resource instrumentation is classified in five categories: sensor, resource, traffic, rule, and interface. The rule category includes statistics about security rulebase matches and attack object matches. Rule statistics are reported to the IDP Series MIB but not to SNMP traps. We recommend you use NSM to analyze security statistics. We support SNMP reporting for cases where access to NSM is not immediately available.

To configure the rule category of statistics reporting:

1. Log in to the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the **list** command to display the current settings:

```
[root@defaulthost ~]# scio sri rule list
SRI rule :
sc_enable_rule_stats           = 1      [ 0 - 1 ]
```

By default, rule statistic reporting is enabled (value 1).

3. (Optional) Use the **set** command to change the default. The following example shows the commands you use to enable rule statistic reporting.

```
[root@defaulthost ~]# scio sri rule set sc_enable_rule_stats 1
```



**TIP:** If you know you want to enable statistics for all system resource instrumentation categories, use the following command: **scio sri all set sc\_enable\_all\_stats 1**. If you know you want to enable traps for all categories, use the following command: **scio sri all set \_sc\_enable\_all\_traps 1**.

The following tables describe the device statistics reported to the IDP Series device MIB when rule statistics are enabled.



**NOTE:** Counters used in rule statistics are reset when a new policy is pushed to the device.

### Rule Matches

<b>Name</b>	jnxIdpSensorRuleStatsTable
<b>OID</b>	1.3.6.1.4.1.2636.3.9.1.63
<b>Description</b>	Matches per rule (count).

## Rule Matches

**Example** [host]# `snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorRuleStatsTable`  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorRuleStatsTable

jnxIdpSensorRuleIndex	jnxIdpSensorRulebaseName	jnxIdpSensorRuleID	jnxIdpSensorRuleHits
1	"ids"	1	0

## Attack Object Matches

**Name** jnxIdpSensorSignatureStatsTable

**OID** 1.3.6.1.4.1.2636.3.9.1.64

**Description** Matches per attack object (count).

**Example** [host]# `snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorSignatureStatsTable`  
 JUNIPER-IDP-MIB::jnxIdpSensorSignatureStatsTable: No entries

## Top Ten Rule Matches

**Name** jnxIdpSensorTopTenRuleStatsTable

**OID** 1.3.6.1.4.1.2636.3.9.1.65

**Description** Top ten rules matched.

**Example** [host]# `snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTopTenRuleStatsTable`  
 SNMP table: JUNIPER-IDP-MIB::jnxIdpSensorTopTenRuleStatsTable

jnxIdpSensorTopTenRuleIndex	jnxIdpSensorTopTenRulebaseName	jnxIdpSensorTopTenRuleID	jnxIdpSensorTopTenRuleHits
0	1	"ids"	1

## Top Ten Attack Object Matches

**Name** jnxIdpSensorTopTenSignatureStatsTable

**OID** 1.3.6.1.4.1.2636.3.9.1.66

**Description** Top ten attack objects matched.

**Example** [host]# `snmptable -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTopTenSignatureStatsTable`  
 JUNIPER-IDP-MIB::jnxIdpSensorTopTenSignatureStatsTable: No entries

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)

- [IDP Series MIB Object ID Reference on page 553](#)

## Configuring SNMP Reporting for the Sensor Category of Statistics

IDP OS Release 5.1 supports extensive system resource instrumentation, so you can use SNMP to monitor device health and load. The system resource instrumentation is classified in five categories: sensor, resource, traffic, rule, and interface. In previous releases, the IDP Series device automatically sent alarm logs and SNMP traps when device CPU, memory, or disk space utilization exceeded the default threshold (80%). Beginning with IDP OS Release 5.1, these statistics are grouped in category called “sensor,” and reporting and trap behavior is configurable.

To configure the sensor category of statistics reporting and traps:

1. Log in to the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the **list** command to display the current settings:

```
[root@defaulthost ~]# scio sri sensor list
SRI sensor :
  sc_enable_sensor_stats      = 1      [ 0 - 1 ]
  sc_enable_sensor_traps      = 1      [ 0 - 1 ]
  thrshld_memory              = 80      [ 0 - 100 ]
  thrshld_hard_disk           = 80      [ 0 - 100 ]
  thrshld_control_cpu         = 80      [ 0 - 100 ]
```

By default, statistic reporting and traps are enabled (value 1) and the threshold to trigger traps for memory, hard disk, and control plane CPU utilization is 80% (value 80).

3. (Optional) Use the **set** command to change the defaults. For example, the following commands show how to change the threshold for sending control plane CPU traps to 85% and verify your change:

```
[root@defaulthost ~]# scio sri sensor set thrshld_control_cpu 85
Setting variable thrshld_control_cpu successful
```

```
[root@defaulthost ~]# scio sri sensor list
SRI sensor :
  sc_enable_sensor_stats      = 1      [ 0 - 1 ]
  sc_enable_sensor_traps      = 1      [ 0 - 1 ]
  thrshld_memory              = 80      [ 0 - 100 ]
  thrshld_hard_disk           = 80      [ 0 - 100 ]
  thrshld_control_cpu         = 85      [ 0 - 100 ]
```



**TIP:** If you know you want to enable statistics for all system resource instrumentation categories, use the following command: **scio sri all set sc\_enable\_all\_stats 1**. If you know you want to enable traps for all categories, use the following command: **scio sri all set \_sc\_enable\_all\_traps 1**.

The following tables describe the device statistics reported to the IDP Series device MIB when sensor statistics are enabled.

#### Control Plane CPU - Instant

Name	jnxIdpSensorCpuUsage
OID	1.3.6.1.4.1.2636.3.9.1.1.0
Description	Control plane CPU utilization (percent).
Example	[host]# <b>snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorCpuUsage.0</b> JUNIPER-IDP-MIB::jnxIdpSensorCpuUsage.0 = Gauge32: 19

#### Control Plane CPU - Last 1 Minute

Name	jnxIdpSensorCpuUsageOneMin
OID	1.3.6.1.4.1.2636.3.9.1.10.0
Description	Average control plane CPU utilization in the last 1 minute (percent).
Example	[host]# <b>snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorCpuUsageOneMin.0</b> JUNIPER-IDP-MIB::jnxIdpSensorCpuUsageOneMin.0 = Gauge32: 25

#### Control Plane CPU - Last 5 Minutes

Name	jnxIdpSensorCpuUsageFiveMin
OID	1.3.6.1.4.1.2636.3.9.1.11.0
Description	Average control plane CPU utilization in the last 5 minutes (percent).
Example	[host]# <b>snmpget -v2c -c public 10.209.96.78 snmpget -v2c -c public 10.209.96.78</b> <b>JUNIPER-IDP-MIB::jnxIdpSensorCpuUsageFiveMin.0</b> JUNIPER-IDP-MIB::jnxIdpSensorCpuUsageFiveMin.0 = Gauge32: 20

#### Device Memory

Name	jnxIdpSensorMemUsage
OID	1.3.6.1.4.1.2636.3.9.1.2.0
Description	Memory utilization (percent).



## Device Memory

**Example**      [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorMemUsage.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorMemUsage.0 = Gauge32: 44

## Device Disk Utilization

**Name**          jnxIdpSensorFreeDiskSpace

**OID**            1.3.6.1.4.1.2636.3.9.1.5.0

**Description**   Available disk space (megabytes).

**Example**      [host]# **snmpget -v2c -c public 10.209.96.78 JUNIPER-IDP-MIB::jnxIdpSensorFreeDiskSpace.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorFreeDiskSpace.0 = Gauge32: 55865 Megabytes

Table 111 on page 437 describes the device statistics reported to SNMP traps when sensor traps are enabled.

**Table 111: System Resource Instrumentation: Sensor SNMP Traps**

Object Name	OID	Description
jnxIdpCPUUtilizationNotify	jnxIdpTrapsPrefix 3	Alarm is triggered when control plane CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuThresh
jnxIdpCPUUtilizationLimitRestored	jnxIdpTrapsPrefix 4	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuThresh
jnxIdpMemoryNotify	jnxIdpTrapsPrefix 5	Alarm is triggered when utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorMemUsage, jnxIdpSensorMemThresh
jnxIdpMemoryLimitRestored	jnxIdpTrapsPrefix 6	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorMemUsage
jnxIdpDiskUtilizationNotify	jnxIdpTrapsPrefix 7	Alarm is triggered when utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorFreeDiskSpace, jnxIdpSensorDiskSpaceThresh

Table 111: System Resource Instrumentation: Sensor SNMP Traps (*continued*)

Object Name	OID	Description
jnxldpDiskUtilizationLimitRestored	jnxldpTrapsPrefix 8	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxldpSensorFreeDiskSpace
jnxldpControlCpuUsgFiveMinNotify	jnxldpTrapsPrefix 9	Alarm is triggered when control plane CPU utilization exceeds the configured threshold.  Objects referenced: jnxldpSensorCpuUsage, jnxldpSensorCpuUld, jnxldpSensorCpuThreshold
jnxldpControlCpuUsgFiveMinRestored	jnxldpTrapsPrefix 10	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxldpSensorCpuUsage, jnxldpSensorCpuUld



**NOTE:** After a down trap is sent, the alarm is suspended for 1 minute. After this period, alarms can again be triggered.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)
- [IDP Series MIB Object ID Reference on page 553](#)

## Configuring SNMP Reporting for the Traffic Category of Statistics

IDP OS Release 5.1 supports extensive system resource instrumentation so you can use SNMP to monitor device health. The system resource instrumentation is classified in five categories: sensor, resource, traffic, rule, and interface. The traffic category includes detailed statistics about traffic processed and traffic dropped by the IDP Series device. Traffic statistics are reported to the MIB but not to SNMP traps. You use the traffic category of statistics to understand traffic volume and causes of packet drops.

To configure the traffic category of statistics reporting:

1. Log in to the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the **list** command to display the current settings:

```
[root@defaulthost ~]# scio sri traffic list
SRI traffic :
  sc_enable_traffic_stats           = 1      [ 0 - 1 ]
```

By default, traffic statistic reporting is enabled (value 1).

3. (Optional) Use the **set** command to change the default:

```
[root@defaulthost ~]# scio sri traffic set sc_enable_traffic_stats 0
```

```
[root@defaulthost ~]# scio sri traffic list
SRI traffic :
sc_enable_traffic_stats          = 0      [ 0 - 1 ]
```



**TIP:** If you know you want to enable statistics for all system resource instrumentation categories, use the following command: `scio sri all set sc_enable_all_stats 1`. If you know you want to enable traps for all categories, use the following command: `scio sri all set _sc_enable_all_traps 1`.

The following tables describe the device statistics reported to the IDP Series device MIB when traffic statistics are enabled.



**NOTE:** Traffic statistics use the same counters as `scio subs status s0` and `scio counter get fragment` commands. Counters used for cumulative packet counts are reset when the IDP Series device is rebooted. Counters used for rate statistics are reset when the IDP engine is restarted. The statistics returned are the results of an instant polling. If the device is not currently processing traffic, the rate reported is the valid value for the last time the device was processing traffic.

#### Total Packets Received (Rate)

**Name** jnxIdpSensorPacketsPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.22.0

**Description** Rate that packets are received (packets per second, or pps).

**Example** [host]# `snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPacketsPerSec.0`  
JUNIPER-IDP-MIB::jnxIdpSensorPacketsPerSec.0 = INTEGER: 2

#### Bytes Received (Rate)

**Name** jnxIdpSensorBytesPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.23.0

**Description** Rate that bytes are received (bytes per second, or bps).

**Example** [host]# `snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorBytesPerSec.0`  
JUNIPER-IDP-MIB::jnxIdpSensorBytesPerSec.0 = INTEGER: 1

#### IPv4 Packets Received (Rate)

**Name** jnxIdpSensorIPv4PktsPerSec

## IPv4 Packets Received (Rate)

**OID** 1.3.6.1.4.1.2636.3.9.1.24.0

**Description** Rate that IPv4 packets are received.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorIPv4PktsPerSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorIPv4PktsPerSec.0 = INTEGER: 2

## Non-IPv4 Packets Received (Rate)

**Name** jnxIdpSensorNonIPv4PktsPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.25.0

**Description** Rate that non-IPv4 packets are received.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorNonIPv4PktsPerSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorNonIPv4PktsPerSec.0 = INTEGER: 0

## TCP Packets Received (Rate)

**Name** jnxIdpSensorTCPPktsPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.26.0

**Description** Rate that TCP packets are received.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTCPPktsPerSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorTCPPktsPerSec.0 = INTEGER: 0

## UDP Packets Received (Rate)

**Name** jnxIdpSensorUDPPktsPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.27.0

**Description** Rate that UDP packets are received.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorUDPPktsPerSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorUDPPktsPerSec.0 = INTEGER: 0

## ICMP Packets Received (Rate)

**Name** jnxIdpSensorICMPPktsPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.28.0

### ICMP Packets Received (Rate)

**Description** Rate that ICMP packets are received.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorICMPPktsPerSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorICMPPktsPerSec.0 = INTEGER: 2

### Other Packets Received (Rate)

**Name** jnxIdpSensorOtherPktsPerSec

**OID** 1.3.6.1.4.1.2636.3.9.1.29.0

**Description** Rate that traffic other than TCP, UDP, and ICMP is received.

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorOtherPktsPerSec.0**  
JUNIPER-IDP-MIB::jnxIdpSensorOtherPktsPerSec.0 = INTEGER: 0

### Total Packets Processed (Count)

**Name** jnxIdpSensorPktsProcessed

**OID** 1.3.6.1.4.1.2636.3.9.1.31.0

**Description** Total packets processed (count).

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsProcessed.0**  
JUNIPER-IDP-MIB::jnxIdpSensorPktsProcessed.0 = INTEGER: 2082

### Total Bytes Processed (Count)

**Name** jnxIdpSensorBytesProcessed

**OID** 1.3.6.1.4.1.2636.3.9.1.32.0

**Description** Total bytes processed (count).

**Example** [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorBytesProcessed.0**  
JUNIPER-IDP-MIB::jnxIdpSensorBytesProcessed.0 = INTEGER: 172203

### Total TCP Packets Processed (Count)

**Name** jnxIdpSensorTCPPktsProcessed

**OID** 1.3.6.1.4.1.2636.3.9.1.33.0

**Description** Total TCP packets processed (count).

## Total TCP Packets Processed (Count)

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTCPPktsProcessed.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorTCPPktsProcessed.0 = INTEGER: 183`

## Total UDP Packets Processed (Count)

**Name**          `jnxIdpSensorUDPPktsProcessed`

**OID**            `1.3.6.1.4.1.2636.3.9.1.34.0`

**Description**   `Total UDP packets processed (count).`

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorUDPPktsProcessed.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorUDPPktsProcessed.0 = INTEGER: 21`

## Total ICMP Packets Processed (Count)

**Name**          `jnxIdpSensorICMPPktsProcessed`

**OID**            `1.3.6.1.4.1.2636.3.9.1.35.0`

**Description**   `Total ICMP packets processed (count).`

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorICMPPktsProcessed.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorICMPPktsProcessed.0 = INTEGER: 1918`

## Total Other Packets Processed (Count)

**Name**          `jnxIdpSensorOtherPktsProcessed`

**OID**            `1.3.6.1.4.1.2636.3.9.1.36.0`

**Description**   `Total packets processed for traffic that is not TCP, UDP, or ICMP (count).`

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorOtherPktsProcessed.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorOtherPktsProcessed.0 = INTEGER: 0`

## Total Fragments Received (Count)

**Name**          `jnxIdpSensorFragmentsRxd`

**OID**            `1.3.6.1.4.1.2636.3.9.1.38.0`

**Description**   `Fragments received (count).`

## Total Fragments Received (Count)

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorFragmentsRxd.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorFragmentsRxd.0 = INTEGER: 0`

## Total Fragments Reassembled (Count)

**Name**          `jnxIdpSensorFragmentsReassembled`

**OID**            `1.3.6.1.4.1.2636.3.9.1.39.0`

**Description**   `Fragments reassembled (count).`

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorFragmentsReassembled.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorFragmentsReassembled.0 = INTEGER: 0`

## Total Fragments Dropped (Count)

**Name**          `jnxIdpSensorFragmentsDropped`

**OID**            `1.3.6.1.4.1.2636.3.9.1.40.0`

**Description**   `Fragments dropped (count).`

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorFragmentsDropped.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorFragmentsDropped.0 = INTEGER: 0`

## Packets Dropped By Security Policy Action (Count)

**Name**          `jnxIdpSensorPktsDroppedToRule`

**OID**            `1.3.6.1.4.1.2636.3.9.1.41.0`

**Description**   `Packets dropped because of a security policy rule being applied (count).`

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToRule.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToRule.0 = INTEGER: 0`

## Packets Dropped Because of a Checksum Error (Count)

**Name**          `jnxIdpSensorPktsDroppedToChksum`

**OID**            `1.3.6.1.4.1.2636.3.9.1.42.0`

**Description**   `Packets dropped because of a checksum error (count).`

## Packets Dropped Because of a Checksum Error (Count)

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToChksum.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToChksum.0 = INTEGER: 0`

## Packets Dropped Because of Protocol Anomaly (Count)

**Name**          `jnxIdpSensorPktsDroppedToAnomaly`

**OID**            `1.3.6.1.4.1.2636.3.9.1.43.0`

**Description**    Packets dropped because of a protocol anomaly detected (count). Reported value is derived from the `sc_ids_l4_anomalies` counter (`scio counter get ids | grep sc_ids_l4_anomalies`).

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToAnomaly.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToAnomaly.0 = INTEGER: 0`

## Packets Dropped for Other Reasons (Count)

**Name**          `jnxIdpSensorPktsDroppedToMisc`

**OID**            `1.3.6.1.4.1.2636.3.9.1.44.0`

**Description**    Packets dropped for other reasons (count).

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToMisc.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToMisc.0 = INTEGER: 0`

## Packets Dropped For Nonpolicy Reasons (Count)

**Name**          `jnxIdpSensorPktsDroppedToNonRule`

**OID**            `1.3.6.1.4.1.2636.3.9.1.45.0`

**Description**    Packets dropped for a nonpolicy reason (count).

**Example**      `[host]# snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToNonRule.0`  
`JUNIPER-IDP-MIB::jnxIdpSensorPktsDroppedToNonRule.0 = INTEGER: 0`

## Total Alerts (Count)

**Name**          `jnxIdpSensorTotalAlerts`

**OID**            `1.3.6.1.4.1.2636.3.9.1.46.0`

**Description**    Total alerts generated (count).



### Total Alerts (Count)

**Example**      [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTotalAlerts.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorTotalAlerts.0 = INTEGER: 0

### Total Logs (Count)

**Name**          jnxIdpSensorTotalLogs

**OID**            1.3.6.1.4.1.2636.3.9.1.47.0

**Description**   Total logs generated (count).

**Example**      [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorTotalLogs.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorTotalLogs.0 = INTEGER: 89

### Log Rate

**Name**          jnxIdpSensorLogsPerSec

**OID**            1.3.6.1.4.1.2636.3.9.1.48.0

**Description**   Rate of logs (per second).

**Example**      [host]# **snmpget -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensorLogsPerSec.0**  
 JUNIPER-IDP-MIB::jnxIdpSensorLogsPerSec.0 = INTEGER: 0

- Related Documentation**    The following related topics are included in the *IDP Series Administration Guide*:
- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)
  - [IDP Series MIB Object ID Reference on page 553](#)



# Using NSM Logs and Reports

- IDP Series Logs and Reports in NSM Task Summary on page 447
- Viewing Device Status (NSM Procedure) on page 448
- Using NSM Logs on page 453
- Viewing Simulation Mode Logs on page 461
- Using Profiler Viewer (NSM Procedure) on page 463
- Viewing NSM Predefined Reports (NSM Procedure) on page 469
- Creating NSM Custom Reports (NSM Procedure) on page 472

## IDP Series Logs and Reports in NSM Task Summary

---

IDP Series devices generate logs about *device status* based on built-in criteria and about *security events* based on the security policy notification settings. These logs are automatically sent to the NSM GUI server and can be viewed in the NSM log viewer.

IDP Series administration includes the following log-related tasks:

- Viewing device status, logs, and reports.
- Viewing attack logs and reports.
- Viewing application usage logs and reports.
- Configuring interface aliasing, if you want to identify IDP Series traffic interfaces by name in logs and reports.
- Configuring log suppression, if you want to reduce the number of identical log files.
- Configuring communication with an SNMP or syslog server, if you use external log programs to view alerts or analyze or archive log data.
- Ensuring collection of packet data in NSM logs is enabled, if you want to drill into packet data from NSM logs.



**NOTE:** To avoid issues with reports, we highly recommend that you synchronize the network clocks for all devices to the same NTP server. For example, the network clocks for all IDP devices and NSM clients should be synchronized to the NTP server specified in the NSM configuration.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing Device Status \(NSM Procedure\) on page 448](#)
- [Using NSM Logs on page 453](#)
- [Viewing NSM Predefined Reports \(NSM Procedure\) on page 469](#)
- [Configuring Interface Aliasing \(ACM Procedure\) on page 296](#)
- [Configuring Log Suppression \(NSM Procedure\) on page 298](#)
- [Configuring an SNMP Agent \(NSM Procedure\) on page 299](#)
- [Configuring Syslog Collection \(NSM Procedure\) on page 301](#)
- [Enabling Collection of Packet Data in NSM Logs \(NSM Procedure\) on page 303](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)

## Viewing Device Status (NSM Procedure)

**Purpose** You monitor NSM device status to see whether there are any issues with the communication between NSM and the IDP Series device. Within the NSM Device Monitor, you can also drill-down for status on IDP Series device CPU, memory, and session utilization.

Figure 141 on page 448 shows the NSM Device Monitor.

**Figure 141: NSM Device Monitor**

Device Monitor											
Name	Domain	Platform	OS Version	Conn. Status	Alarm	HW Inven...	SW Inven...	Config St.	License In...	First Conn.	Latest Co.
EP-1100F	global	NS-IDP-1100F	IDP5.0	Up	N/A	N/A	N/A	Managed	N/A	Tue Mar 17 1...	Thu Mar 26 1...
idp-75	global	NS-IDP-75	IDP5.0.124008	Up	N/A	N/A	N/A	Managed	N/A	Sat Mar 14 1...	Thu Mar 26 1...
EP-NS-2...	global	NS-IDP-200	IDP5.0	Up	N/A	N/A	N/A	Managed	N/A	Thu Mar 12 1...	Thu Apr 02 1...
JTAC-ID...	global	NS-IDP-800	IDP5.0	Down	N/A	N/A	N/A	Managed	N/A	Fri Mar 27 10...	Mon Mar 30 ...

**Action** To display the Device Monitor, in the NSM navigation tree, select **Investigate > Realtime Monitor > Device Monitor**.

Table 112 on page 448 describes NSM device monitor status data.

**Table 112: NSM Device Monitor Status Data**

Column	Description
Name	Displays the NSM name for the device. The NSM name is a value you specify when you add the device to the NSM Device Manager.
Domain	Displays the NSM domain to which the device is a member.
Platform	Displays the device model number.
OS version	Displays the operating system version.

Table 112: NSM Device Monitor Status Data (*continued*)

Column	Description
Connection status	<p>Displays the status of the connection between the device and NSM:</p> <ul style="list-style-type: none"> <li>• <b>Up</b></li> <li>• <b>Down</b></li> <li>• <b>Never Connected</b></li> </ul>
Alarm	<p>Displays the most severe alarm for the device (if applicable). Double-click an alarm to view the Alarm Details dialog box, which lists all alarms and their polling time for that device.</p>
Hardware inventory status	<p>Displays the status of hardware inventory data:</p> <ul style="list-style-type: none"> <li>• <b>In Sync</b>—The NSM device configuration data matches the IDP Series device running configuration data.</li> <li>• <b>Reconciliation needed</b>—The NSM device configuration data does not match the IDP Series device running configuration data.</li> <li>• <b>Unknown</b>—Inventory information is unknown (the device might not be deployed yet).</li> <li>• <b>N/A</b>—Inventory information is not available.</li> </ul>
Software inventory status	<p>Displays the status of software inventory data:</p> <ul style="list-style-type: none"> <li>• <b>In Sync</b>—The NSM device configuration data matches the IDP Series device running configuration data.</li> <li>• <b>Reconciliation needed</b>—The NSM device configuration data does not match the IDP Series device running configuration data.</li> <li>• <b>Unknown</b>—Inventory information is unknown (the device might not be deployed yet).</li> <li>• <b>N/A</b>—Inventory information is not available.</li> </ul>
Configuration state	<p>Displays the status of the NSM device configuration compared to the IDP Series device running configuration:</p> <ul style="list-style-type: none"> <li>• <b>Managed</b>—Indicates that the device is managed by NSM.</li> <li>• <b>Modeled</b>—Indicates that the security device exists in NSM but has not been pushed to the IDP Series device.</li> <li>• <b>RMA</b>—Indicates a device that has been reverted to a modeled state.</li> <li>• <b>Waiting for 1st connect</b>—Indicates that NSM is waiting for the device to connect.</li> <li>• <b>Import Needed</b>—Indicates the running configuration has changed and you should import the configuration from the IDP Series device to synchronize the NSM device configuration.</li> <li>• <b>Update Needed</b>—Indicates a change to the NSM device configuration that needs to be pushed to the IDP Series device.</li> <li>• <b>OS Version Adjustment Needed</b>—Indicates that the firmware version detected on the device is different from what was previously detected by NSM.</li> </ul>
License inventory status	<p>Displays hardware inventory information:</p> <ul style="list-style-type: none"> <li>• <b>In Sync</b>—The NSM device configuration data matches the IDP Series device running configuration data.</li> <li>• <b>Reconciliation needed</b>—The NSM device configuration data does not match the IDP Series device running configuration data.</li> <li>• <b>Unknown</b>—Inventory information is unknown (the device might not be deployed yet).</li> <li>• <b>N/A</b>—Inventory information is not available.</li> </ul>

Table 112: NSM Device Monitor Status Data (*continued*)

Column	Description
First connect	Displays the date and time the device first connected to NSM.
Latest connect	Displays the date and time the device last connected to NSM.
Latest disconnect	Displays the date and time the device last disconnected from NSM.

To drill down to CPU and memory statistics, right-click the name of the device and select **View Details**.

Figure 142 on page 450 shows the NSM Device Details page, which includes CPU and memory statistics.

Figure 142: NSM Device Detail

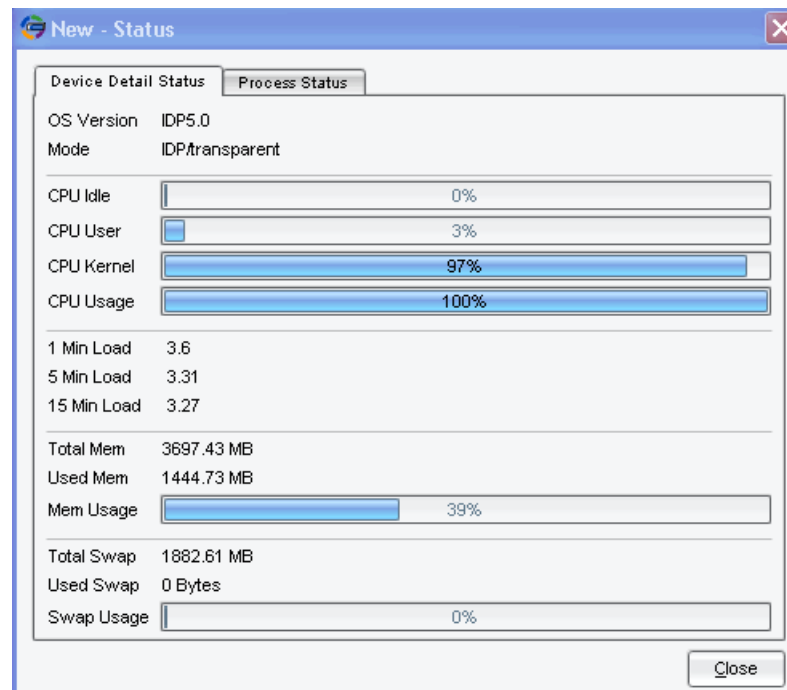


Table 113 on page 450 describes the NSM Device Details Page.

Table 113: NSM Device Monitor: Device Details Page

Column	Description
OS Version	The IDP operating system version.
Mode	Current operation mode of the device.
CPU Idle	Percentage of the time the CPU was idle.

Table 113: NSM Device Monitor: Device Details Page (*continued*)

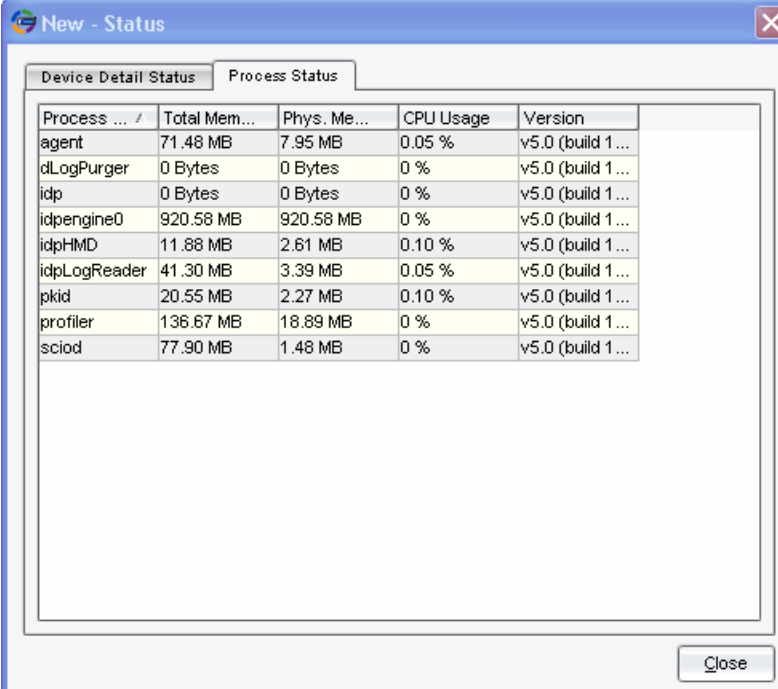
Column	Description
CPU User	Percentage of CPU utilization for user processes.
CPU Kernel	Percentage of CPU utilization for kernel processes.
CPU Usage	Combined CPU utilization for user and kernel processes.  <b>NOTE:</b> This CPU statistic shown here is the value returned from the Linux <b>top</b> command for CPU0. On IDP8200, only control plane processes run on CPU0. On IDP1100, IDP800, and IDP250, control plane processes and JNET driver processes run on CPU0. On IDP600 and IDP75, all processes run on CPU0.
1 Min Load	One-minute load average.
5 Min Load	Five-minute load average.
15 Min Load	Fifteen-minute load average.
Total Mem	Total amount (in megabytes) of memory.
Used Mem	Amount (in megabytes) of used memory.
Mem Usage	Percentage of used memory.
Total Swap	Total amount (in megabytes) of swap space.
Used Swap	Amount (in megabytes) of used swap space.
Swap Usage	Percentage of used swap space.

To drill down to process status:

1. Right-click a row in the Device Monitor report and select **View Device Details**.
2. Click the **Process Status** tab.

[Figure 143 on page 452](#) shows Process Status page, which gives details on memory and CPU usage per IDP process.

Figure 143: NSM Device Monitor: Process Status Page



Process ... /	Total Mem...	Phys. Me...	CPU Usage	Version
agent	71.48 MB	7.95 MB	0.05 %	v5.0 (build 1 ...
dLogPurger	0 Bytes	0 Bytes	0 %	v5.0 (build 1 ...
idp	0 Bytes	0 Bytes	0 %	v5.0 (build 1 ...
idpengine0	920.58 MB	920.58 MB	0 %	v5.0 (build 1 ...
idpHMD	11.88 MB	2.61 MB	0.10 %	v5.0 (build 1 ...
idpLogReader	41.30 MB	3.39 MB	0.05 %	v5.0 (build 1 ...
pkid	20.55 MB	2.27 MB	0.10 %	v5.0 (build 1 ...
profiler	136.67 MB	18.89 MB	0 %	v5.0 (build 1 ...
sciold	77.90 MB	1.48 MB	0 %	v5.0 (build 1 ...

**NOTE:**

Due to a limitation, the CPU usage for the IDP engine is reported as 0%. To see the actual CPU usage for an IDP engine

- For multicore platforms, log into the IDP OS command-line interface (CLI) and use the `scio idp-cpu-utilization` command.
- For single core platforms, log into the IDP OS command-line interface (CLI) and use the Linux `top` command.

The correct CPU usage is also reported via SNMP.

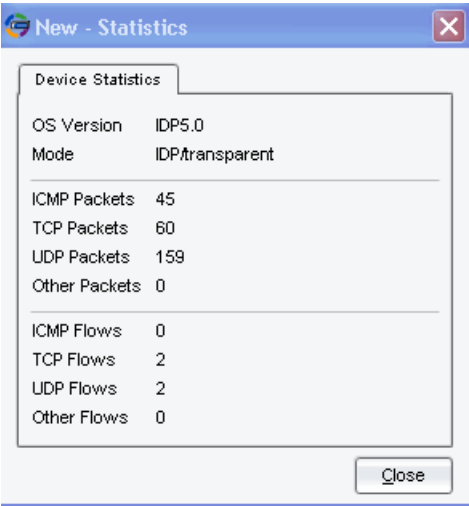
To drill down to packet and flow counters for current traffic:

1. In the NSM navigation tree, select **Investigate > Realtime Monitor > Device Monitor**.
2. Right-click the name of the device and select **View Statistics**.

Figure 144 on page 453 shows the Device Statistics page.



Figure 144: NSM Device Monitor: Device Statistics



- Related Documentation**    The following related topics are included in the *IDP Series Administration Guide*:
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)
  - [SNMP Statistic Reporting and Traps Task Summary on page 417](#)

## Using NSM Logs

You use NSM to view logs related to IDP Series device status and security events. This section includes the following topics:

- [NSM Logs Overview on page 453](#)
- [Using NSM Log Viewer \(NSM Procedure\) on page 454](#)
- [Using NSM Log Investigator \(NSM Procedure\) on page 459](#)
- [Using NSM Audit Log Viewer \(NSM Procedure\) on page 460](#)

### NSM Logs Overview

NSM collects logs from managed IDP Series devices and stores them in a central log database. You can use NSM to view, manipulate, and export logs.

[Table 114 on page 453](#) provides a reference of log views.

Table 114: Log Viewing Options

Log Views	Description
NSM Log Viewer / Log Investigator	Logs based on notification options you set for security policy rules.
	Logs related to device events, such as changes in the state of a traffic interface.
NSM Security Monitor	Logs produced by the Profiler feature.

Table 114: Log Viewing Options (*continued*)

Log Views	Description
NSM Audit Log Viewer	Logs generated by NSM related to the use of NSM to manage the IDP Series device.

## Using NSM Log Viewer (NSM Procedure)

**Purpose** You use the NSM Log Viewer to access logs generated when traffic matches a security policy rule.

Figure 145 on page 454 shows the NSM log viewer. You can use NSM management features to flag logs for filtering or follow up. The bottom panes include summary information for the attack and the data that matched the rule.

Figure 145: NSM Log Viewer

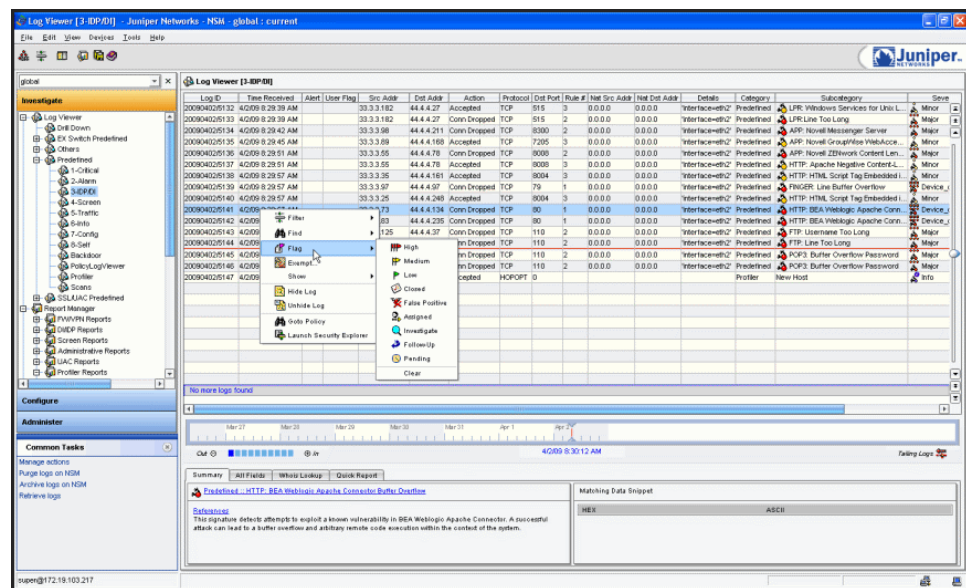


Table 115 on page 454 describes the columns in the NSM Log Viewer table display.

Table 115: NSM Log Viewer: Log Columns

Column	Description
Log ID	Unique ID for the log entry, derived from the combination of the date and log number.
Time Received	Date and time that the management system received the log entry.
Alert	Displays an icon if the log matches a rule for which the alert flag was selected.

Table 115: NSM Log Viewer: Log Columns (*continued*)

Column	Description
User Flag	<p>To set a flag, right-click the log row, select Flag, and then select one of the following flags:</p> <ul style="list-style-type: none"> <li>• High</li> <li>• Medium</li> <li>• Low</li> <li>• Closed</li> <li>• False Positive</li> <li>• Assigned</li> <li>• Investigate</li> <li>• Follow-up</li> <li>• Pending</li> </ul>
Src Addr	Source IP address of the packet that generated the log entry.
Dst Addr	Destination IP address of the packet that generated the log entry.
Action	<p>Action the security device performed on the packet/connection that generated this log entry:</p> <ul style="list-style-type: none"> <li>• Accepted—The device did not block the packet.</li> <li>• Closed Client—The device closed the connection and sent a RST packet to the client, but did neither to the server.</li> <li>• Closed Server—The device closed the connection and sent a RST packet to the server, but did neither to the client.</li> <li>• Closed—The device closed the connection and sent a RST packet to both the client and the server.</li> <li>• Dropped—The device dropped the connection without sending a RST packet to the sender, preventing the traffic from reaching its destination.</li> <li>• Dropped Packet—The device dropped a matching packet before it could reach its destination but did not close the connection.</li> <li>• Ignored—Matched the attack, did not take action, and ignored the remainder of the connection.</li> </ul> <p><b>NOTE:</b> IDP logs show the action that was set in the rule, not necessarily the actual action taken. For TCP events, these are the same. For UDP and ICMP events, the IDP logs show close client, close server, and close client and server actions, even when the actual action taken was a drop (close actions are not possible for UDP or ICMP packets).</p>
Protocol	Protocol that the packet that generated the log entry used.
Dst Port	Destination port of the packet that generated the log entry.
Rule #	The rule in a policy rulebase (in a specific version of a domain) that generated the log entry.
Nat Src Addr	The NAT source address of the packet that generated the log entry.
Nat Dst Addr	The NAT destination address of the packet that generated the log entry.
Details	Miscellaneous string associated with log entry.

Table 115: NSM Log Viewer: Log Columns (*continued*)

Column	Description
Category	<p>Type of log entry:</p> <ul style="list-style-type: none"> <li>Alarm. The device generates event alarms for any security event that has a predefined severity level of emergency, critical, or alert. Additionally, the device generates traffic alarm log entries when it detects network traffic that exceeds the specified alarm threshold in a rule (the traffic alarm log entry describes the security event that triggered the alarm).</li> <li>Config. A configuration change occurred on the device.</li> <li>Custom. A match with a custom attack object was detected.</li> <li>Implicit. An implicit rule was matched.</li> <li>Info. General system information.</li> <li>Profiler. Traffic matches a Profiler alert setting.</li> <li>Screen. Not applicable for IDP Series devices. Screen alarms are generated by ScreenOS firewall devices.</li> <li>Self. The device generated this log for a non-traffic related reason.</li> <li>Signature. Traffic matches an attack object.</li> <li>Traffic. Traffic matches a rule you have configured for harmless traffic.</li> </ul>
Subcategory	Category-specific type of log entry (examples are "Reboot" or message ID).
Severity	<p>Severity rating associated (if any) with this type of log entry:</p> <ul style="list-style-type: none"> <li>Not Set (the device could not determine a severity for this log entry)</li> <li>Info</li> <li>Device_warning_log</li> <li>Minor</li> <li>Major</li> <li>Device_critical_log</li> <li>Emergency</li> <li>Error</li> <li>Notice</li> <li>Informational</li> <li>Debug</li> </ul>
Device	Device that generated this log entry.
Comment	User defined comment about the log entry.
Application Name	Application associated with the current log.
Bytes In	For sessions, specifies the number of inbound bytes.
Bytes Out	For sessions, specifies the number of outbound bytes.
Bytes Total	For sessions, specifies the combined number of inbound and outbound bytes.
Dev Domain Ver	Domain version that generated this log entry.

Table 115: NSM Log Viewer: Log Columns (*continued*)

Column	Description
Device Domain	Domain for the device that generated this log entry.
Device family	Family of the device that generated this log entry.
Dst Intf	Name of the outbound interface of the packet that generated this log entry. <i>TIP:</i> Use ACM to configure an alias for the interface if you want to be able to view or sort on the alias.
Dst Zone	Destination zone associated with a traffic log entry.
Elapsed Secs	For sessions, specifies how long the session lasted.
Has Packet Data	If a marker appears in this column, you can right click the row and select <b>Show &gt; Packet Data</b> or <b>Show &gt; Packet Data in External Viewer</b> to view the packet capture.
NAT Dst Port	The NAT destination port of the packet that generated the log entry.
NAT Src Port	The NAT source port of the packet that generated the log entry.
Packets In	For sessions, specifies the number of inbound packets.
Packets Out	For sessions, specifies the number of outbound packets.
Packets Total	For sessions, specifies the combined number of inbound and outbound packets.
Policy	The security policy (in a specific version of a domain) whose rule generated the log entry.
Roles	Role group associated with this log entry.
Rule Domain	The domain of the rule that generated the log entry.
Rule Domain Ver	The domain version of the rule that generated the log entry.
Rulebase	The security policy rulebase (in a specific version of a domain) that generated the log entry.
Src Intf	Name of the inbound interface of the packet that generated this log entry. <i>TIP:</i> Use ACM to configure an alias for the interface if you want to be able to view or sort on the alias.
Src Port	Source port of the packet that generated the log entry.
Src Zone	Source zone associated with a traffic log entry.
Time Generated	Date and time the device generated the log entry.
User	User associated with this log entry.



**NOTE:** Data is collected for all fields but not all columns are displayed by default. Select **View > Choose Columns** to select the columns you want to monitor.

You can drill from logs to packet captures by right clicking a log that contains the packet capture and selecting the NSM packet viewer or an external packet viewer. [Figure 146 on page 458](#) shows the NSM packet viewer.

**Figure 146: NSM Packet Viewer**

No	Source IP	Source MAC	Dest IP	Dest MAC	Protocol	Src Port	Dst Port	Length	VLAN
33	3.3.3.73	0:50:56:C0:0:0	44.4.4.134	0:50:56:C0:0:1	TCP	60562	80	1500	false

0000	50 4f 53 54 20 2f 2e 6a	73 70 20 63 6d 64 2e 65	POST /.jsp cmd.e
0010	78 65 20 2f 63 20 22 65	63 68 6f 20 77 6f 72 6b	xe /c "echo work
0020	73 20 3e 20 63 3a 5c 64	65 73 69 72 65 64 66 69	s > c:\desiredfi
0030	6c 65 2e 74 78 74 22 7c	41 41 41 41 41 41 41 41	le.txt"  AAAAA
0040	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0050	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0060	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0070	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0080	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0090	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
00a0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
00b0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
00c0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
00d0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
00e0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
00f0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0100	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA
0110	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAAAAAA



**NOTE:** Packet captures are included in NSM log records only if you configure the packet logging notification option in your security policy rule.

**Action** To display logs in NSM Log Viewer:

1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined**.
2. Click a predefined category to display a filtered view of logs. [Table 116 on page 459](#) describes the predefined views.

Table 116: NSM Log Viewer: Predefined Views

View	Description
Critical	Displays events that match security policy rules marked with severity of critical.
Alarm	Displays events that match security policy rules with notification options set to mark the event as an alarm event.
DI/IDP	Displays all log entries with signature, anomaly, or custom in the sub category column. IDP log entries provide information about an attack match against an IDP attack object. DI log entries provide information about an attack match against a deep inspection profile object.
Screen	Not applicable for IDP Series devices. Screen alarms are generated by ScreenOS firewall devices.
Traffic	Displays logs for traffic that matches a rule but the severity is low and notification option is log only.
Info	Displays info log entries. Info log entries provide general system information.
Config	Displays all configuration log entries. Configuration log entries provide information about a configuration or operational state change in Network and Security Manager.
Self	Displays all logs generated for non-traffic related reasons.
Profiler	Displays Profiler logs.
Backdoor	Displays log records generated by rules in the Backdoor rulebase.
Scans	Displays log records with a scan entry in the subcategory column, such as port scan.



**TIP:** For details on using NSM to create custom views, see the NSM online Help.

## Using NSM Log Investigator (NSM Procedure)

**Purpose** You use the NSM Log Investigator to analyze aggregations of logs and drill down based on properties of interest.

**Action** To display logs in NSM Log Investigator, in the NSM navigation tree, select **Investigate > Log Investigator**.



**TIP:** For details on using NSM to modify aggregation or display options, see the NSM online Help.

## Using NSM Audit Log Viewer (NSM Procedure)

**Purpose** You use the NSM Audit Log Viewer to view logs generated by NSM related to the use of NSM to manage the IDP Series device.

**Action** To display the NSM Audit Log Viewer table, in the NSM navigation tree, select **Investigate > Audit Log Viewer**.

[Table 117 on page 460](#) describes the columns in the Audit Log Viewer table.

**Table 117: NSM Audit Log Viewer Table**

Column	Description
Time Generated	The time the object was changed. The Audit Log Viewer displays log entries in order of time generated by Greenwich Mean Time (GMT).
Admin Name	The name of the NSM administrator who changed the object.
Admin Login Domain	The name of the domain (global or subdomain) that contains the changed object.
Authorization Status	The final access-control status of activities is either success or failure.
Command	The command applied to the object or system, for example, sys_logout or modify.
Targets	For changes made to a device configuration or object, the Audit Log Viewer displays the object type, object name, and object domain.
Devices	For changes made to a device, the Audit Log Viewer displays the device name, object type, and device domain.  For changes made to the management system, such as administrator login or logout, the Audit Log Viewer does not display target or device data.
Miscellaneous	Additional information that is not displayed in other audit log columns.

To display details of a configuration change, such as a changed IP address or renamed device, select the audit log entry for that change in the Audit Log table and view details in the Target View table, which appears below the Audit Log Viewer table.

[Table 118 on page 460](#) describes the Target View table.

**Table 118: NSM Audit Log Viewer: Target View Table**

Column	Description
Target Name	To see additional details for an target view entry, double-click the entry. NSM displays the configuration screen that the change was made in and marks the changed field with a solid green triangle.
Table	To set the table details for the target view entry, double-click the table. Enter or update the options.



Table 118: NSM Audit Log Viewer: Target View Table (*continued*)

Column	Description
Domain ID	Specifies the domain ID of the target view.

To display details of a nonconfiguration event, such as adding the device, auto-detecting a device, or rebooting a device, select the audit log entry for that change in the Audit Log table and view details in the Device View table, which is displayed below the Audit Log Viewer table.

[Table 119 on page 461](#) describes the Device View table.

Table 119: NSM Audit Log Viewer: Device View Table

Column	Description
Device Name	To see additional details for an device view entry, double-click the entry. NSM displays the Job Manager information window for the job task.
Table	To set the table details for the device view entry, double-click the table. Enter or update the options.
Domain ID	Specifies the domain ID of the device view.

- Related Documentation**
- The following related topic is included in the *IDP Series Administration Guide*:

  - [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

  - [Example: Using NSM Log Viewer Features on page 139](#)

Viewing Simulation Mode Logs

**Purpose** In the NSM Log Viewer, logs that are generated during simulation mode include the string [Simulation Mode] in the Details column, as shown in [Figure 147 on page 462](#). You can use the NSM log and report filter feature to filter for (or filter out) columns that include the [Simulation Mode] string.

Figure 147: NSM Log Viewer: Simulation Mode Logs

**Log Viewer [3-IDP/DI]**

Src Addr	Nat Dst Addr	Details	Category	Subcategory	Severity	Device	Comment
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth10' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth11' [Simulation Mode]	Predefined	Unnamed	Info	idp-10.209.85.107	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth2' ,alias=eth2' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth2' ,alias=eth2' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth2' ,alias=eth2' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	
0.0.0.0		'interface=eth1' ,alias=eth1' [Simulation ...	Predefined	Unnamed	Info	idp-10.209.85.9	

Timeline: Jul 12, Jul 13, Jul 14, Jul 15, Jul 16. Filter: Out, In. 7/16/10 2:15:24 AM. Tailing Logs.

**Summary** | All Fields | Whois Lookup | Quick Report

**Predefined :: Unnamed**

No Description

**Variable Data**

Header Checksum	0xec5a
Header Code	210
Header Type	14
N Id	1234
N Seq	1

**Action** To display logs in NSM Log Viewer:

1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined**.
2. Click **IDP/DI**.
3. (Optional) Click the filter icon to display the filter dialog box, which allows you to filter on the Details column and set a filter for [Simulation Mode].

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- Simulation Mode Task Summary
- IDP Series Logs and Reports in NSM Task Summary on page 295

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- Simulation Mode Overview on page 33

## Using Profiler Viewer (NSM Procedure)

The Profiler Viewer contains multiple tabs with different views of Profiler data. The following topics describe these views:

- [Application Profiler Tab on page 463](#)
- [Protocol Profiler Tab on page 465](#)
- [Network Profiler Tab on page 466](#)
- [Violation Viewer Tab on page 468](#)

### Application Profiler Tab

The Application Profiler tab displays application volume tracking (AVT) data. [Figure 32 on page 134](#) shows the Application Profiler tab.

Figure 148: Profiler Viewer: Application Profiler Tab

Application Category	Bytes	Packets	Src IP	Dest IP	Dest Port	VLAN ID	Application	Byte Count	Packet Count	User	Role	First Time	Last Time	Dom...	Device
Application	10.63 Mb	22,856	HTTP (1.03 Mb, 2,153)												
Unknown-application-category	9.51 Mb	19,887	Window...	128.241.220...	80	0	HTTP	222.34 Kb	329			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Unknown	9.51 Mb	19,887	Window...	204.160.122...	80	0	HTTP	188.69 Kb	342			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Web	1.03 Mb	2,153	Window...	204.160.122...	80	0	HTTP	150.56 Kb	295			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Http	1.03 Mb	2,153	Window...	207.46.26.20...	80	0	HTTP	122.94 Kb	312			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Remote-access	50.75 Kb	584	Window...	128.241.220...	80	0	HTTP	72.01 Kb	111			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Misc	14.61 Kb	109	Window...	0.12.204.126...	80	0	HTTP	55.79 Kb	97			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Nbds	8.83 Kb	39	Window...	4.59.126.37...	80	0	HTTP	52.42 Kb	76			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Noname	5.79 Kb	70	Window...	204.160.122...	80	0	HTTP	50.16 Kb	82			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Enterprise	12.75 Kb	86	Window...	by2msg1204...	80	0	HTTP	30.49 Kb	111			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Infrastructure	12.75 Kb	86	Window...	cf-in-r103 go...	80	0	HTTP	12.41 Kb	31			Mon Apr 06	Mon Apr 06	global	DP-NS-2...
Dns	6.98 Kb	68													
Dhcp	5.77 Kb	18													
Encryption	2.31 Kb	24													
Ssl	2.31 Kb	24													
Messaging	880	22													
Smtp	880	22													
Peer-to-peer	678	11													
File-sharing	378	6													
Gnutella-udp	378	6													
Chat	300	5													

Extended applications, also called nested applications, are reported separately from HTTP results. [Figure 33 on page 135](#) shows the Application Profiler tab where results for the HTTP Google Earth application are distinguished from HTTP results.

Figure 149: Profiler Viewer: Application Profiler Tab: Nested Applications

Application Category	Bytes	Packets	Src IP	Dest IP	Dest Port	VLAN ID	Application	Byte Count	Packet Co...	User	Role
Application	99.22 Mb	76,036	HTTP (62.14 Mb, 47,809)								
Web	99.22 Mb	76,036	vict1	att1	80	0	HTTP	62.14 Mb	47,809		
Http	62.14 Mb	47,809	GOOGLE-EARTH (37.08 Mb, 28,227)								
Google-earth	37.08 Mb	28,227	vict1	att1	80	0	GOOGLE-EA...	37.08 Mb	28,227		

The Application Profiler view is divided into two sections:

- In the left pane, the Application Profiler tab displays a hierarchical tree of application categories. Applications are grouped by common functionality. For example, peer-to-peer applications include chat and file sharing applications. Under chat, you can display Yahoo messenger, MSN, and AIM; under file sharing, you can display Kazaa, Bittorrent, and Gnutella.

The left pane also displays aggregate statistics for volume (bytes) and packet count for the application category, application group, or application you select in the tree.

- In the right pane, the Application Profiler tab displays tables of session logs related to the application category or application you select in the left pane.

Table 43 on page 135 describes the Application Profiler session table.

**Table 120: Application Profiler Session Table**

Column	Description
Src IP	Source IP address of the session.
Dst IP	Destination IP address.
VLAN ID	VLAN ID (if any).
Application ID	Application. The application identified by the application identification feature. Extended applications (also called nested applications) are reported separately from HTTP results. A 0 indicates the application was not identified.
Byte count	Byte count.
Packet count	Packet count.
User	The user associated with the session.
Role	The role to which the user belongs.
First Time	Timestamp for the first time the device logged the event (within the specified time interval).
Last Time	Timestamp for the last time the device logged the event (within the specified time interval).
Domain	NSM domain.
Device	Device through which the session was forwarded.

The Application Profiler session table contains data collected during a configurable time interval.

To display the Application Profiler tab:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Application Profiler** tab.



**TIP:** You can jump from the Application Profiler tab to the APE rulebase editor by right-clicking an application in the left pane and selecting a policy editor option. For information about using other NSM features to sort, filter, and drill down in records, see the NSM online Help.

## Protocol Profiler Tab

The Protocol Profiler tab displays information about applications that are running on your network.

Figure 150 on page 465 shows the Protocol Profiler tab.

Figure 150: Profiler Viewer: Protocol Profiler Tab

Src IP	Dst IP	User	Role	Context	Value	Src MAC	Dst MAC	Src OUI	Dst OUI	Src OS
192.168.1.21	73.78.69.84			SSL Selected Cipher Suite	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	00:04:23:B4...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Linux 2.4.12
192.168.1.106	73.78.69.84			SSL Selected Cipher Suite	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	00:04:23:A9...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.17	73.78.69.84			SSL Selected Cipher Suite	TLS_RSA_WITH_RC4_128_MD5	00:0E:0C:63...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.139	73.78.69.84			SSL Selected Cipher Suite	TLS_RSA_WITH_RC4_128_MD5	00:30:4B:5C...	4B:4F:57:53...	Supernova...	EXTERNAL H...	Unknown
192.168.1.106	73.78.69.84			SSL Selected Cipher Suite	TLS_RSA_WITH_RC4_128_MD5	00:04:23:A9...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.108	73.78.69.84			SSL Selected Cipher Suite	TLS_RSA_WITH_RC4_128_MD5	00:06:5B:0B...	4B:4F:57:53...	Dell Compute...	EXTERNAL H...	Windows 2
73.78.69.84	192.168.1.139			SSL Selected Cipher Suite	TLS_RSA_WITH_RC4_128_MD5	4B:4F:57:53...	00:30:4B:5C...	EXTERNAL H...	Supernova...	Unknown
192.168.1.17	192.168.1.255			NIDS Destination Name	001002_MSBROWSE_002001	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows 2
73.78.69.84	73.78.69.84			NIDS Destination Name	001002_MSBROWSE_002001	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
192.168.1.17	192.168.1.255			NIDS Destination Name	WORKGROUP_036	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows 2
73.78.69.84	73.78.69.84			NIDS Destination Name	WORKGROUP_036	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Source Name	0P7M0P_000	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
192.168.1.17	192.168.1.255			NIDS Source Name	YOUR-VAFDRUMM	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows 2
192.168.1.17	192.168.1.255			NIDS Source Name	YOUR-VAFDRUMM#000	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows 2
73.78.69.84	73.78.69.84			NIDS Source Name	TME-XP2	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Source Name	TME-XP2_000	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
192.168.1.17	73.78.69.84			SSL Client Version	3.1	00:0E:0C:63...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.139	73.78.69.84			SSL Client Version	3.1	00:30:4B:5C...	4B:4F:57:53...	Supernova...	EXTERNAL H...	Unknown
192.168.1.106	73.78.69.84			SSL Client Version	3.1	00:04:23:A9...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.21	73.78.69.84			SSL Client Version	3.1	00:04:23:B4...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Linux 2.4.12
192.168.1.106	73.78.69.84			SSL Client Version	3.1	00:06:5B:0B...	4B:4F:57:53...	Dell Compute...	EXTERNAL H...	Windows 2
73.78.69.84	192.168.1.139			SSL Client Version	3.1	4B:4F:57:53...	00:30:4B:5C...	EXTERNAL H...	Supernova...	Unknown
73.78.69.84	73.78.69.84			NIDS Browse Server Name	MKS00001500003770362367w9C303784003	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
192.168.1.17	192.168.1.255			NIDS Browse Server Name	YOUR-VAFDRUMM#000	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows 2
192.168.1.17	192.168.1.255			NIDS Browse Server Name	WORKGROUP0000302c346v377377	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows 2
73.78.69.84	73.78.69.84			NIDS Browse Server Name	TME-XP2000e000W000P000e000	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Browse Server Name	WORKGROUP00001070240030000220	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Browse Server Name	WORKGROUP00001070240030000260	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Browse Server Name	TME-XP2000_1226005000000000000000	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Browse Server Name	TME-XP2000_2702700500000000000000	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
73.78.69.84	73.78.69.84			NIDS Browse Server Name	TME-XP2000_2702700500000000000000	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown
192.168.1.17	73.78.69.84			SSL Server Version	3.1	00:0E:0C:63...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.139	73.78.69.84			SSL Server Version	3.1	00:30:4B:5C...	4B:4F:57:53...	Supernova...	EXTERNAL H...	Unknown
192.168.1.106	73.78.69.84			SSL Server Version	3.1	00:04:23:A9...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Windows 2
192.168.1.21	73.78.69.84			SSL Server Version	3.1	00:04:23:B4...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Linux 2.4.12
192.168.1.108	73.78.69.84			SSL Server Version	3.1	00:06:5B:0B...	4B:4F:57:53...	Dell Compute...	EXTERNAL H...	Windows 2
73.78.69.84	192.168.1.139			SSL Server Version	3.1	4B:4F:57:53...	00:30:4B:5C...	EXTERNAL H...	Supernova...	Unknown

Table 121 on page 465 describes Protocol Profiler data.

Table 121: Protocol Profiler Data

Column	Description
Src IP	Source IP address of the session.
	<b>NOTE:</b> Profiler tracks all traffic through the IDP Series device, including traffic for hosts not in your tracked hosts list. It records a value of 73.78.69.84 for the IP address for hosts not defined in the Tracked Hosts tab, such as external hosts you would not know and therefore could not configure.
Dst IP	Destination IP address.
	<b>NOTE:</b> Communication between an internal host and an external host is recorded only once. For example, the device records internal host A communicating to www.yahoo.com and www.cnn.com as one entry in the Profiler DB.
User	The user associated with the session.
Role	The role to which the user belongs.
Context	Matching contexts.
Value	Value retrieved from matching context.
Src MAC	Source MAC addresses.

Table 121: Protocol Profiler Data (*continued*)

Column	Description
Dst MAC	Destination MAC addresses.
Src OUI	Source OUI.  <b>NOTE:</b> The Organizationally Unique Identifier (OUI) value is a mapping of the first three bytes of the MAC address and the organization that owns the block of MACs. You can obtain a list of OUIs at <a href="http://standards.ieee.org/regauth/oui/oui.txt">http://standards.ieee.org/regauth/oui/oui.txt</a> .
Dst OUI	Destination OUI.
Src OS Name	Operating-system version running on the source IP.
Dst OS Name	Operating-system version running on the destination IP.
Hits	Number of occurrences that match the session.
First Time	Timestamp for the first time the device logged the event (within the specified time interval).
Last Time	Timestamp for the last time the device logged the event (within the specified time interval).
Domain	NSM domain.
Device	Device through which the session was forwarded.

By default, the Protocol Profiler tab contains only the data collected during the configured time interval.

To display the Protocol Profiler tab:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Protocol Profiler** tab.



**TIP:** For information about using NSM features to sort, filter, and drill down in records, see the NSM online Help.

## Network Profiler Tab

The Network Profiler tab displays information about the hosts in your network.

Figure 151 on page 467 shows the Network Profiler tab.

Figure 151: Profiler Viewer: Network Profiler Tab

Profiler														
Protocol Profiler		Network Profiler		Violation Viewer		Application Profiler								
Src IP	Dst IP	User	Role	Service	Access Type	Src MAC	Dst MAC	Src OUI	Dst OUI	Src OS N...	Dst OS N...	Host	First Time	L
73.78.69.84	73.78.69.84			netbios-dgm-udp	attempt	4B 4F 57 53...	4B 4F 57 53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	Sun Feb 08 ...	T
73.78.69.84	73.78.69.84			netbios-ns-udp	attempt	4B 4F 57 53...	4B 4F 57 53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	Sun Feb 08 ...	T
73.78.69.84	73.78.69.84			ntp-udp	attempt	4B 4F 57 53...	4B 4F 57 53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	Tue Feb 24 1...	T
73.78.69.84	73.78.69.84			dns-udp	attempt	4B 4F 57 53...	4B 4F 57 53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	Fri Mar 13 12...	T
73.78.69.84	73.78.69.84			UDP/1900	attempt	4B 4F 57 53...	4B 4F 57 53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	Wed Feb 11 ...	V
73.78.69.84	192.168.1.139			TCP/7008	attempt	4B 4F 57 53...	00 30 48 5C...	EXTERNAL H...	Supermicro ...	Unknown	Unknown	37989	Mon Feb 09 ...	T
73.78.69.84	192.168.1.139			TCP/8443	attempt	4B 4F 57 53...	00 30 48 5C...	EXTERNAL H...	Supermicro ...	Unknown	Unknown	37989	Mon Feb 09 ...	T
73.78.69.84	192.168.1.112			TCP/3389	attempt	4B 4F 57 53...	00 02 B3 B2...	EXTERNAL H...	Intel Corpora...	Unknown	Unknown	170954	Sun Feb 08 ...	T
73.78.69.84	192.168.1.118			TCP/3389	attempt	4B 4F 57 53...	00 50 DA 5D...	EXTERNAL H...	3COM CORP...	Unknown	Unknown	5187	Sun Feb 08 ...	T
73.78.69.84	192.168.1.166			netbios-ssn-tcp	attempt	4B 4F 57 53...	00 10 DB 92...	EXTERNAL H...	Juniper Net...	Unknown	Unknown	414	Thu Apr 02 0...	T
73.78.69.84	192.168.1.166			SMTP	attempt	4B 4F 57 53...	00 10 DB 92...	EXTERNAL H...	Juniper Net...	Unknown	Unknown	414	Thu Apr 02 0...	T
73.78.69.84	192.168.1.214			TELNET	attempt	4B 4F 57 53...	00 17 08 AB...	EXTERNAL H...	Hewlett Pack...	Unknown	Unknown	24	Tue Mar 31 1...	V
73.78.69.84	192.168.1.215			TELNET	attempt	4B 4F 57 53...	00 17 08 AB...	EXTERNAL H...	Hewlett Pack...	Unknown	Unknown	25	Tue Mar 31 1...	V
73.78.69.84	192.168.1.130			SSH	attempt	4B 4F 57 53...	00 30 48 77...	EXTERNAL H...	Supermicro ...	Unknown	Unknown	2565	Fri Mar 20 15...	V
73.78.69.84	192.168.1.30			SSH	attempt	4B 4F 57 53...	00 30 48 76...	EXTERNAL H...	Supermicro ...	Unknown	Unknown	1094	Tue Mar 31 1...	V
192.168.1.139	73.78.69.84			HTTPS	attempt	00 30 48 5C...	4B 4F 57 53...	Supermicro ...	EXTERNAL H...	Unknown	Unknown	346	Sun Feb 08 ...	T
192.168.1.139	73.78.69.84			dns-udp	attempt	00 30 48 5C...	4B 4F 57 53...	Supermicro ...	EXTERNAL H...	Unknown	Unknown	346	Sun Feb 08 ...	T
192.168.1.139	73.78.69.84			UDP/5353	attempt	00 30 48 5C...	4B 4F 57 53...	Supermicro ...	EXTERNAL H...	Unknown	Unknown	346	Mon Mar 02 ...	V
192.168.1.139	73.78.69.84			ICMP/0	attempt	00 30 48 5C...	4B 4F 57 53...	Supermicro ...	EXTERNAL H...	Unknown	Unknown	346	Mon Mar 02 ...	V
192.168.1.17	73.78.69.84			HTTPS	attempt	00 0E 0C 63...	4B 4F 57 53...	Intel Corpora...	EXTERNAL H...	Windows:20...	Unknown	32368	Sun Feb 08 ...	T
192.168.1.17	73.78.69.84			HTTP	attempt	00 0E 0C 63...	4B 4F 57 53...	Intel Corpora...	EXTERNAL H...	Windows:20...	Unknown	32368	Sun Feb 08 ...	T
192.168.1.17	73.78.69.84			dns-udp	attempt	00 0E 0C 63...	4B 4F 57 53...	Intel Corpora...	EXTERNAL H...	Windows:20...	Unknown	32368	Sun Feb 08 ...	T
192.168.1.117	192.168.1.255			netbios-dgm-udp	attempt	00 0E 0C 63...	FF FF FF FF...	Intel Corpora...	UNKNOWN H...	Windows:20...	Unknown	9152	Sun Feb 08 ...	T
192.168.1.117	192.168.1.255			netbios-ns-udp	attempt	00 0E 0C 63...	FF FF FF FF...	Intel Corpora...	UNKNOWN H...	Windows:20...	Unknown	9152	Sun Feb 08 ...	T
192.168.1.27	73.78.69.84			ntp-udp	attempt	00 04 5F 86...	4B 4F 57 53...	Evalue Tech...	EXTERNAL H...	Unknown	Unknown	11653	Sun Feb 08 ...	T

Table 122 on page 467 describes Network Profiler data.

Table 122: Network Profiler Data

Column	Description
Src IP	Source IP address of the session.  <b>NOTE:</b> Profiler tracks all traffic through the IDP Series device, including traffic for hosts not in your tracked hosts list. It records a value of 73.78.69.84 for the IP address for hosts not defined in the Tracked Hosts tab, such as external hosts you would not know and therefore could not configure.
Dst IP	Destination IP address.  <b>NOTE:</b> Communication between an internal host and an external host is recorded only once. For example, the device records internal host A communicating to www.yahoo.com and www.cnn.com as one entry in the Profiler DB.
User	The user associated with the session.
Role	The role to which the user belongs.
Service	Service for the session.
Access Type	Type of communication: <ul style="list-style-type: none"><li>Access indicates a successful connection, during which the device recorded valid requests and responses from the server to a client.</li><li>Attempt indicates a request that did not receive a reply. The device recorded a packet from a client to a server, but never saw a reply.</li><li>Probe indicates a request that does not expect a reply. For non-TCP sessions, the device recorded an ICMP error; for TCP sessions, the device recorded a SYN packet from the client followed by a RST from the server.</li></ul>
Src MAC	Source MAC addresses.
Dst MAC	Destination MAC addresses.

Table 122: Network Profiler Data (*continued*)

Column	Description
Src OUI	Source OUI.  <b>NOTE:</b> OUI stands for Organizationally Unique Identifier. This value is a mapping of the first three bytes of the MAC address and the organization that owns the block of MACs. You can obtain a list of OUIs at <a href="http://standards.ieee.org/regauth/oui/oui.txt">http://standards.ieee.org/regauth/oui/oui.txt</a> .
Dst OUI	Destination OUI.
Src OS Name	Operating-system version running on the source IP.
Dst OS Name	Operating-system version running on the destination IP.
Hits	Number of occurrences that match.
First Time	Timestamp for the first time the device logged the event (within the specified time interval).
Last Time	Timestamp for the last time the device logged the event (within the specified time interval).
Domain	NSM domain.
Device	Device through which the session was forwarded.

To display the Network Profiler tab:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Network Profiler** tab.



**TIP:** For information about using NSM features to sort, filter, and drill down in records, see the NSM online Help.

## Violation Viewer Tab

The Violation Viewer tab displays a filtered view of the Network Profiler data. In the Violation Viewer tab, you configure permitted objects. Permitted objects are filtered from the display, allowing you to focus on unpermitted traffic.

Figure 152 on page 469 shows the Violation Viewer tab.



Figure 152: Profiler Viewer: Violation Viewer Tab

Src IP	Dest IP	User	Role	Service	Access T...	Src MAC	Dest MAC	Src OUI	Dest OUI	Src OS N	Dest OS N	Hts	Filter Criteria (Double Click to Activate)
192.168.1.17	192.168.1.255			netbios-...	attempt	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Windows:20...	Unknown	9152	
192.168.1.27	73.78.69.84			rtp-udp	attempt	00:04:5F:06...	4B:4F:57:53...	Evalue Tech...	EXTERNAL H...	Unknown	Unknown	11663	
192.168.1.112	73.78.69.84			TCP/80	attempt	00:02:B3:B2...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Unknown	Unknown	765585	
192.168.1.112	73.78.69.84			netbios-...	attempt	00:02:B3:B2...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Unknown	Unknown	765585	
192.168.1.112	73.78.69.84			IP41	attempt	00:02:B3:B2...	4B:4F:57:53...	Intel Corpora...	EXTERNAL H...	Unknown	Unknown	765585	
73.78.69.84	73.78.69.84			UDP/1900	attempt	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	
73.78.69.84	73.78.69.84			rtp-udp	attempt	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	
73.78.69.84	73.78.69.84			netbios-...	attempt	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	
73.78.69.84	73.78.69.84			netbios-...	attempt	4B:4F:57:53...	4B:4F:57:53...	EXTERNAL H...	EXTERNAL H...	Unknown	Unknown	18505	
192.168.1.221	73.78.69.84			rtp-udp	attempt	00:21:59:90...	4B:4F:57:53...	Juniper Net...	EXTERNAL H...	Unknown	Unknown	4852	
192.168.1.138	73.78.69.84			UDP/1707	attempt	00:30:48:5F...	4B:4F:57:53...	Supermicro...	EXTERNAL H...	Unknown	Unknown	6671	
192.168.1.138	73.78.69.84			UDP/4570	attempt	00:30:48:5F...	4B:4F:57:53...	Supermicro...	EXTERNAL H...	Unknown	Unknown	6671	
192.168.1.222	73.78.69.84			rtp-udp	attempt	00:21:59:90...	4B:4F:57:53...	Juniper Net...	EXTERNAL H...	Unknown	Unknown	4819	
192.168.1.17	192.168.1.255			netbios-...	attempt	00:0E:0C:63...	FF:FF:FF:FF...	Intel Corpora...	UNKNOWN H...	Unknown	Unknown	2251	
192.168.2.242	192.168.2.255			netbios-...	attempt	00:0C:29:25...	FF:FF:FF:FF...	VMware, Inc.	UNKNOWN H...	Unknown	Unknown	2333	
192.168.2.242	192.168.2.255			netbios-...	attempt	00:0C:29:25...	FF:FF:FF:FF...	VMware, Inc.	UNKNOWN H...	Unknown	Unknown	2333	
192.168.200.1	192.168.1.166			netbios-...	attempt	00:04:23:BD...	00:10:0B:92...	Intel Corpora...	Juniper Net...	Windows:20...	Unknown	9012	
192.168.2.200	192.168.1.166			LDAP	attempt	00:04:23:BD...	00:10:0B:92...	Intel Corpora...	Juniper Net...	Windows:20...	Unknown	9249	
192.168.2.200	192.168.1.166			SMB	attempt	00:04:23:BD...	00:10:0B:92...	Intel Corpora...	Juniper Net...	Windows:20...	Unknown	9249	
192.168.220.1	192.168.1.166			netbios-...	attempt	00:04:23:BD...	00:10:0B:92...	Intel Corpora...	Juniper Net...	Windows:20...	Unknown	9011	
172.19.100.99	172.19.100.2...			netbios-...	attempt	00:13:20:21...	FF:FF:FF:FF...	Intel Corporate	UNKNOWN H...	Unknown	Unknown	4268	
172.19.100.99	172.19.100.2...			netbios-...	attempt	00:13:20:21...	FF:FF:FF:FF...	Intel Corporate	UNKNOWN H...	Unknown	Unknown	4268	
0.0.0.0	255.255.255...			bootp-se	attempt	00:17:0B:AB...	FF:FF:FF:FF...	Hewlett Pack...	UNKNOWN H...	Unknown	Unknown	2523	
192.168.2.200	172.17.27.46			rtp-udp	attempt	00:04:23:BD...	00:10:0B:92...	Intel Corpora...	Juniper Net...	Windows:20...	Unknown	410	

To configure permitted objects:

1. In the NSM navigation tree, select **Investigate > Security Monitor > Profiler**.
2. Click the **Violation Viewer** tab.
3. Click the + icon that appears on the top of the right-hand window to display the New Permitted Object dialog box.
4. Type a name for the permitted object.
5. Within the New Permitted Object dialog box, click the + icon to add rows for rules.
6. Use the selection controls to select the desired address objects or service objects and click **OK**.
7. Click **OK** to save the permitted object.

The object appears in the filters windows within the Violation Viewer tab.

8. Select the object and click **Apply** to filter matching objects from the display.



**TIP:** For information about using additional NSM features to sort, filter, and drill down in records, see the NSM online Help.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Profiler Task Summary on page 203](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Profiler Overview on page 21](#)

## Viewing NSM Predefined Reports (NSM Procedure)

**Purpose** You can use the predefined reports to validate the effectiveness of your security policies.

Table 12 on page 30 describes NSM DI/IDP predefined reports. These reports are related to attacks.

**Table 123: NSM DI/IDP Predefined Reports**

Report	Description
Top 100 Attacks (last 24 hours)	Those attacks that are detected most frequently within the last 24 hours.
Top 100 Attacks Prevented (last 24 hours)	Those attacks that are prevented most frequently within the last 24 hours.
Top 20 Attackers (All Attacks - last 24 hours)	IP addresses that have most frequently been the source of an attack during the last 24 hours.
Top 20 Attackers Prevented (All Attacks - last 24 hours)	IP addresses that have most frequently been prevented from attacking the network during the last 24 hours.
Top 20 Targets (last 24 hours)	IP addresses that have most frequently been the target of an attack during the last 24 hours.
Top 20 Targets Prevented (last 24 hours)	IP addresses that have most frequently prevented attacks during the last 24 hours.
All Attacks by Severity (last 24 hours)	Number of attacks by severity level (set in attack objects).
All Attacks Prevented by Severity (last 24 hours)	Number of attacks prevented by severity level (set in attack objects).
All Attacks Over Time (last 7 days)	All attacks detected during the last 7 days.
All Attacks Prevented Over Time (last 7 days)	All attacks prevented during the last 7 days.
All Attacks Over Time (last 30 days)	All attacks detected during the last 30 days.
All Attacks Prevented Over Time (last 30 days)	All attacks prevented during the last 30 days.
Critical Attacks (last 24 hours)	All attacks categorized as "critical" detected during the past 24 hours.
Critical Attacks Prevented (last 24 hours)	All attacks categorized as "critical" prevented during the past 24 hours.
Critical through Medium Attacks (last 24 hours)	All attacks categorized as either "critical" or "medium" detected during the past 24 hours.
Critical through Medium Attacks Prevented (last 24 hours)	All attacks categorized as either "critical" or "medium" prevented during the past 24 hours.
Top 50 Scan Sources (last 7 days)	IP addresses that have most frequently performed a scan of a managed device.
Top 50 Scan Targets (last 7 days)	IP addresses that have most frequently been the target of a scan over the last 7 days.
Profiler - New Hosts (last 7 days)	New Hosts listed in the Profiler over the last 7 days.

**Table 123: NSM DI/IDP Predefined Reports (*continued*)**

Report	Description
Profiler - New Ports (last 7 days)	New Ports listed in the Profiler over the last 7 days.
Profiler - New Protocols (last 7 days)	New Protocols listed in the Profiler over the last 7 days.
Top IDP Rules	The total number of log entries generated by specific rules in your IDP security policies. You can use the Top Rules report to identify those rules that are generating the most log events. This enables you to better optimize your rulebases by identifying those rules that are most and least effective. You can then modify or remove those rules from your security policies.

[Table 123 on page 470](#) describes Profiler predefined reports. These reports are related to activity by hosts in your network.

**Table 124: NSM Profiler Predefined Reports**

Report	Description
Top 10 Peers by Count	Ten source and destination IP addresses that appeared most frequently in the Profiler logs.
Top 10 Peers with maximum hits	Ten source and destination IP addresses that had highest number of hits in the Profiler logs.

[Table 125 on page 471](#) describes the predefined application volume tracking (AVT) reports. The reports are related to application use in your network.

**Table 125: NSM: Application Volume Tracking Reports**

Report	Description
Top 10 Applications by Volume	Applications with the highest volume in bytes in the past 24 hours.
Top 10 Application Categories by Volume	Application categories with the highest volume in bytes in the past 24 hours.
Top 5 Applications by Volume over Time (last 1 hour)	Applications with the highest volume in bytes in the past 1 hour.
Top 5 Application Categories by Volume (last 1 hour)	Application categories with the highest volume in bytes in the past 1 hour.
Top 5 Source by Volume over Time (last 1 hour)	Source IP addresses with the highest volume in bytes in the past 1 hour.
Top 5 Destination by Volume over Time (last 1 hour)	Destination IP addresses with the highest volume in bytes in the past 1 hour.

**Action** To view predefined reports:

1. In the NSM navigation tree, select **Investigate > Report Manager**.
2. Expand one of the following report nodes related to IDP events.
  - DI/IDP Reports
  - Profiler
  - AVT
3. Click the name of a report to display its contents.



**TIP:** For details on modifying report options, see the NSM online Help.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

---

## Creating NSM Custom Reports (NSM Procedure)

**Purpose** You use custom reports if you require a view of data not covered by predefined reports.

**Action** To create a custom report:

1. In the NSM navigation tree, select **Investigate > Report Manager**.
2. Select a predefined report with data similar to what you ultimately want to save.
3. Select **File > Save As**.
4. Use the predefined report as a template and example. Complete the configuration options described in [Table 126 on page 473](#).
5. Click **OK**.

Table 126: Custom Report Configuration Options

Tab	Field	Description
General	Name	Specify a name for the report as you would like it to appear in the NSM navigation tree.
	Report Title	Specify a name for the report as you would like it to appear at the top of the report.
	Type of Report	<p>Select a report type:</p> <ul style="list-style-type: none"> <li>• <b>Count-Based</b>—Displays total current activity to date. For example, the Top Scan Targets report is a count-based report that displays the total number of scans currently recorded against a specified number of destination IP addresses.</li> <li>• <b>Time-Based</b>—Displays activity over time. For example, the Attacks Over Time report is a time-based report that measures the top attacks recorded in log records over a specified period.</li> <li>• <b>Sum-Based</b>—Displays sum-based data.</li> </ul>
	Columns for Report	In reports, columns are the same as log fields.
	Time Period	<p>You can configure a report to display all available data from either a specific date and time or during a specific time interval. For example, if you suspect that your network was attacked on September 15 at 6:00 PM, you could set the Starting At Time Period Duration report field in the options on a Top Screen Attacks report to that time, then generate the report. If you are not sure of the exact date or time of the attack, but know it occurred during the past 2 days, set the Duration field in the Time Period Duration report options on a Top Screen Attacks report to two days, then generate the report.</p> <p><b>NOTE:</b> The data that you can display in each report is limited by the amount of log information available.</p>
	Data point count	<p>Typically, the top 50 occurrences of each data type are displayed in each report. You can configure a report to display more or fewer data points depending upon the level of detail you need. For example, if you want to obtain a more precise view of the top occurrences of events, you would configure a lower data point count (such as 25).</p> <p><b>NOTE:</b> The minimum data point count that you can configure in all reports is 5; the maximum data point count is 200.</p>
	Chart type	<p>Select from the following choices:</p> <ul style="list-style-type: none"> <li>• Horizontal bar (default)</li> <li>• Pie</li> <li>• Line</li> <li>• Vertical bar</li> </ul>
	Save Report In	<p>In the first selection box, specify whether to save in the My Reports or Shared Reports node.</p> <p>In the second box, select the <b>Others</b> folder or type a new folder name.</p>
Filter	Columns for Report	The columns you selected on the General tab are passed through. Select the column with the cursor to display the corresponding Filter Settings controls.
	Filter Settings	Specify filter values related to column settings.



**TIP:** For information on deleting custom reports, organizing report folders, exporting reports, and using the NSM `guiSvrCli.sh` command and Linux `cron` command to automate reporting jobs, see the NSM online Help.

**Related  
Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing NSM Predefined Reports \(NSM Procedure\) on page 469](#)
- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)

## CHAPTER 42

# Packet Logging

- [Example: Packet Logging Workflow on page 475](#)
- [Using tcpdump to Capture Packets on page 481](#)
- [Using jnetTcpdump to Capture Packets on page 481](#)

### Example: Packet Logging Workflow

---

This topic summarizes IDP Series packet logging basics. It includes the following sections:

- [Using Packet Captures on page 475](#)
- [Enabling Packet Capture in Security Policy Rules on page 476](#)
- [Forwarding Packet Capture Logs to NSM on page 477](#)
- [Viewing Packet Capture Logs on page 477](#)

### Using Packet Captures

The IDP solution supports packet capture logging triggered by security policy rules.

You can use packet captures for a number of response activities, including:

- Validation of the security policy rule and attack object. You may choose to enable packet logging to test a new attack object. Once verified, you may find packet logging for the rule unnecessary.
- Further analysis of traffic surrounding the matching event. The surrounding traffic might provide information that helps you determine whether you need to take further steps to protect the target or whether the attack should be considered a false positive.
- Reproducibility and documentation for Internet security groups, including the Juniper Networks Security Center.
- Legal evidence. Consult with your legal counsel for guidance on how local laws and rules of evidence apply if you want to use packet capture data as evidence in the prosecution of attackers.

## Enabling Packet Capture in Security Policy Rules

When traffic matches a rule where packet logging is configured, the IDP Series device captures the packet that matched the rule, as well as the preceding and trailing packets (according to your configured preference).

To enable packet logging within a security policy rule, use the Security Policy editor. Right-click a cell in the Notification column and select **Configure** to display the dialog box where you can set packet logging options.

Figure 153: Notification Options: Packet Logging

The 'Configure Notification' dialog box has a 'Logging' tab selected. It includes a 'Logging' checkbox (checked), an 'Alert' checkbox (unchecked), and a 'Log Packets' checkbox (checked). Below these are three input fields: 'Packets Before' (10), 'Packets After' (10), and 'Post window timeout(secs)' (1).

In the NSM Log Viewer, logs for events where packet captures have been generated are noted by an icon in the Has Packet Data column (the last column in Figure 43 on page 146).

Figure 154: NSM Log Viewer: Has Packet Data Column

Log Viewer (3-10P-00)

Src Addr	Dest Addr	Action	Protocol	Port	Port #	Rule #	Net Src Addr	Net Dest Addr	Details	Category	Subcategory	Severity	Device	Comment	Has Packet Data
1.1.0.86	1.2.0.40	Conn Dropped	TCP	554	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	RTSP: Real Server Describe Overl...	Major	DP8202			
1.1.0.28	1.2.0.26	Conn Dropped	TCP	554	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	RTSP: Real Server Transport Over...	Major	DP8202			
1.1.0.206	1.2.0.111	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	HTTP: Cisco IOS HTTP Configuratio...	Major	DP8202			
1.1.0.56	1.2.0.34	Conn Dropped	TCP	25	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SMTP: Email Domain Name	Major	DP8202			
1.1.0.56	1.2.0.34	Conn Dropped	TCP	25	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SMTP: Email Address	Major	DP8202			
1.1.0.48	1.2.0.159	Conn Dropped	TCP	25	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SMTP: Email Address	Major	DP8202			
1.1.0.80	1.2.0.175	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	HTTP: Cisco IOS HTTP Configuratio...	Major	DP8202			
1.1.0.110	1.2.0.192	Conn Dropped	TCP	25	3	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SMTP: Microsoft Exchange Mailer...	Device_critical_log	DP8202			
1.1.0.118	1.2.0.198	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: VUL-FTP (plogob) Input Valid...	Major	DP8202			
1.1.0.230	1.2.0.123	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Username Too Long	Major	DP8202			
1.1.0.62	1.2.0.168	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Username Too Long	Major	DP8202			
1.1.0.231	1.2.0.116	Conn Dropped	TCP	25	3	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SMTP: Exchange Multiple Long Mail...	Device_critical_log	DP8202			
1.1.0.4	1.2.0.133	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Username Too Long	Major	DP8202			
1.1.0.67	1.2.0.161	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SHELLCODE: X86 NOOP (TCP)	Major	DP8202			
1.1.0.81	1.2.0.168	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Username Too Long	Major	DP8202			
1.1.0.150	1.2.0.208	Conn Dropped	TCP	25	3	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	SMTP: Sendmail Oversized Address...	Device_critical_log	DP8202			
1.1.0.231	1.2.0.116	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Username Too Long	Major	DP8202			
1.1.0.112	1.2.0.193	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: (psvch) W3 FTP Server FTP...	Major	DP8202			
1.1.0.224	1.2.0.122	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Pathname Too Long	Major	DP8202			
1.1.0.17	1.2.0.136	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Username Too Long	Major	DP8202			
1.1.0.224	1.2.0.122	Conn Dropped	TCP	21	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	FTP: Pathname Too Long	Major	DP8202			
1.1.0.223	1.2.0.239	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	HTTP: Missing HTTP Version	Major	DP8202			
1.1.0.98	1.2.0.184	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	HTTP: Microsoft VM/Info Buffer Ov...	Major	DP8202			
1.1.0.98	1.2.0.184	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	HTTP: Missing HTTP Version	Major	DP8202			
1.1.0.56	1.2.0.163	Conn Dropped	TCP	119	3	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	NNTP: XPAT Pattern Overflow	Device_critical_log	DP8202			
1.1.0.222	1.2.0.244	Conn Dropped	TCP	80	4	0.0.0.0	0.0.0.0	Interface-eth2/	Predifined	HTTP: All-N-WebAdmin USER Buff...	Major	DP8202			

Summary: All Fields | Whole Lookup | Outside Report

Predefined: SMTP: Microsoft Exchange Malformed Intra-Exchange Verb

References: This signature detects attempts to exploit a known vulnerability in Microsoft Exchange Server 6.5 and 2000. The command verb "AccessMail" which is valid only for communication between validated Exchange servers, is handled incorrectly. Attackers can send the command verb with a negative number or a very large positive number to crash the Exchange server, and, in extreme cases with Exchange Server 2000, can also take control of the server.

Matching Data Snippet

HEX	ASCII

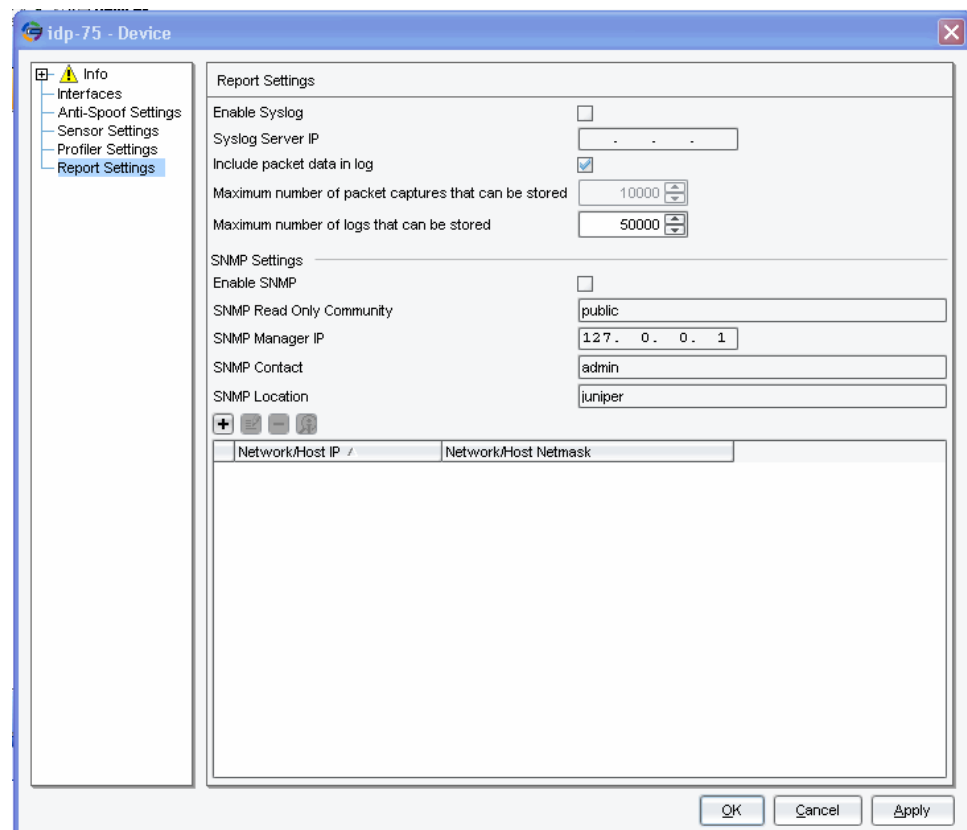


## Forwarding Packet Capture Logs to NSM

The IDP Series device writes packet captures locally to subdirectories of `/usr/idp/device/var/pktlogs/`. It forwards the packet data to NSM according to your NSM Report Settings:

- **Include packet data in log** selected. Forwards the packet capture to NSM automatically whenever it sends the corresponding event log.
- **Include packet data in log** not selected. Forwards a reference to the packet capture file to NSM automatically but forwards the packet data itself only on-demand (when an NSM user takes action to display the packet data).

Figure 155: NSM Device Configuration Editor: Report Settings



## Viewing Packet Capture Logs

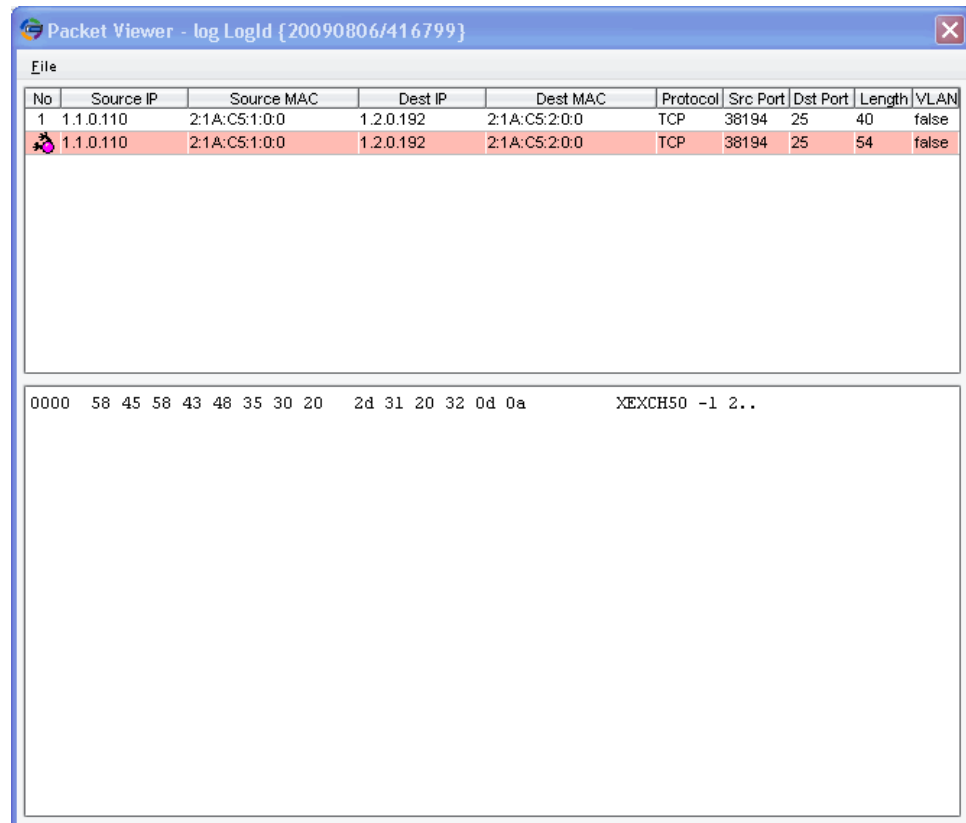
You have two options for viewing packet captures:

- [Using the NSM Packet Viewer on page 478](#)
- [Using an External Viewer to View Packet Data on page 478](#)

### Using the NSM Packet Viewer

The NSM packet viewer displays the offending attack payload that triggered the alert as well as preceding and trailing packets (according to your configuration). [Figure 45 on page 148](#) shows the NSM packet capture viewer.

Figure 156: NSM Packet Capture Viewer



To view a packet capture in the NSM packet viewer:

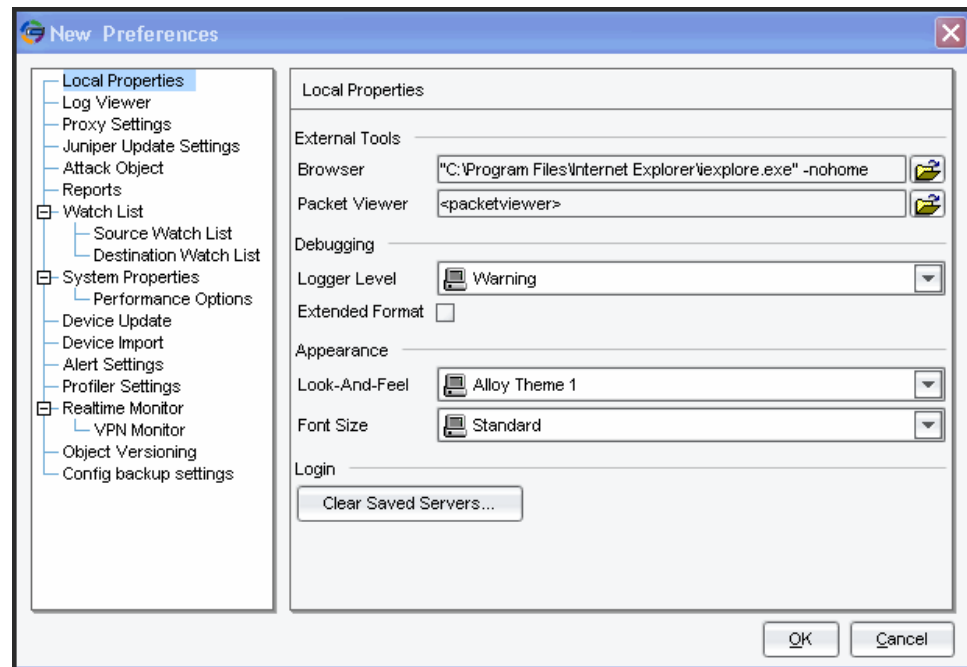
1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined > DI/IDP** to display the IDP table.
2. Select **View > Choose Columns** to display the dialog box you use to show and hide log table columns.
3. Select **Has Packet Data** to show this column.  
If a security event log has packet data, an icon appears in the table cell under this column.
4. Double-click the Has Packet data icon to display the packet data in the NSM packet viewer.

### Using an External Viewer to View Packet Data

You can configure NSM to launch an external viewer for packet captures.

Figure 46 on page 149 shows the NSM dialog box where you can specify the location of an external packet viewer.

Figure 157: Specifying an External Viewer

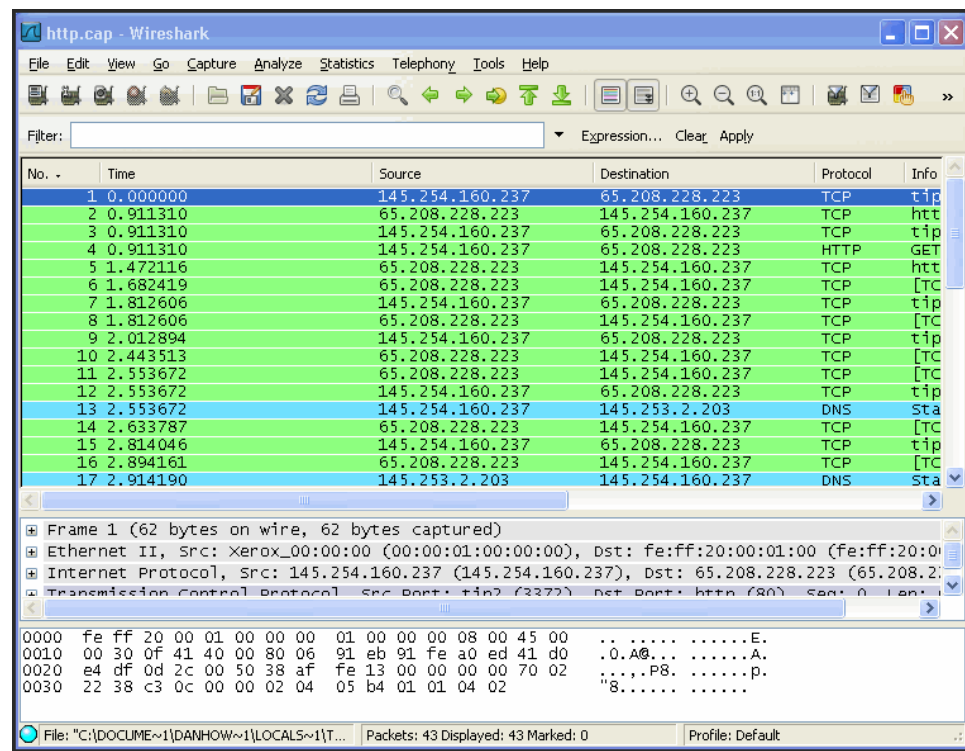


To set the location of the external viewer:

1. In NSM, select **Tools > Preferences**.
2. Select **Local Properties**.
3. Under External Tools > Packet Viewer, click the browse button and select the executable file for the external viewer (for example: **C:\Program Files\Wireshark\wireshark.exe**).
4. Click **OK** to close the New Preferences dialog box.

Figure 47 on page 150 shows packet data displayed in the Wireshark packet viewer.

Figure 158: Wireshark Packet Viewer



To view a packet capture in an external packet viewer:

1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined > DI/IDP** to display the IDP table.
2. Select **View > Choose Columns** to display the dialog box you use to show and hide log table columns.
3. Select **Has Packet Data** to show this column.  
If a security event log has packet data, an icon appears in the table cell under this column.
4. Right-click the Has Packet data icon and select **Show > Packet Data in External Viewer**.

#### Related Documentation

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Understanding IDP Rulebase Notification Options on page 65](#)
- [Using tcpdump to Capture Packets on page 481](#)

The following related topics are included in the *IDP Series Administration Guide*:

- [Specifying Rule Notification Options \(NSM Procedure\) on page 221](#)
- [Enabling Collection of Packet Data in NSM Logs \(NSM Procedure\) on page 303](#)

## Using tcpdump to Capture Packets

The IDP OS includes a Linux version of the commonly used **tcpdump** utility. On IDP Series devices, the **tcpdump** utility can capture only received packets (Rx packets). If you need to capture transmitted packets (Tx packets), use the **jnetTcpdump** utility.

To display a reference of **tcpdump** options and Berkeley Packet Filter (BFT) primitive expressions, enter **man tcpdump**.

The following example shows the syntax for capturing SMTP traffic on port 25. Here, **tcpdump** starts listening on the eth1 interface for traffic matching the expression **tcp port 25**.

```
[root@localhost ~]# tcpdump -i eth1 -s 0 -w /tmp/smtp.pcap tcp port 25
```

The following example shows the syntax for capturing all traffic except your SSH session to the IDP Series device:

```
[root@localhost ~]# tcpdump -s 0 -l eth2 -w eth2-all-but-ssh.pcap not tcp port 22
```

If you later decide you want to extract only HTTP traffic from the “all-but” pcap, you can use the following syntax to filter the previously collected file:

```
[root@localhost ~]# tcpdump -r eth2-all-but-ssh.pcap -w http.pcap tcp port 80
```

To view captured traffic, you can use **tcpdump** data display options or use a packet viewer, such as Wireshark.

### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Using jnetTcpdump to Capture Packets on page 481](#)
- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Example: Packet Logging Workflow on page 145](#)

## Using jnetTcpdump to Capture Packets

Typically, when you want to capture the packet data surrounding a security event, you configure the packet logging option in security policy rules. In the course of monitoring your network, you might encounter suspicious traffic where you have not set up rule-based packet capture. In these cases, you can use the **jnetTcpdump** utility or the Linux-based **tcpdump** utility to capture the traffic.

The **jnetTcpdump** utility copies packets from the JNET driver packet queuing module. This allows it to capture packets as they are received (Rx packets) or as they are transmitted (Tx packets). In contrast, on the IDP Series device, the **tcpdump** utility can capture only Rx packets. The command options for the **jnetTcpdump** utility are similar to the standard **tcpdump** utility options (though there are fewer options).

The following example starts listening on interface eth4 for packets with a destination IP address of 4.0.0.4:

```
[root@localhost ~]# jnetTcpdump -i eth4 -f 4.0.0.4 dst
jnetPassiveAttach done
jnet tcpdump Started on eth4 for both Receive & Transmit side
Filter enabled - Host:4.0.0.4 as dst
0 50 56 a4 21 6c 0 50 56 a4 d 9 8 0 45 0 0 54 0 0 40 0 40 1 32 a3 4 0 0 3 4 0 0
4 8 0 55 8e 8e 4f 0 0
ba 9f 3e 4d 21 32 f 0 8 9 a b c d e f 10 11 12 13 14 15
0 50 56 a4 21 6c 0 50 56 a4 d 9 8 0 45 0 0 54 0 0 40 0 40 1 32 a3 4 0 0 3 4 0 0
4 8 0 97 88 8e 4f 0 1
bb 9f 3e 4d de 36 f 0 8 9 a b c d e f 10 11 12 13 14 15
Done...No of Packet Captured is 2
No of Packets filtered-out 2
```

Type Ctrl-C to stop the capture.

To view captured traffic, you can use **tcpdump** data display options or use a packet viewer, such as Wireshark.

**Related  
Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [jnetTcpdump](#)
- [Using tcpdump to Capture Packets on page 481](#)
- [Connecting to the Command-Line Interface \(CLI Procedure\) on page 192](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Example: Packet Logging Workflow on page 145](#)

## CHAPTER 43

# Using the `bypassStatus` Utility to Monitor the Internal Bypass Daemon

- [bypassStatus Utility Task Summary on page 483](#)

### **bypassStatus Utility Task Summary**

---

You use the **bypassStatus** command to monitor the NIC state for traffic interfaces. The **bypassStatus** command displays the state of the bypass daemon, the watchdog timer setting, the watchdog timer reset interval, and the state of each interface pair.

The **bypassStatus** utility is located in the `/usr/idp/device/utls/` directory.

#### **Related Documentation**

- [bypassStatus Command Reference on page 484](#)

## bypassStatus Command Reference

**Syntax** `bypassStatus interval iterations`

**Description** Displays status of internal bypass processes.

**Options** [Table 127 on page 484](#) describes **bypassStatus** options and arguments and provides examples of command syntax.

**Table 127: Command Reference: bypassStatus**

Options	Usage and Examples																								
[none]	<p>Displays the status of the NIC bypass daemon, settings for the watchdog timer, and the status of traffic interfaces.</p> <p>For an overview of the NIC bypass feature, see the <i>IDP Series Concepts and Examples Guide</i>.</p> <p>The watchdog timer setting and watchdog loop interval are default settings in the <b>idp.cfg</b> file. You should not change these settings unless advised to do so by JTAC.</p> <p>The current state reported is <b>Normal</b> when the operating system is available and the virtual router receives its reset signal from the interface within the threshold (3 seconds); <b>NICs off</b> or <b>Bypass</b> when the timer has not received its reset signal within the threshold.</p> <p>The following example shows the output of the <b>bypassStatus</b> command on IDP200, IDP600, and IDP1100 appliances:</p> <pre>[root@localhost ~]# bypassStatus BYPASS STATUS: Thu Sep 16 10:50:29 PDT 2010 Status for nicBypass daemon      : on Watchdog timer setting(sec)      : 8 Watchdog loop reset interval(sec): 200000</pre> <table><tr><th>NIC</th><th>Status</th><th>Current State</th><th>WD Time Left(ms)</th></tr><tr><td>eth2,eth3</td><td>ENABLED</td><td>Normal</td><td>12168</td></tr><tr><td>eth4,eth5</td><td>ENABLED</td><td>Normal</td><td>12112</td></tr><tr><td>eth6,eth7</td><td>ENABLED</td><td>Normal</td><td>12040</td></tr><tr><td>eth8,eth9</td><td>ENABLED</td><td>Normal</td><td>11984</td></tr><tr><td>eth10,eth11</td><td>ENABLED</td><td>Normal</td><td>12116</td></tr></table> <p><b>NOTE:</b> The watchdog loop reset interval is 200,000 microseconds, not seconds.</p>	NIC	Status	Current State	WD Time Left(ms)	eth2,eth3	ENABLED	Normal	12168	eth4,eth5	ENABLED	Normal	12112	eth6,eth7	ENABLED	Normal	12040	eth8,eth9	ENABLED	Normal	11984	eth10,eth11	ENABLED	Normal	12116
NIC	Status	Current State	WD Time Left(ms)																						
eth2,eth3	ENABLED	Normal	12168																						
eth4,eth5	ENABLED	Normal	12112																						
eth6,eth7	ENABLED	Normal	12040																						
eth8,eth9	ENABLED	Normal	11984																						
eth10,eth11	ENABLED	Normal	12116																						



Table 127: Command Reference: `bypassStatus` (*continued*)

Options	Usage and Examples
	<p>The following examples show the output of the <code>bypassStatus</code> command on IDP75, IDP250, IDP800, and IDP8200 appliances:</p> <pre>[root@idp51 ~]# bypassStatus BYPASS STATUS: Wed Apr  8 10:08:38 PDT 2009 Status for nicBypass daemon      : on Watchdog timer setting(sec)      : 3 Watchdog loop reset interval(sec): 200000</pre> <pre>NIC                Current State ----- eth2,eth3          Normal or NICs off eth4,eth5          Normal eth6,eth7          Normal eth12,eth13        Normal eth14,eth15        Normal [root@idp51 ~]#</pre> <pre>[root@defaultthost ~]# bypassStatus  BYPASS STATUS: Wed Mar  4 14:47:01 IST 2009 Status for nicBypass daemon      : off Watchdog timer setting(sec)      : 3 Watchdog loop reset interval(sec): 200000</pre> <pre>NIC                Current State ----- eth2,eth3          Bypass or NICs off eth4,eth5          Bypass or NICs off eth6,eth7          Bypass or NICs off eth10,eth11        Bypass or NICs off eth12,eth13        Normal</pre> <p><b>NOTE:</b> Due to a hardware limitation, the <code>bypassStatus</code> command does not report status or the watchdog timer time left for IDP75, IDP250, IDP800, and IDP8200 appliances.</p>
<code>interval</code>	<p>Specifies a number of seconds at which to refresh the status display. The status is updated every <code>interval</code> seconds until you press Ctrl-C.</p> <pre>[root@defaultthost admin]# bypassStatus 3</pre>

Table 127: Command Reference: `bypassStatus` (*continued*)

Options	Usage and Examples																																																																												
<i>iterations</i>	<p>If you specify an interval, you can also specify a number of iterations after which the status display exits and returns to the command prompt.</p> <pre>[root@defaulthost admin]# <b>bypassStatus 3 3</b> BYPASS STATUS: Wed Oct 22 18:47:03 EDT 2008 Status for nicBypass daemon      : on Watchdog timer setting(sec)      : 3 Watchdog loop reset interval(sec): 200000</pre> <table><tr><th>NIC</th><th>Status</th><th>Current State</th><th>WD Time Left(ms)</th></tr><tr><td colspan="4">-----</td></tr><tr><td>eth2,eth3</td><td>ENABLED</td><td>Normal</td><td>3110</td></tr><tr><td>eth4,eth5</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth6,eth7</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth8,eth9</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth10,eth11</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td colspan="4">-----</td></tr><tr><td>eth2,eth3</td><td>ENABLED</td><td>Normal</td><td>3060</td></tr><tr><td>eth4,eth5</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth6,eth7</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth8,eth9</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth10,eth11</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td colspan="4">-----</td></tr><tr><td>eth2,eth3</td><td>ENABLED</td><td>Normal</td><td>3000</td></tr><tr><td>eth4,eth5</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth6,eth7</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth8,eth9</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr><tr><td>eth10,eth11</td><td>disabled</td><td>Normal</td><td>(wdt inactive)</td></tr></table>	NIC	Status	Current State	WD Time Left(ms)	-----				eth2,eth3	ENABLED	Normal	3110	eth4,eth5	disabled	Normal	(wdt inactive)	eth6,eth7	disabled	Normal	(wdt inactive)	eth8,eth9	disabled	Normal	(wdt inactive)	eth10,eth11	disabled	Normal	(wdt inactive)	-----				eth2,eth3	ENABLED	Normal	3060	eth4,eth5	disabled	Normal	(wdt inactive)	eth6,eth7	disabled	Normal	(wdt inactive)	eth8,eth9	disabled	Normal	(wdt inactive)	eth10,eth11	disabled	Normal	(wdt inactive)	-----				eth2,eth3	ENABLED	Normal	3000	eth4,eth5	disabled	Normal	(wdt inactive)	eth6,eth7	disabled	Normal	(wdt inactive)	eth8,eth9	disabled	Normal	(wdt inactive)	eth10,eth11	disabled	Normal	(wdt inactive)
NIC	Status	Current State	WD Time Left(ms)																																																																										
-----																																																																													
eth2,eth3	ENABLED	Normal	3110																																																																										
eth4,eth5	disabled	Normal	(wdt inactive)																																																																										
eth6,eth7	disabled	Normal	(wdt inactive)																																																																										
eth8,eth9	disabled	Normal	(wdt inactive)																																																																										
eth10,eth11	disabled	Normal	(wdt inactive)																																																																										
-----																																																																													
eth2,eth3	ENABLED	Normal	3060																																																																										
eth4,eth5	disabled	Normal	(wdt inactive)																																																																										
eth6,eth7	disabled	Normal	(wdt inactive)																																																																										
eth8,eth9	disabled	Normal	(wdt inactive)																																																																										
eth10,eth11	disabled	Normal	(wdt inactive)																																																																										
-----																																																																													
eth2,eth3	ENABLED	Normal	3000																																																																										
eth4,eth5	disabled	Normal	(wdt inactive)																																																																										
eth6,eth7	disabled	Normal	(wdt inactive)																																																																										
eth8,eth9	disabled	Normal	(wdt inactive)																																																																										
eth10,eth11	disabled	Normal	(wdt inactive)																																																																										

**Additional Information** Due to a hardware limitation with IDP800 and IDP8200 I/O modules, the current state reported by the `bypassStatus` command is ambiguous for these interfaces. A bypass NIC has two physical states: on or off. The programmed behavior supports three possibilities: normal, bypass, or NICs off.

For fiber interfaces, the `bypassStatus` command reliably reports **Bypass** when the NIC is in bypass state. It reports **Normal or NICs off** when not in bypass.

For copper interfaces, the `bypassStatus` command reliably reports **Normal** during normal operations. It reports **Bypass or NICs off** when not normal.

When using the `bypassStatus` command, we recommend you refresh your recollection of your NIC state settings and use logs to disambiguate the `bypassStatus` output. For example, if you know you have not configured **NICs off** but instead always use **Bypass**, you can disambiguate the status. In addition, when you shutdown the IDP Series appliance, it sends a log to NSM that clearly indicates the action to change NIC state to your specified setting.

## CHAPTER 44

# Using the sctop Utility to Monitor Session Flow

- [sctop Task Summary on page 487](#)
- [Using the sctop Utility \(CLI Procedure\) on page 487](#)
- [Understanding sctop Flow Table Reports on page 490](#)

### sctop Task Summary

---

You use the **sctop** command to monitor the IDP system connection tables and view IDP system status.

The **sctop** utility is located in the **/usr/bin** directory.

You use the **sctop** utility to perform the following tasks:

- Display the ARP and MAC tables.
- Display the IP, ICMP, TCP, and UDP flow tables.
- Display STP information.
- Display RPC information.
- Display performance statistics for traffic, IDP security policies, and IDP processors.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Using the sctop Utility \(CLI Procedure\) on page 487](#)

### Using the sctop Utility (CLI Procedure)

---

**Purpose** You use the **sctop** command to monitor session information.

**Action** To connect to the command-line interface and use the **sctop** utility:

1. Use SSH to connect to the IP address or hostname for the management interface.
2. Log into the CLI as **admin** and enter **su -** to switch to **root**.
3. At the secure shell, define the IDPDIR:

```
IDPDIR=/usr/idp
export IDPDIR
```



**NOTE:** Bash is the default shell and bash commands are shown in the example. If you use a different shell, use the equivalent commands.

4. At the command-line, type **sctop** to enter the **sctop** environment.



**NOTE:** For IDP8200, you also specify the IDP engine (0 through 5). For example, use **sctop 0** to enter the **sctop** environment for IDP engine 0 and **sctop 1** to enter the **sctop** environment for IDP engine 1.

5. Press alphabetic keyboard keys to display the desired report. You can press numeric keys to sort report data.

[Table 128 on page 488](#) describes the function of keyboard keys within the **sctop** environment.

**Table 128: Command Key Reference: sctop Utility**

Key	Function
a	Displays the ARP/MAC table.
b	Displays the table.
c	Displays the ICMP flow table.
d	Displays a strip chart, a text-based chart for packets per second, Kbps, and sessions.
e	Displays rulebase statistics.
f	Displays fragment chains.
g	Displays aggregate statistics.
h	Displays help for the <b>sctop</b> utility.
i	Displays the IP flow table. The IP flow table includes flows not accounted for in the ICMP, TCP, or UDP flow tables.
k	Displays attack statistics.
l	Displays qmodule statistics.
m	Displays system memory statistics.
o	Displays the flow table for flows that triggered the APE rulebase rate-limiting action.

Table 128: Command Key Reference: sctop Utility (*continued*)

Key	Function
p	Displays Spanning Tree Protocol (STP) information.
r	Displays the RPC program table.
s	Displays IDP Series device status.
t	Displays the TCP flow table.
u	Displays the UDP flow table.
v	Sorts in reverse order.
w	Displays HA status.
x	Displays the RPC XID table.
y	Displays IDS cache statistics.
z	Displays packet distribution.
0	Disables sorting.
1	Sorts by bytes per session.
2	Sorts by packets per session.
3	Sorts by expiration.
4	Sort by service.
5	Sorts by destination port.
6	Sorts by source address.
7	Sorts by destination address.



**TIP:** You can also display flow tables with the `scio var` command. With the `scio var` command, you can use the `-f` option to save the output of the table to a file.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Understanding sctop Flow Table Reports on page 490](#)

- [scio var on page 546](#)

## Understanding sctop Flow Table Reports

Table 129 on page 490 is a sample **sctop** flow table report.

**Table 129: sctop Flow Table Report**

Source-IP	Port	Destination-IP	Port	Flag	Direction	State	Service	Timeout
10.150.98.62	4137	10.150.20.43	139	R----	->>	Ltn	SMB	30/30
10.150.20.43	139	10.150.98.62	4137	R----	<<-	Close	-	30/30
10.150.73.39	6000	10.150.20.242	43117	R----	->>	Ltn	-	30/30

The Flag column includes 5 bits. [Table 130 on page 490](#) describes the **sctop** flow table flag column.

**Table 130: sctop Flow Table: Flag Column**

Position 1	Position 2	Position 3	Position 4	Position 5
Flow state. One of the following:	Management flow. One of the following:	Auxiliary flow. One of the following:	Packet logging. One of the following:	Flow sync. One of the following:
<ul style="list-style-type: none"> <li>• R (ready)</li> <li>• A (anticipated)</li> <li>• V (virtual)</li> <li>• X (rejected)</li> <li>• U (unknown)</li> </ul>	<ul style="list-style-type: none"> <li>• m (management flow)</li> <li>• – (not management flow)</li> </ul>	<ul style="list-style-type: none"> <li>• a (auxiliary flow)</li> <li>• - (not auxiliary flow)</li> </ul>	<ul style="list-style-type: none"> <li>• P (packet logging)</li> <li>• - (not packet logging)</li> </ul>	<ul style="list-style-type: none"> <li>• - (normal flow)</li> <li>• f (flow from failover)</li> <li>• s (flow synced from another IDP Series device)</li> </ul>

For example, the flag R– – – signifies ready, nonmanagement, nonauxiliary, no packet logging, normal; the flag A--ps signifies anticipated, nonmanagement, nonauxiliary, with packet logging, and synced over from another IDP Series device.

### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [sctop Task Summary on page 487](#)

# Using the scio Utility to Verify Feature Implementation

- [scio Monitoring Commands Task Summary on page 491](#)
- [Verifying the APE Rulebase on page 491](#)
- [Verifying Integration with an IC Series Unified Access Control Appliance on page 493](#)
- [Verifying MPLS Decapsulation on page 494](#)
- [Verifying the Flow Bypass Feature on page 495](#)

## scio Monitoring Commands Task Summary

---

You can use **scio** monitoring commands to verify implementation of IDP Series features. These commands display counters that increment when the feature is in use. In general, you do not need to use these commands to monitor normal operations.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying the APE Rulebase on page 491](#)
- [Verifying Integration with an IC Series Unified Access Control Appliance on page 493](#)
- [Verifying MPLS Decapsulation on page 494](#)
- [Verifying the Flow Bypass Feature on page 495](#)

## Verifying the APE Rulebase

---

**Purpose** When you are initially verifying APE rulebase functionality in your lab, you can use the **scio** utility to view APE-related process statistics. The counters should increase or decrement in accordance with your test load.

**Action** To view APE-related statistics in the CLI:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the commands described in the following table to verify APE rulebase functionality.

Table 131: APE-Related scio Commands

Command Syntax	Usage and Examples																																													
<b>scio ape-stats s0</b>	<p>Displays counts related to the APE rulebase rules where the action has been set to <b>Rate Limit</b>. For each applicable rule, the counter displays the rate limit, current utilization, and dropped packet count for both client-to-server (c2s), server-to-client (s2c) flows.</p> <pre>[root@default host admin]# scio ape-stats s0</pre> <table><tr><th>Rule</th><th>C2S(Mb)</th><th>S2C(MB)</th><th>C2S bytes</th><th>S2C bytes</th><th>C2S pkts</th><th>S2C pkts</th><th>C2S D-pkts</th><th>S2C D-pkts</th></tr><tr><td>1</td><td>100</td><td>10</td><td>73866</td><td>2622002</td><td>1234</td><td>1</td><td>75615</td><td>123</td></tr></table>	Rule	C2S(Mb)	S2C(MB)	C2S bytes	S2C bytes	C2S pkts	S2C pkts	C2S D-pkts	S2C D-pkts	1	100	10	73866	2622002	1234	1	75615	123																											
Rule	C2S(Mb)	S2C(MB)	C2S bytes	S2C bytes	C2S pkts	S2C pkts	C2S D-pkts	S2C D-pkts																																						
1	100	10	73866	2622002	1234	1	75615	123																																						
<b>scio var -s s0 sc_ape_flow_table</b>	<p>Displays the flow table for any current sessions where the rate-limit action is applied:</p> <pre>[root@default host admin]# scio var -s s0 sc_ape_flow_table</pre> <pre>sc_ape_flow_table:</pre> <table><tr><th>Source IP</th><th>Port</th><th>Destination IP</th><th>Port</th><th>FSt</th><th>Dir</th><th>Xtra info</th><th>VLAN</th><th>Timeout</th></tr><tr><th>Rule-index</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr><tr><td>1</td><td>[10.10.0.227</td><td>1050]</td><td>[67.99.176.30</td><td>80]</td><td>R</td><td>CTS</td><td>Estblshd</td><td>0</td></tr><tr><td>1</td><td>[67.99.176.30</td><td>80]</td><td>[10.10.0.227</td><td>1050]</td><td>R</td><td>STC</td><td>Estblshd</td><td>0</td></tr><tr><td>1</td><td>[10.157.5.2</td><td>1722]</td><td>[10.157.6.234</td><td>80]</td><td>R</td><td>CTS</td><td>Estblshd</td><td>0</td></tr></table> <p><b>TIP:</b> You can also use <b>sctop</b> to view the flow table for sessions where matching APE rate-limit rules. With <b>sctop</b>, use the <b>-o</b> option.</p> <p><b>NOTE:</b> Collection of APE statistics is disabled by default. Use the following command to turn on collection:</p> <pre>scio const -s s0 set sc_enable_ape_stats 1</pre>	Source IP	Port	Destination IP	Port	FSt	Dir	Xtra info	VLAN	Timeout	Rule-index									1	[10.10.0.227	1050]	[67.99.176.30	80]	R	CTS	Estblshd	0	1	[67.99.176.30	80]	[10.10.0.227	1050]	R	STC	Estblshd	0	1	[10.157.5.2	1722]	[10.157.6.234	80]	R	CTS	Estblshd	0
Source IP	Port	Destination IP	Port	FSt	Dir	Xtra info	VLAN	Timeout																																						
Rule-index																																														
1	[10.10.0.227	1050]	[67.99.176.30	80]	R	CTS	Estblshd	0																																						
1	[67.99.176.30	80]	[10.10.0.227	1050]	R	STC	Estblshd	0																																						
1	[10.157.5.2	1722]	[10.157.6.234	80]	R	CTS	Estblshd	0																																						

To view APE-related logs in NSM:

1. In the NSM navigation tree, select **Investigate > Log Viewer > Predefined**.
2. Click **Traffic** to display the predefined view of traffic logs, where APE logs are collected.
3. Use NSM sorting and filtering features to locate APE-related logs.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [scio var on page 546](#)
- [Using the sctop Utility \(CLI Procedure\) on page 487](#)
- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Configuring a Default Rate Limit on page 321](#)
- [Disabling the APE Rulebase on page 598](#)



## Verifying Integration with an IC Series Unified Access Control Appliance

**Purpose** The user role-based policy feature depends on integration with a compatible IC Series appliance. After you have configured the IC Series appliance to communicate with the IDP Series device, you can use the IDP OS command-line interface (CLI) to verify connectivity and verify receipt of the user session data used in user role-based policies.

If you encounter connectivity issues, you most likely need to troubleshoot from the IC Series side of the communication. From the IDP Series side, you need to ensure the IDP Series device can receive data from the IC Series appliance on port 7103 (that is, that your firewall does not block port 7103).

**Action** To verify integration with an IC Series appliance:

1. Log into the CLI as **admin** and enter **su** - to switch to **root**.
2. Enter the following command to verify connectivity:

```
[root@defaulthost admin]# scio user status
IDP-IC Connectivity is.....[Up]
User Session Table Lookup.....[Enabled]
```

3. Enter the following command to display the user session table:

```
[root@defaulthost admin]# scio user list
1. IP[      10.1.1.3] USER[test3] ROLES(1)[test-users3]
2. IP[      10.1.1.2] USER[test2] ROLES(1)[test-users2]
3. IP[      10.1.1.1] USER[test]  ROLES(1)[test-users]
```

```
=====
Total Matches Found (3)
=====
[root@defaulthost ~]#
```

4. Enter the following command to display a counter of changes made to the user session table:

```
[root@defaulthost admin]# scio user counters list all
+-----+-----+
      | SUCCESS | FAILURE |
+-----+-----+
Add   |      3  |      0  |
+-----+-----+
Delete |      0  |      0  |
+-----+-----+
Lookup |      0  |      0  |
+-----+-----+
```

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [scio user on page 544](#)
- IDP Series Configuration Requirements for Deployments with SA Series SSL VPN and IC Series Unified Access Control Appliances
- [Configuring Advanced Settings for the User-Role-Based Policy Feature on page 322](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- Understanding Communication Between IDP Series and IC Series Appliances

---

## Verifying MPLS Decapsulation

**Purpose** You can use the IDP OS command-line interface to verify the IDP engine processes Multiprotocol Label Switching (MPLS) frames.

**Action** To verify MPLS traffic has been processed:

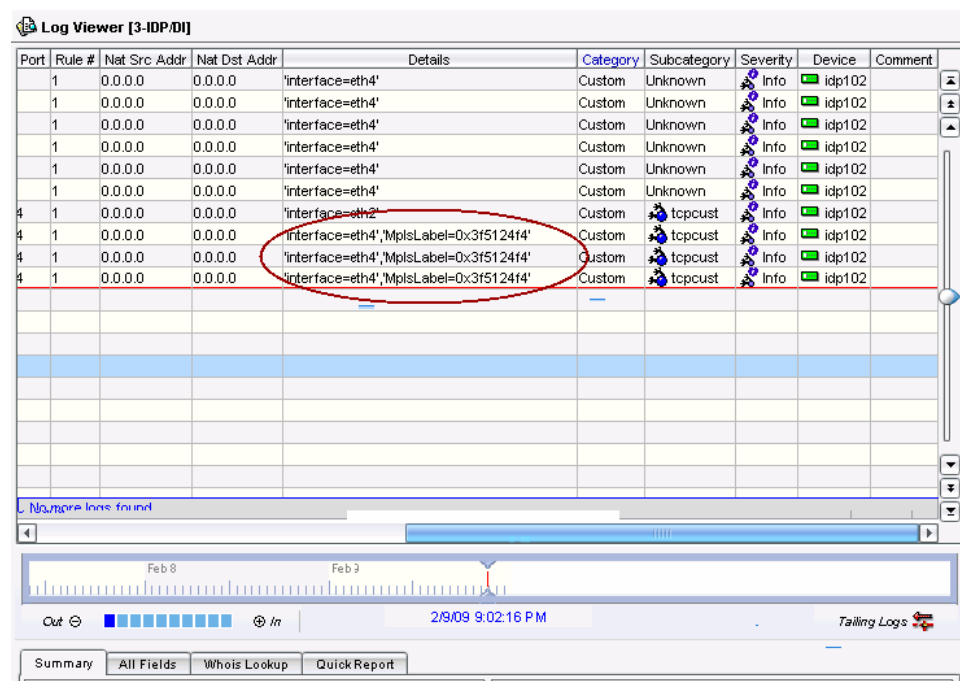
1. Log into the CLI as **admin** and enter **su** - to switch to **root**.
2. Enter the following command to display traffic counters:

```
[root@defaulthost admin]# scio counter get kpp
Name                               Value
sc_kpp_bad_ip_header               0
sc_kpp_ip_options                  0
sc_kpp_decapsulate                 0
sc_kpp_gre_decapsulate             0
sc_kpp_ppp_decapsulate             0
sc_kpp_gtp_decapsulate             0
sc_kpp_gtp_flow                   0
sc_kpp_tcpdecomp_uncompressed_ip  0
sc_kpp_tcpdecomp_compressed_ip    0
sc_kpp_deferred_send               0
sc_kpp_send_in_ip                  0
sc_kpp_ttl_error                   0
sc_kpp_routing_loop                0
sc_kpp_stp_drop                    0
sc_kpp_drop                        0
sc_kpp_drop_session                0
sc_kpp_no_route                    0
sc_kpp_flood_ip                    0
sc_kpp_push_eth_hdr_failed         0
sc_kpp_busy_inc                    0
sc_kpp_busy_dec                    0
sc_kpp_xmit_not_ready              0
sc_kpp_trylock_dev_failed          0
sc_kpp_hard_start_xmit_failed      0
sc_kpp_netif_queue_stopped         0
sc_kpp_vlan_pskb_expand_head       0
sc_kpp_packet_from_szp             0
sc_kpp_packet_freed                6
sc_kpp_packet_from_pool            6
sc_kpp_packet_copied               0
sc_kpp_mpls                        0
sc_kpp_clone_stopped               0
sc_kpp_clone                       0
sc_kpp_jpkt_free                   6
sc_kpp_tx_hold                     0
sc_kpp_fdrop                       0
```

The **sc\_kpp\_mpls** counter indicates the number of MPLS frames processed.

You can use Network and Security Manager (NSM) to filter and sort logs based on MPLS label. The MPLS label appears in the Details column, in the following format: **label =nnnnn,mmmm**. [Figure 159 on page 495](#) shows MPLS label information in logs.

**Figure 159: NSM Log Viewer: MPLS Label Information**



The screenshot shows the NSM Log Viewer interface. The table below represents the data shown in the log viewer. A red oval highlights the rows where the 'Details' column contains the MPLS label information: 'interface=eth4','MplsLabel=0x3f5124f4'.

Port	Rule #	Nat Src Addr	Nat Dst Addr	Details	Category	Subcategory	Severity	Device	Comment
1	1	0.0.0.0	0.0.0.0	'interface=eth4'	Custom	Unknown	Info	idp102	
1	1	0.0.0.0	0.0.0.0	'interface=eth4'	Custom	Unknown	Info	idp102	
1	1	0.0.0.0	0.0.0.0	'interface=eth4'	Custom	Unknown	Info	idp102	
1	1	0.0.0.0	0.0.0.0	'interface=eth4'	Custom	Unknown	Info	idp102	
1	1	0.0.0.0	0.0.0.0	'interface=eth4'	Custom	Unknown	Info	idp102	
1	1	0.0.0.0	0.0.0.0	'interface=eth4'	Custom	Unknown	Info	idp102	
4	1	0.0.0.0	0.0.0.0	'interface=eth2'	Custom	tcpcust	Info	idp102	
4	1	0.0.0.0	0.0.0.0	'interface=eth4','MplsLabel=0x3f5124f4'	Custom	tcpcust	Info	idp102	
4	1	0.0.0.0	0.0.0.0	'interface=eth4','MplsLabel=0x3f5124f4'	Custom	tcpcust	Info	idp102	
4	1	0.0.0.0	0.0.0.0	'interface=eth4','MplsLabel=0x3f5124f4'	Custom	tcpcust	Info	idp102	

To display the NSM Log Viewer:

1. Connect to NSM.
2. In the NSM navigation tree, select **Investigate > Log Viewer > IDP/DI**.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [scio counter on page 522](#)
- [Enabling Inspection of MPLS Traffic on page 318](#)
- [Disabling MPLS Decapsulation on page 601](#)

## Verifying the Flow Bypass Feature

**Purpose** You can use the command-line interface (CLI) to verify successful implementation of the flow bypass feature.

**Action** To verify successful implementation of the flow bypass feature:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to verify the feature is enabled:

```
[root@defaultthost admin]# scio subs status s0
Status for subs 's0'
up since - Mon Apr 27 13:11:43 2009
Packets/second: 6873          peak: 251741 @ Mon Apr 27 16:17:31 2009
KBits/second:   2953          peak: 108170 @ Mon Apr 27 16:17:31 2009
Packets received: icmp 1121, tcp 2823211, udp 1855862, other 0
Current flows: icmp 0, tcp 0, udp 98905, other 0
Current sessions: icmp 0, tcp 0, udp 40481, other 0
Current bypassed flows : 45076
Current bypass mode : ON
Latency Statistics (time in micro seconds):
Min: 0 Max: 0 Ave: 0
Performance statistics
Average packet lifetime:
Cycles: 0 Instructions: 0 CPI: 0.00 Cache misses: 0 hits: 0
Current policy: idpengine v0
```

The **scio subs status** command returns details on whether flow bypass is enabled and the number of flows currently marked for flow bypass (if any).

3. Enter the following command to display the current system packet queue size and IDP engine packet queue size:

```
[root@defaultthost admin]# scio var -s s0 sc_bypass_counts
sc_bypass_counts:
| System PktRxQueue Count / EnginePktRxQueue Count |
|-----+-----|
| 103441 / 103441 |
| 103415 / 103415 |
| 103411 / 103411 |
| 103372 / 103372 |
| 103466 / 103466 |
```

This is not a count of flows that have been bypassed, rather the running packet count used in the calculation to trigger flow bypass. When the flow bypass feature is enabled and functioning properly, the counts will increment and decrement. The first number is the number of packets in the queue for all IDP engines (the system count). The second number is the number of packets in the queue for the particular IDP engine.

4. Enter the following command to verify the system packet size rising threshold:

```
[root@defaultthost admin]# scio const -s s0:flow get sc_flow_bypass_threshold_hi
scio: sc_flow_bypass_threshold_hi = 0x5a
[root@defaultthost admin]#
```

The command returns the rising packet size threshold (percent).

5. Enter the following command to verify the system packet size reset threshold:

```
[root@defaultthost admin]# scio const -s s0:flow get sc_flow_bypass_threshold_low
scio: sc_flow_bypass_threshold_low = 0x50
[root@defaultthost admin]#
```

The command returns the reset packet size threshold (percent).

6. Enter the following command to display statistics for any current flows marked for flow bypass:

```
[root@defaultthost admin]# scio var -s s0 sc_bypass_flow_table
sc_bypass_flow_table:
| Source IP | Port | Destination IP | Port | FSt | Dir | Xtra info | VLAN |
Timeout |
```

```
|-----+-----+-----+-----+-----+-----+-----+-----|
[8.0.0.51      14253] [8.0.0.201      24253] B   CTS   -   0
39/60
[8.0.0.201     24253] [8.0.0.51      14253] B   STC   -   0
39/60
```

The command returns details of current flows marked for flow bypass.

7. Enter the following command to display counters related to flow bypass:

```
[root@defaulthost admin]# scio counter get flow
```

Name	Value
sc_flow_fast_path	2526998
sc_flow_slow_path	196631
sc_flow_icmp_error	0
sc_flow_session_failed	0
sc_flow_session_deleted	0
sc_flow_session_ageout	0
sc_flow_ageout_in_use	0
sc_flow_ageout_in_fpga	0
sc_flow_delete_wrong_cookie	0
sc_flow_delete_null_session	0
sc_flow_packet_log	0
sc_flow_busy_packet	0
sc_flow_out_of_order	0
sc_flow_device_fifo_size	0
sc_flow_device_fifo_overflow	0
sc_flow_policy_cache_hit	0
sc_flow_policy_cache_miss	11519
sc_flow_hash_collision_max	8192
sc_flow_hash_collision	4095
sc_flow_ha_flip	0
sc_flow_bad_udp_csum	0
sc_flow_gate_add	0
sc_flow_gate_found	0
sc_flow_cookie_unmatched	0
sc_flow_tag_cookie_unmatched	0
sc_flow_sm_cookie_unmatched	0
sc_flow_idp_cookie_unmatched	0
sc_flow_tag_cookie_unmatched_no	0
sc_flow_go_away	0
sc_flow_wrong_sm_index	0
sc_flow_periodic_stat_update	15764
sc_flow_stack_max_usage	0
sc_avt_update_drop_sess	1
sc_avt_update_drop_nobuf	862
sc_avt_update_flow_init	0
sc_avt_update_flow_fini	7264
sc_avt_update_flow_stat	14901
sc_avt_buf_size	478
sc_flow_bypass_flows	185103
sc_flow_bypass_mode_on	3
sc_flow_bypass_mode_off	2

The command returns counts related to flow statistics. [Table 132 on page 497](#) describes the counters related to flow bypass.

**Table 132: scio counters Related to Flow Bypass**

Counter	Description
sc_flow_bypass_flows	Total number of flows marked for flow bypass.

Table 132: scio counters Related to Flow Bypass (*continued*)

Counter	Description
sc_flow_bypass_mode_on	Number of times flow bypass mode was triggered.
sc_flow_bypass_mode_off	Number of times flow bypass mode was reset.

**Related  
Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Enabling the Flow Bypass Feature on page 319](#)
- [scio const on page 505](#)
- [scio counter on page 522](#)
- [scio subs on page 535](#)
- [scio var on page 546](#)

## CHAPTER 46

# scio Commands

## scio agentconfig

**Syntax** `scio agentconfig options arguments`

**Description** Displays or sets values for the Network and Security Manager (NSM) agent configuration. The NSM agent is installed on the IDP Series device and used to communicate with the NSM device server.

The following example shows how to use **scio agentconfig** options to display and then set values for the NSM agent configuration:

```
[root@defaulthost admin]# scio agentconfig list
Primary management IP: 10.158.111.2
Primary management port: 7803
No secondary management IP defined
Secondary management port: 7803
Device id: device_id
One time password is erased

[root@defaulthost admin]# scio agentconfig server1-ip 10.158.131.2
[root@defaulthost admin]#

[root@defaulthost admin]# scio agentconfig server1-ip list
Primary management IP address: 10.158.131.2
```

**Options** [Table 133 on page 500](#) describes command options and arguments.

**Table 133: Command Reference: scio agentconfig**

Options	Description
list	<b>list</b> —Lists the current settings for all NSM agent parameters.
server1-ip {list   <i>IPaddress</i> }	<b>list</b> —Lists the current setting for the NSM agent option. <i>IPaddress</i> —Specifies the IP address for the primary NSM server.
server1-port {list   <i>port</i> }	<b>list</b> —Lists the current setting for the NSM agent option. <i>port</i> —Specifies the port number for the primary NSM server.
server2-ip {list   <i>IPaddress</i> }	<b>list</b> —Lists the current setting for the NSM agent option. <i>IPaddress</i> —Specifies the IP address for the secondary NSM server.
server2-port {list   <i>port</i> }	<b>list</b> —Lists the current setting for the NSM agent option. <i>port</i> —Specifies the port number for the primary NSM server.
otp {list   <i>password</i> }	<b>list</b> —Lists the current on-time password used by NSM to connect to the IDP Series device. <i>password</i> —Specifies a password to be used by NSM to make a connection to the IDP Series device.



Table 133: Command Reference: `scio agentconfig` (*continued*)

Options	Description
deviceid {list   id}	<b>list</b> —Lists the current setting for the NSM agent option.
	<b>id</b> —Specifies a device ID string. The ID must be 42 hexadecimal characters.

## scio app cache

**Syntax** `scio app cache argument`

**Description** Lists applications identified by the application identification feature. To optimize performance, the application identification feature maintains a cache of application definitions for applications it identifies. When processing traffic, the application identification feature compares traffic against the cached application definitions. If it does not identify any matches, it uses heuristic methods to identify the application and saves the resulting definition to the cache.

When verifying or troubleshooting features, you might find it useful to track changes to the application identification cache.

**Options** [Table 134 on page 502](#) describes arguments to the **scio app cache** command and provides examples of command syntax.

**Table 134: Command Reference: scio app cache**

Arguments	Usage and Example
list	<p>Lists the last 32 applications identified.</p> <pre>[root@defaulthost admin]# scio app cache list Application system cache: total 66 show 32 Index VLAN IP Port Proto Application 0 0 21.0.0.101 22 6 SSH 1 0 23.0.0.101 22 6 SSH 2 0 29.0.0.101 22 6 SSH 3 0 25.0.0.101 22 6 SSH 4 0 27.0.0.101 22 6 SSH 5 0 23.0.0.101 80 6 HTTP 6 0 29.0.0.101 21 6 FTP 7 0 21.0.0.101 80 6 HTTP 8 0 25.0.0.101 21 6 FTP 9 0 27.0.0.101 21 6 FTP ... 29 0 25.0.0.1 80 6 HTTP 30 0 20.0.0.101 22 6 SSH 31 0 22.0.0.101 22 6 SSH</pre>

Table 134: Command Reference: scio app cache (*continued*)

Arguments	Usage and Example
listall	<p>Lists all applications identified since the last time the cache was cleared.</p> <pre>[root@defaulthost admin]# scio app cache listall Application system cache: total 66 show 66 Index VLAN IP Port Proto Application 0 0 21.0.0.101 22 6 SSH 1 0 23.0.0.101 22 6 SSH 2 0 29.0.0.101 22 6 SSH 3 0 25.0.0.101 22 6 SSH 4 0 27.0.0.101 22 6 SSH 5 0 23.0.0.101 80 6 HTTP 6 0 29.0.0.101 21 6 FTP 7 0 21.0.0.101 80 6 HTTP 8 0 25.0.0.101 21 6 FTP 9 0 27.0.0.101 21 6 FTP ... 62 0 22.0.0.1 21 6 FTP 63 0 28.0.0.1 80 6 HTTP 64 0 26.0.0.1 80 6 HTTP 65 0 24.0.0.1 80 6 HTTP</pre>
clear	<p>Clears the cache.</p> <pre>[root@defaulthost admin]# scio app cache clear</pre>

## scio app sig list

**Syntax** `scio app sig list`

**Description** Lists details of the application signatures that are relevant to the current policy. The application signatures are relevant if the application is specified or implicated by IDP rulebase or APE rulebase rules.

When verifying or troubleshooting features with the CLI, you might find this command useful for verifying expected behavior.

**Options** The list option is the only argument and is required.

```
[root@defaulthost ~]# scio app sig list
Application signatures: total 165 show 165
APPLICATIONID:ARES, index 0, service 77, mindata 7, order 61, tcp 0-65535, no udp
APPLICATIONID:ICCP, index 1, service 106, mindata 2, order 147, tcp 102-102, no
udp
APPLICATIONID:HALF-LIFE, index 2, service 113, mindata 4, order 127, no tcp, udp
1024-65535
APPLICATIONID:ICQ, index 3, service 172, mindata 10, order 22, tcp 0-65535, no
udp
APPLICATIONID:PPTP, index 4, service 128, mindata 11, order 173, tcp 1723-1723,
no udp
APPLICATIONID:X11, index 5, service 144, mindata 6, order 85, tcp 0-65535, no udp
APPLICATIONID:GNUCLEUSLAN-CONNECT, index 6, service 151, mindata 16, order 40,
tcp 0-65535, no udp
APPLICATIONID:VMWARE-WEBUI, index 7, service 96, mindata 180, order 183, tcp
8333-8333, no udp
APPLICATIONID:FREecast, index 8, service 81, mindata 50, order 142, no tcp, udp
0-65535
APPLICATIONID:MSRPC, index 9, service 55, mindata 20, order 42, tcp 135-135 137-139
445-445 1024-65535, udp 135-135 137-139 445-445 1024-65535
APPLICATIONID:MSN, index 10, service 41, mindata 20, order 107, tcp 0-65535, no
udp
APPLICATIONID:DRDA, index 11, service 121, mindata 20, order 52, tcp 0-65535, no
udp
APPLICATIONID:GNUTELLA-URN-DOWNLOAD, index 12, service 131, mindata 26, order
133, tcp 0-65535, no udp
APPLICATIONID:GNUTELLA-FIREWALLED, index 13, service 86, mindata 70, order 34,
tcp 0-65535, no udp
APPLICATIONID:IRC, index 14, service 32, mindata 32, order 46, tcp 0-65535, no
udp
APPLICATIONID:QQ, index 15, service 125, mindata 3, order 105, tcp 80-80 443-443,
udp 0-65535
APPLICATIONID:LOTUSNOTES, index 16, service 135, mindata 21, order 134, tcp
0-65535, no udp
```

## scio const

**Syntax** `scio const {list | -c name | -d | -p service | -s sO:qmodule | -v name} {list | get constant | set constant value}`

**Description** Displays or sets values for IDP OS kernel constants. Kernel constants determine whether features are enabled or disabled, as well as feature configuration parameters.

Changes you make to kernel constants from the CLI do not persist across restarts. To make your change persistent:

1. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as `vi`.
2. Add the constant below the line `user_start_end()`. For example:

```
user_start_end()
{
    $SCIO const -s sO set sc_ssl_sessid_timeout 90
}
```

3. Save the file.

4. Restart the IDP engine:

```
[root@default host admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

**Options** [Table 135 on page 505](#) describes the basic parameters of `scio const` commands.

**Table 135: Command Reference: scio const**

Options and Arguments	Usage and Examples
list	<p>When specified with no other options or arguments, the <code>scio const list</code> command lists constants related to memory, logging, storage, and debugging.</p> <pre>[root@default host admin]# scio const list sc_debug_features           = 0x10      [ 0...ffffffff ] sc_debug_qmodules           = 0x0       [ 0...ffffffff ] sc_debug_services           = 0x0       [ 0...ffffffff ] sc_debug_services2          = 0x0       [ 0...ffffffff ] sc_debug_level              = 0x1       [ 0...3 ] sc_debug_detail              = 0x0       [ 0...1 ] sc_panic_on_assert          = 0x0       [ 0...1 ] sc_malloc_debug             = 0x0       [ 0...1 ] sc_malloc_debug_size        = 0x200     [ 0...f4240 ] sc_malloc_fail_report_freq  = 0xc350  [ 0...ffffffff ] sc_log_cache_size           = 0x3200   [ 1...ffff ] sc_log_chunk_size           = 0x4000   [ 400...4000 ] sc_log_chunk_timeout        = 0x186a0  [ 1...f4240 ] sc_pktlog_cache_size        = 0x100000 [ 400...ffffffff ] sc_pktlog_chunk_size        = 0x1f82e  [ 400...ffffffff ] sc_pktlog_chunk_timeout     = 0x186a0  [ 1...f4240 ] sc_pktlog_capture_timeout   = 0x5      [ 1...708 ] [...]</pre>

Table 135: Command Reference: scio const (*continued*)

Options and Arguments

Usage and Examples

-d

Specify the **-d** option for commands related to protocol decoders.

Specify the **list** option to display a list of which protocol decoders are enabled or disabled:

[root@defaulthost admin]# scio const -d list

Protocol Decoders Enabled are:

AIM	APE	BGP	BWMON	CHARGEN	DHCP
DISCARD	DNS	ECHO	FINGER	FTP	GNUTELLA
GOPHER	H225RAS	H225SGN	ICMP	IDENT	IEC104
IKE	IRC	LDAP	LPR	MGCP	MSN
MSRPC	MSSQL	MYSQL	NBDS	NBNAME	NFS
NNTP	NTP	POP3	PORTMAPPER	PROFILER	PTYPE
REXEC	RLOGIN	RPC	RSH	RTSP	RUSERS
SIP	SMB	SNMPTRAP	SQLMON	SSH	SSL
SYSLOG	TELNET	TNS	VNC	WHOIS	YMSG

Protocol Decoders Disabled are:

HTTP	IMAP	RADIUS	SMTP	SNMP	TFTP
------	------	--------	------	------	------

Specify the **get decoder** option to display whether the specified decoder is enabled or disabled. (1 = enabled; 0 = disabled). For example, the following command displays the value for the SIP decoder. 1 indicates the SIP decoder is enabled.

[root@defaulthost admin]# scio const -d get SIP

scio: SIP = 0x1

Specify the **set decoder value** option to change the enabled/disabled setting. The following example turns off the SIP decoder.

[root@defaulthost admin]# scio const -d set SIP 0

scio: setting SIP to 0x0

[root@defaulthost admin]#

-v *name*

Specify the **-v** option for commands related to virtual routers.

[root@defaulthost admin]# scio const -v vr1 list

sc_arp_timeout	= 0xe10	[ 1...ffffffff ]
sc_arp_proxy_timeout	= 0x14	[ 1...ffffffff ]
sc_arp_logging	= 0x1	[ 0...1 ]
sc_arp_spoof_detect	= 0x1	[ 0...1 ]
sc_mac_timeout	= 0xe10	[ 1...ffffffff ]
sc_mac_unknown_timeout	= 0x14	[ 1...ffffffff ]
sc_stp_enabled	= 0x0	[ 0...1 ]
sc_stp_bridge_priority	= 0x8000	[ 0...ffff ]
sc_stp_bridge_max_age	= 0x14	[ 6...28 ]
sc_stp_bridge_hello_time	= 0x2	[ 1...a ]
sc_stp_bridge_forward_delay	= 0xf	[ 4...1e ]
sc_stp_check_interval_ticks	= 0xa	[ 1...3e8 ]
sc_stp_logging	= 0x1	[ 0...1 ]
sc_arp_request_record	= 0x1	[ 0...1 ]
sc_arp_spoof_pass_thru	= 0x1	[ 0...1 ]

Table 135: Command Reference: scio const (*continued*)

Options and Arguments	Usage and Examples
-s <i>s0:qmodule</i>	<p>Specify the -s option for commands related to subscriber settings.</p> <p><b>s0</b> specifies subscriber s0, the only valid argument for <b>scio const -s</b>.</p> <p>In some cases, <b>scio const</b> syntax requires you specify the subscriber qmodule. The example commands in this reference use the construction <b>s0:qmodule</b> to include the subscriber qmodule when it is required. The example commands do not include the subscriber qmodule when it is not required.</p> <pre>[root@default host admin]# scio const -s s0 list sc_rpc_xid_timeout           = 0x5          [ 1...3c ] sc_rpc_program_timeout       = 0x12c         [ 1...12c ] sc_exempt_mgt_traffic        = 0x1          [ 0...1 ] sc_enable_statistics         = 0x0          [ 0...1 ] sc_bypass_dfa                = 0x0          [ 0...1 ] sc_enable_packet_count       = 0x1          [ 0...1 ] sc_enable_rule_stats         = 0x0          [ 0...1 ] sc_ip_fragment_timeout       = 0x5          [ 1...3c ] sc_ip_fragment_min_size      = 0x0          [ 0...ffff ] sc_ip_fragment_max_ppf       = 0xffff       [ 8...ffff ]  [...]</pre>
-c <i>name</i>	<p>Specify the -c option for commands related to virtual circuits.</p> <pre>[root@default host admin]# scio const -c eth2 list sc_stp_port_enabled          = 0x1          [ 0...1 ] sc_stp_change_detection_enabled = 0x1        [ 0...1 ] sc_stp_port_priority         = 0x80         [ 0...ff ] sc_stp_port_path_cost        = 0x64         [ 1...ffff ] sc_xmit_queue_size           = 0x400        [ 0...4000 ]</pre>
-p <i>service</i>	<p>Specify the -p option for commands related to service settings.</p> <pre>[root@default host admin]# scio const -p http list sc_http_request_length       = 0x2000        [ 1...2000 ] sc_http_header_length        = 0x2000        [ 1...2000 ] sc_http_cookie_length        = 0x2000        [ 1...2000 ] sc_http_auth_length          = 0x200         [ 1...400 ] sc_http_content_type_length  = 0x200         [ 1...2000 ] sc_http_user_agent_length    = 0x100         [ 1...2000 ] sc_http_soapaction_length    = 0x400         [ 1...2000 ] sc_http_host_length          = 0x40          [ 1...2000 ] sc_http_referer_length       = 0x2000        [ 1...2000 ] sc_http_alternate_ports      = 0x1          [ 0...1 ] sc_http_failed_logins        = 0x4          [ 2...64 ] sc_http_brute_search         = 0x10         [ 2...64 ] sc_http_ignore               = 0x0          [ 0...4 ] sc_http_jpeg_depth           = 0x1000        [ 0...1000 ] sc_http_min_html_tag_len     = 0xa          [ 0...2000 ] sc_http_enable_parse_html    = 0x1          [ 0...1 ] sc_http_enable_parse_html_tags = 0x1        [ 0...1 ] sc_http_enable_chunk_contexts = 0x1          [ 0...1 ] sc_http_chunk_min_len        = 0xa          [ 0...32 ]</pre>

Table 135: Command Reference: `scio const` (*continued*)

Options and Arguments	Usage and Examples
<code>list</code>	<p>When specified in syntax after the <code>-c</code>, <code>-p</code>, <code>-s</code>, or <code>-v</code> options, lists all constants related to the class specified by the flag.</p> <pre>[root@defaulthost admin]# scio const -s s0 list sc_rpc_xid_timeout           = 0x5          [ 1...3c ] sc_rpc_program_timeout      = 0x12c         [ 1...12c ] sc_exempt_mgt_traffic       = 0x1          [ 0...1 ] sc_enable_statistics        = 0x0          [ 0...1 ] sc_bypass_dfa               = 0x0          [ 0...1 ] sc_enable_packet_count      = 0x1          [ 0...1 ] sc_enable_rule_stats        = 0x0          [ 0...1 ] sc_ip_fragment_timeout      = 0x5          [ 1...3c ] sc_ip_fragment_min_size     = 0x0          [ 0...ffff ] sc_ip_fragment_max_ppf      = 0xffff       [ 8...ffff ]  [...]</pre>
<code>get constant</code>	<p>Gets values for the specified kernel constant.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_gre_decapsulation scio: sc_gre_decapsulation = 0x0</pre>



Table 135: Command Reference: `scio const` (*continued*)

Options and Arguments	Usage and Examples
<code>set constant value</code>	<p>Sets values for the specified kernel constant.</p> <pre>[root@defaulthost admin]# scio const -s s0 set sc_gre_decapsulation 1</pre> <p>scio: setting <code>sc_gre_decapsulation</code> to 0x1</p> <p>For information on particular constants, refer to the following tables:</p> <ul style="list-style-type: none"> <li>• <a href="#">Table 136 on page 509</a> provides usage and examples of kernel constants related to the application identification feature.</li> <li>• <a href="#">Table 137 on page 511</a> provides usage and examples of kernel constants related to the application policy enforcement (APE) rulebase.</li> <li>• <a href="#">Table 138 on page 512</a> provides usage and examples of kernel constants related to the application volume tracking (AVT) feature.</li> <li>• <a href="#">Table 139 on page 513</a> provides usage and examples of kernel constants related to the flow bypass feature.</li> <li>• <a href="#">Table 140 on page 514</a> provides usage and examples of kernel constants related to flow behavior during policy load.</li> <li>• <a href="#">Table 141 on page 515</a> provides usage and examples of kernel constants related to GRE decapsulation.</li> <li>• <a href="#">Table 142 on page 515</a> provides usage and examples of kernel constants related to GTP decapsulation.</li> <li>• <a href="#">Table 143 on page 517</a> provides usage and examples of kernel constants related to IPsec ESP NULL decapsulation.</li> <li>• <a href="#">Table 144 on page 517</a> provides usage and examples of kernel constants related to MPLS decapsulation.</li> <li>• <a href="#">Table 145 on page 518</a> provides usage and examples of kernel constants related to SSL inspection.</li> <li>• <a href="#">Table 146 on page 519</a> provides usage and examples of the kernel constant that determines the maximum frame size processed by the IDP engine.</li> <li>• <a href="#">Table 147 on page 519</a> provides usage and examples of kernel constants related to the SYN Protector rulebase.</li> <li>• <a href="#">Table 148 on page 521</a> provides usage and examples of kernel constants related to the user role-based policy feature.</li> </ul>

[Table 136 on page 509](#) provides usage and examples of kernel constants related to the application identification feature.

Table 136: `scio const` Arguments Related to the Application Identification Feature

Constants and Values	Usage and Examples
<code>sc_ai_enable</code>	<p>Gets or sets the constant that determines whether the application identification feature is enabled or disabled.</p> <p>The default is 1 (on). 0 turns application identification off.</p> <pre>[root@defaulthost admin]# scio const get sc_ai_enable</pre> <p>scio: <code>sc_ai_enable</code> = 0x1</p> <pre>[root@defaulthost admin]# scio const set sc_ai_enable 0</pre> <p>scio: setting <code>sc_ai_enable</code> to 0x0</p> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>

Table 136: scio const Arguments Related to the Application Identification Feature (*continued*)

Constants and Values	Usage and Examples
sc_ai_check_first_session	<p>Gets or sets the constant that determines whether the application identification feature attempts to identify the application from the first session.</p> <p>The default is 1 (on). 0 turns the setting off.</p> <pre>[root@defaulthost admin]# scio const get sc_ai_check_first_session scio: sc_ai_check_first_session = 0x1  [root@defaulthost admin]# scio const set sc_ai_check_first_session 0 scio: setting sc_ai_check_first_session to 0x0</pre>
sc_ai_max_tcp_sess_pkt_mem	<p>Gets or sets the constant that determines the maximum bytes of memory used to perform application identification on TCP sessions.</p> <p>The default is 30,000 (0x7530).</p> <p>Possible values: 0 to 60,000.</p> <pre>[root@defaulthost admin]# scio const get sc_ai_max_tcp_sess_pkt_mem scio: sc_ai_max_tcp_sess_pkt_mem = 0x7530  [root@defaulthost admin]# scio const set sc_ai_max_tcp_sess_pkt_mem 60000 scio: setting sc_ai_max_tcp_sess_pkt_mem to 0xEA60</pre>
sc_ai_max_udp_sess_pkt_mem	<p>Gets or sets the constant that determines the maximum bytes of memory used to perform application identification on UDP sessions.</p> <p>The default is 10,000 (0x2710).</p> <p>Possible values: 0 to 20,000 (0x4e20).</p> <pre>[root@defaulthost admin]# scio const get sc_ai_max_udp_sess_pkt_mem scio: sc_ai_max_udp_sess_pkt_mem = 0x7530  [root@defaulthost admin]# scio const set sc_ai_max_udp_sess_pkt_mem 20000 scio: setting sc_ai_max_udp_sess_pkt_mem to 0x4e20</pre>
sc_ai_num_sess	<p>Gets or sets the constant that determines whether the maximum number of concurrent sessions where application identification can be used.</p> <p>The default is 50,000 (0xc350).</p> <p>Possible values: 0 to 200,000 (0x30d40).</p> <pre>[root@defaulthost admin]# scio const get sc_ai_num_sess scio: sc_ai_num_sess = 0xc350  [root@defaulthost admin]# scio const set sc_ai_num_sess 200000 scio: setting sc_ai_num_sess to 0x30d40</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>

Table 136: scio const Arguments Related to the Application Identification Feature (*continued*)

Constants and Values	Usage and Examples
sc_ai_max_pkt_mem	<p>Gets or sets the constant that determines the maximum bytes of memory used to store packets processed by the application identification feature.</p> <p>The default is 50,000,000 (0x2faf080).</p> <p>Possible values: 0 to 200,000,000 (bebc200).</p> <pre>[root@defaulthost admin]# scio const get sc_ai_max_pkt_mem scio: sc_ai_max_pkt_mem = 0x0x2faf080  [root@defaulthost admin]# scio const set sc_ai_max_pkt_mem 200000000 scio: setting sc_ai_max_pkt_mem to 0xbebc200</pre>
sc_ai_check_bytes	<p>Gets or sets the constant that determines the length of the check byte.</p> <p>The default is 10 (0xa).</p> <p>Possible values: 0 to 2000 (0x7d0).</p> <pre>[root@defaulthost admin]# scio const get sc_ai_check_bytes scio: sc_ai_check_bytes = 0xa  [root@defaulthost admin]# scio const set sc_ai_check_bytes 20 scio: setting sc_ai_check_bytes to 0x14</pre>

[Table 137 on page 511](#) provides usage and examples of kernel constants related to the application policy enforcement (APE) rulebase.

Table 137: scio const Arguments Related to the APE Rulebase

Constants and Values	Usage and Examples
sc_ape_enable	<p>Gets or sets the constant that determines whether the application policy enforcement rulebase is enabled or disabled.</p> <p>The default is 1 (on). 0 turns the APE rulebase off.</p> <pre>[root@defaulthost admin]# scio const get sc_ape_enable scio: sc_ape_enable = 0x1  [root@defaulthost admin]# scio const set sc_ape_enable 0 scio: setting sc_ape_enable to 0x0</pre>
sc_enable_ape_stats	<p>Gets or sets the constant for APE statistics collection.</p> <p>The default is 0 (off). 1 turns statistics collection on.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_enable_ape_stats scio: sc_enable_ape_stats = 0x0  [root@defaulthost admin]# scio const -s s0 set sc_enable_ape_stats 1 scio: setting sc_enable_ape_stats to 0x1</pre>

Table 137: scio const Arguments Related to the APE Rulebase (*continued*)

Constants and Values	Usage and Examples
sc_enable_ape_stats	<p>Gets or sets the constant for APE statistics collection.</p> <p>The default is 0 (off). 1 turns statistics collection on.</p> <pre>[root@default host admin]# scio const -s s0 get sc_enable_ape_stats scio: sc_enable_ape_stats = 0x0</pre> <pre>[root@default host admin]# scio const -s s0 set sc_enable_ape_stats 1 scio: setting sc_enable_ape_stats to 0x1</pre>
sc_ape_default_rate_limit	<p>Gets or sets the constant that determines the default rate limit for sessions that do not match APE rules.</p> <p><b>NOTE:</b> If you have enabled per user rate limiting (also called per subscriber rate limiting), the default rate limit is applied per user. If not, the default rate limit is a maximum allocation for all sessions that do not match APE rules.</p> <pre>[root@default host admin]# scio const get sc_ape_default_rate_limit scio: sc_ape_default_rate_limit = 0xffffffff</pre> <p>The default is 4,294,967,295 bps (0xffffffff in hexadecimal; 4,096 Mbps or .5 Gbps), which effectively turns off the “default rate limit”.</p> <p>The following example sets a limit of .25 Gbps:</p> <pre>[root@default host admin]# scio const set sc_ape_default_rate_limit 2147483648 scio: setting sc_ape_default_rate_limit to 0x80000000</pre>
sc_per_subscriber_ratelimit	<p>Gets or sets the constant that determines whether rate limits are enforced per user role or per user. The default is 0 (rate limit applied when aggregate bandwidth for the user role reaches the threshold). Change to 1 if you want the rate limit applied when bandwidth utilization for any user reaches the threshold.</p> <pre>[root@default host admin]# scio const -s s0 get sc_per_subscriber_ratelimit scio: sc_per_subscriber_ratelimit = 0x0</pre> <pre>[root@default host admin]# scio const -s s0 set sc_per_subscriber_ratelimit 1 scio: setting sc_per_subscriber_ratelimit to 0x1</pre>

[Table 138 on page 512](#) provides usage and examples of kernel constants related to the application volume tracking (AVT) feature.

Table 138: scio const Arguments Related to the Application Volume Tracking Feature

Constants and Values	Usage and Examples
sc_periodic_stat_update	<p>Gets or sets the constant that determines whether the application volume tracking feature is enabled or disabled.</p> <p>The default is 1 (on). 0 turns AVT off.</p> <pre>[root@default host admin]# scio const -s s0:flow get sc_periodic_stat_update scio: sc_periodic_stat_update = 0x1</pre> <pre>[root@default host admin]# scio const -s s0:flow set sc_periodic_stat_update 0 scio: setting sc_periodic_stat_update to 0x01</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>

[Table 139 on page 513](#) provides usage and examples of kernel constants related to the flow bypass feature.

**Table 139: scio const Arguments Related to Flow Bypass**

Constants and Values	Usage and Examples
sc_flow_bypass_enable	<p>Gets or sets the constant that determines whether the flow bypass feature is enabled or disabled.</p> <p>The default is 0 (off). 1 turns the flow bypass feature on.</p> <pre>[root@defaultthost admin]# scio const -s s0:flow get sc_flow_bypass_enable scio: sc_flow_bypass_enable = 0x0</pre> <pre>[root@defaultthost admin]# scio const -s s0:flow set sc_flow_bypass_enable 1 scio: setting sc_flow_bypass_enable to 0x1</pre>
sc_flow_bypass_threshold_hi	<p>Gets or sets the constant that determines the system packet queue size rising threshold.</p> <p>The default is 90 (percent).</p> <p>Possible values 0-100.</p> <pre>[root@defaultthost admin]# scio const -s s0:flow get sc_flow_bypass_threshold_hi scio: sc_flow_bypass_threshold_hi = 0x5a</pre> <pre>[root@defaultthost admin]# scio const -s s0:flow set sc_flow_bypass_threshold_hi 95 scio: setting sc_flow_bypass_threshold_hi to 0x5f</pre>
sc_flow_bypass_threshold_low	<p>Gets or sets the constant that determines the system packet queue size reset threshold.</p> <p>The default is 80 (percent).</p> <p>Possible values 0-100.</p> <pre>[root@defaultthost admin]# scio const -s s0:flow get sc_flow_bypass_threshold_low scio: sc_flow_bypass_threshold_low = 0x50</pre> <pre>[root@defaultthost admin]# scio const -s s0:flow set sc_flow_bypass_threshold_low 85 scio: setting sc_flow_bypass_threshold_low to 0x55</pre>

[Table 140 on page 514](#) provides usage and examples of kernel constants related to flow behavior during policy load.

Table 140: scio const Arguments Related to Policy Load

Constants and Values	Usage and Examples
sc_flow_reset_on_policy	<p>Gets or sets the constant that determines whether the flow table is reset when a new policy is loaded. When the flow table is reset, existing sessions are passed through uninspected.</p> <p>Valid values are 0 (do not reset on policy load) or 1 (reset on policy load).</p> <p>For IDP75 and IDP200, the default is 1, and you cannot override the default.</p> <p>For high-end appliances, the default is 0. When you load a new policy, the IDP system flow table will maintain sessions belonging to the previously installed policy as well as the newly installed policy. The IDP engine will continue to use the previously installed security policy to inspect previous sessions; and use the newly installed security policy to inspect new sessions. When the previously installed policy is no longer in use, it is unloaded and all traffic is inspected using the newly installed policy. For IDP8200 and IDP250, the IDP system can maintain flows for as many as two security policies. For IDP1100, IDP800, and IDP600, the IDP system can maintain flows for as many as four security policies.</p> <p>The default is 0 (off). 1 turns the flow bypass feature on.</p> <pre>[root@default host admin]# scio const -s s0:flow get sc_flow_reset_on_policy scio: sc_flow_reset_on_policy = 0x0</pre> <pre>[root@default host admin]# scio const -s s0:flow set sc_flow_reset_on_policy 1 scio: setting sc_flow_reset_on_policy to 0x1</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>
sc_num_policies	<p>Gets or sets the number of policies maintained in the flow table</p> <p>For IDP75 and IDP200, the default is 1, and you cannot override the default.</p> <p>For IDP8200 and IDP250, the default is 2. Possible values are 1 or 2.</p> <p>For IDP1100, IDP800, and IDP600, the default is 2. Possible values are 1, 2, 3, or 4.</p> <pre>[root@default host admin]# scio const -s s0 get sc_num_policies scio: sc_num_policies = 0x2</pre> <pre>[root@default host admin]# scio const -s s0 set sc_num_policies 4 scio: sc_num_policies = 0x4</pre>

Table 141 on page 515 provides usage and examples of kernel constants related to GRE decapsulation.

Table 141: scio const Arguments Related to GRE Decapsulation

Constants and Values	Usage and Examples
sc_gre_decapsulation	<p>Gets or sets the constant that determines whether GRE decapsulation is enabled or disabled.</p> <p>The default is 0 (off). 1 turns GRE decapsulation on.</p> <pre>[root@default host admin]# scio const -s s0 get sc_gre_decapsulation scio: sc_gre_decapsulation = 0x0</pre> <pre>[root@default host admin]# scio const -s s0 set sc_gre_decapsulation 1 scio: setting sc_gre_decapsulation to 0x1</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>
sc_max_decapsulation	<p>Gets or sets the constant that determines how many layers can be decapsulated.</p> <p>The default is 1 (1 layer).</p> <p>Possible values 1, 2.</p> <pre>[root@default host admin]# scio const -s s0 get sc_max_decapsulation scio: sc_max_decapsulation = 0x1</pre> <pre>[root@default host admin]# scio const -s s0 set sc_max_decapsulation 2 scio: setting sc_max_decapsulation to 0x2</pre> <p><b>NOTE:</b> The <b>sc_max_decapsulation</b> constant is used with GRE, GTP, and IPsec ESP NULL decapsulation.</p>

Table 142 on page 515 provides usage and examples of kernel constants related to GTP decapsulation.

Table 142: scio const Arguments Related to GTP Decapsulation

Constants and Values	Usage and Examples
sc_gtp_decapsulation	<p>Gets or sets the constant that determines whether GTP decapsulation is enabled or disabled.</p> <p>The default is 0 (off). 1 turns GTP decapsulation on.</p> <pre>[root@default host admin]# scio const -s s0 get sc_gtp_decapsulation scio: sc_gtp_decapsulation = 0x0</pre> <pre>[root@default host admin]# scio const -s s0 set sc_gtp_decapsulation 1 scio: setting sc_gtp_decapsulation to 0x1</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>

Table 142: scio const Arguments Related to GTP Decapsulation (*continued*)

Constants and Values	Usage and Examples
sc_max_decapsulation	<p>Gets or sets the constant that determines how many layers can be decapsulated.</p> <p>The default is 1 (1 layer).</p> <p>Possible values 1, 2.</p> <pre>[root@default host admin]# scio const -s s0 get sc_max_decapsulation scio: sc_max_decapsulation = 0x1</pre> <pre>[root@default host admin]# scio const -s s0 set sc_max_decapsulation 2 scio: setting sc_max_decapsulation to 0x2</pre> <p><b>NOTE:</b> The <b>sc_max_decapsulation</b> constant is used with GRE, GTP, and IPsec ESP NULL decapsulation.</p>
sc_gtp_timeout	<p>Gets or sets the constant that determines the time in seconds that the IDP engine maintains the GTP tunnel. If the time elapses before the IDP engine detects another GTP packet, it considers the tunnel closed.</p> <p>The default is 3600 (seconds).</p> <p>Possible values: 1-0xFFFFFFFF.</p> <pre>[root@default host admin]# scio const -s s0 get sc_gtp_timeout scio: sc_gtp_timeout = 0xe10</pre> <pre>[root@default host admin]# scio const -s s0 set sc_gtp_timeout 7200 scio: setting sc_gtp_timeout to 0x1c20</pre>
sc_gtp_max_flows	<p>Gets or sets the constant that determines maximum number of GTP tunnels the IDP engine can handle at once.</p> <p>The default is 0x30D40 (200,000).</p> <p>Possible values: 2-0x61A80 (2-400,000).</p> <pre>[root@default host admin]# scio const -s s0 get sc_gtp_max_flows scio: sc_gtp_max_flows = 0x30d40</pre> <pre>[root@default host admin]# scio const -s s0 set sc_gtp_max_flows 100000 scio: setting sc_gtp_max_flows to 0x186a0</pre>

Table 143 on page 517 provides usage and examples of kernel constants related to IPsec ESP NULL decapsulation.



Table 143: scio const Arguments Related to IPsec ESP NULL Decapsulation

Constants and Values	Usage and Examples
sc_null_esp_decapsulation	<p>Gets or sets the constant that determines whether IPsec ESP NULL traffic decapsulation is enabled or disabled.</p> <p>The default is 0 (off). 1 turns IPsec ESP NULL traffic decapsulation on.</p> <pre>[root@default host admin]# scio const -s s0 get sc_null_esp_decapsulation scio: sc_null_esp_decapsulation = 0x0</pre> <pre>[root@default host admin]# scio const -s s0 set sc_null_esp_decapsulation 1 scio: setting sc_null_esp_decapsulation to 0x1</pre>
sc_max_decapsulation	<p>Gets or sets the constant that determines how many layers can be decapsulated.</p> <p>The default is 1 (1 layer).</p> <p>Possible values 1, 2.</p> <pre>[root@default host admin]# scio const -s s0 get sc_max_decapsulation scio: sc_max_decapsulation = 0x1</pre> <pre>[root@default host admin]# scio const -s s0 set sc_max_decapsulation 2 scio: setting sc_max_decapsulation to 0x2</pre> <p><b>NOTE:</b> The <b>sc_max_decapsulation</b> constant is used with GRE, GTP, and IPsec ESP NULL decapsulation.</p>

[Table 144 on page 517](#) provides usage and examples of kernel constants related to MPLS decapsulation.

Table 144: scio const Arguments Related to MPLS Decapsulation

Constants and Values	Usage and Examples
sc_mpls_decapsulation	<p>Gets or sets the constant that determines whether MPLS decapsulation is enabled or disabled.</p> <p>The default is 0 (off). 1 turns MPLS decapsulation on.</p> <pre>[root@default host admin]# scio const -s s0 get sc_mpls_decapsulation scio: sc_mpls_decapsulation = 0x0</pre> <pre>[root@default host admin]# scio const -s s0 set sc_mpls_decapsulation 1 scio: sc_mpls_decapsulation = 0x1</pre>

[Table 145 on page 518](#) provides usage and examples of kernel constants related to SSL inspection.

Table 145: scio const Arguments Related to SSL Inspection

Constants and Values	Usage and Examples
sc_ssl_decryption	<p>Gets or sets the constant that determines whether SSL decryption is enabled or disabled.</p> <p>The default is 0 (off). 1 turns the feature on.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_ssl_decryption scio: sc_ssl_decryption = 0x0</pre> <pre>[root@defaulthost admin]# scio const -s s0 set sc_ssl_decryption 1 scio: setting sc_ssl_decryption to 0x1</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>
sc_ssl_inspection	<p>Turns off the SSL forward proxy feature. Use this command in test or troubleshooting cases. Note you can also disable the feature using <b>scio ssl ca delete</b> to delete the root CA. We recommend you use <b>scio const -s s0 set sc_ssl_inspection 0</b> when testing or troubleshooting; and <b>scio ssl ca delete</b> when turning the feature off in production.</p> <p>The default is 1 (on). 0 turns the feature off.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_ssl_inspection scio: sc_ssl_inspection = 0x1</pre> <pre>[root@defaulthost admin]# scio const -s s0 set sc_ssl_inspection 0 scio: setting sc_ssl_inspection to 0x0</pre>
sc_ssl_sessid_timeout	<p>Gets or sets the constant that determines the SSL session security parameter cache timeout value (seconds).</p> <p>The default is 60.</p> <p>Possible values: 1–120.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_ssl_sessid_timeout scio: sc_ssl_sessid_timeout = 0x3c</pre> <pre>[root@defaulthost admin]# scio const -s s0 set sc_ssl_sessid_timeout 45 scio: setting sc_ssl_sessid_timeout to 0x2d</pre>
sc_ssl_pending_sessid_timeout	<p>Gets or sets the constant that determines the SSL pending session security parameter cache timeout value (seconds).</p> <p>The default is 30.</p> <p>Possible values: 1–60.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_ssl_pending_sessid_timeout scio: sc_ssl_pending_sessid_timeout = 0x1e</pre> <pre>[root@defaulthost admin]# scio const -s s0 set sc_ssl_pending_sessid_timeout 45 scio: setting sc_ssl_pending_sessid_timeout to 0x2d</pre>

Table 145: scio const Arguments Related to SSL Inspection (*continued*)

Constants and Values	Usage and Examples
sc_ssl_num_decrypt_sessions	<p>Gets or sets the constant that determines the maximum number of sessions that can be decrypted concurrently.</p> <p>The default is 10,000.</p> <p>Possible values: 1-100,000.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_ssl_num_decrypt_sessions scio: sc_ssl_num_decrypt_sessions = 0x2710  [root@defaulthost admin]# scio const -s s0 set sc_ssl_num_decrypt_sessions 20000 scio: setting sc_ssl_num_decrypt_sessions to 0x4e20</pre>

[Table 146 on page 519](#) provides usage and examples of the kernel constant that determines the maximum frame size processed by the IDP Series device.

Table 146: scio const Arguments Related to Maximum Frame Size

Constants and Values	Usage and Examples
sc_max_frame_size	<p>Gets or sets the constant that determines maximum frame size.</p> <p>The default is 9014 (support for jumbo frames).</p> <p>Possible values: 1514–16,014.</p> <pre>[root@defaulthost admin]# scio const -s s0 get sc_max_frame_size scio: sc_max_frame_size = 0x2336  [root@defaulthost admin]# scio const -s s0 set sc_max_frame_size 1514 scio: sc_max_frame_size = 0x5EA</pre>

[Table 147 on page 519](#) provides usage and examples of the kernel constants related to the SYN Protector rulebase.

Table 147: scio const Arguments Related to the SYN Protector Rulebase

Constants and Values	Usage and Examples
sc_syndef_timeout	<p>Gets or sets the constant that determines the timeout for the SYN protector rulebase in passive mode. The timeout specifies how many seconds the IDP system holds an incomplete SYN-ACK handshake before purging it.</p> <p>The default is 5 (seconds).</p> <p>Possible values: 1-0xFFFF.</p> <pre>[root@defaulthost admin]# scio const -s s0:syndef get sc_syndef_timeout scio: sc_syndef_timeout = 0x5  [root@defaulthost admin]# scio const -s s0:syndef set sc_syndef_timeout 10 scio: setting sc_syndef_timeout to 0xa</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>

Table 147: scio const Arguments Related to the SYN Protector Rulebase (*continued*)

Constants and Values	Usage and Examples
sc_syndef_threshhold	<p>Gets or sets the value for the constant that determines the lower threshold of SYNs per second that activates the SYN Protector rulebase. For relay mode, this is the only value that matters. For passive mode, you also set <b>sc_syndef_threshhold_delta</b>.</p> <p>The default is 0x3E8 (1000).</p> <p>Possible values: 1-0xFFFF.</p> <pre>[root@default host admin]# scio const -s s0:syndef get sc_syndef_threshhold scio: sc_syndef_threshhold = 0x3e8</pre> <pre>[root@default host admin]# scio const -s s0:syndef set sc_syndef_threshhold 1020 scio: setting sc_syndef_threshhold to 0x3fc</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>
sc_syndef_threshhold_delta	<p>Gets or sets the value for the constant that sets the upper threshold of SYNs per second. In passive mode, SYN Protection activates once the number of SYN packets per second for a given destination IP exceeds this number plus the lower threshold number. Passive mode protection deactivates once the value drops below the lower threshold.</p> <p>The default is 0x14 (20).</p> <p>Possible values: 1-0xFFFF.</p> <pre>[root@default host admin]# scio const -s s0:syndef get sc_syndef_threshhold_delta scio: sc_syndef_threshhold_delta = 0x14</pre> <pre>[root@default host admin]# scio const -s s0:syndef set sc_syndef_threshhold_delta 25 scio: setting sc_syndef_threshhold_delta to 0x19</pre> <p><b>NOTE:</b> You can also configure this setting in NSM.</p>
sc_syndef_report_freq	<p>Gets or sets the value for the constant that determines how often a SYN flood attempt is reported, in seconds.</p> <p>The default is 30 (seconds).</p> <p>Possible values: 1-86,400 (86,400 seconds is 1 day).</p> <pre>[root@default host admin]# scio const -s s0:syndef get sc_syndef_report_freq scio: sc_syndef_report_freq = 0x1e</pre> <pre>[root@default host admin]# scio const -s s0:syndef set sc_syndef_report_freq 60 scio: setting sc_syndef_report_freq to 0x3c</pre>
sc_syndef_log_detail	<p>Gets or sets the constant that determines whether or not the destination IP address appears in the log variable data.</p> <p>The default is 1 (on).</p> <p>Possible values: 0-1 (0 = off, 1 = on).</p> <pre>[root@default host admin]# scio const -s s0:syndef get sc_syndef_log_detail scio: sc_syndef_log_detail = 0x0</pre> <pre>[root@default host admin]# scio const -s s0:syndef set sc_syndef_log_detail 1 scio: setting sc_syndef_log_detail to 0x1</pre>

Table 147: scio const Arguments Related to the SYN Protector Rulebase (*continued*)

Constants and Values	Usage and Examples
sc_syndef_log_ports	<p>Gets or sets the value for the constant that determines whether or not the destination port appears in the log variable data. If both sc_syndef_log_detail and sc_syndef_log_ports are set to 1 (on), the sc_syndef_log_ports value takes precedence and is displayed, not the IP.</p> <p>The default is 0 (off).</p> <p>Possible values: 0-1 (0 = off, 1 = on).</p> <pre>[root@default host admin]# scio const -s s0:syndef get sc_syndef_log_ports scio: sc_syndef_log_ports = 0x0  [root@default host admin]# scio const -s s0:syndef set sc_syndef_log_ports 1 scio: setting sc_syndef_log_ports to 0x1</pre>

Table 148 on page 521 provides usage and examples of kernel constants related to the user role-based policy feature.

Table 148: scio const Arguments Related to the User Role-Based Policy Feature

Constants and Values	Usage and Examples
sc_enable_user_policy	<p>Gets or sets the constant that determines whether the feature is enabled or disabled.</p> <p>The default is 1 (on). 0 turns the feature off.</p> <pre>[root@default host admin]# scio const -s s0 get sc_enable_user_policy scio: sc_enable_user_policy = 0x1  [root@default host admin]# scio const -s s0 set sc_enable_user_policy 0 scio: setting sc_enable_user_policy to 0x0</pre>
sc_ic_reconcile_timeout	<p>Gets or sets the threshold where lost connectivity stops processing of user role-based rules.</p> <p>The default is 30 (seconds).</p> <p>Possible values 0-3600.</p> <pre>[root@default host admin]# scio const -s s0 get sc_ic_reconcile_timeout scio: sc_ic_reconcile_timeout = 0x1e  [root@default host admin]# scio const -s s0 set sc_ic_reconcile_timeout 3600 scio: setting sc_ic_reconcile_timeout to 0xe10</pre>

## scio counter

**Syntax** `scio counter { list | get class | reset class }`

**Description** Lists available counters, gets counts, or resets counts.

**Options** [Table 149 on page 522](#) describes **scio counter** options and arguments and provides examples of command syntax.

**Table 149: Command Reference: scio var**

Options	Usage and Example																																
list	<p>Displays a list of available counters.</p> <pre>[root@default host admin]# scio counter list</pre> <table> <tr> <th>Name</th><th>#counters</th></tr> <tr> <td>kpp</td><td>47</td></tr> <tr> <td>reass</td><td>27</td></tr> <tr> <td>ids</td><td>24</td></tr> <tr> <td>flow</td><td>41</td></tr> <tr> <td>fragment</td><td>10</td></tr> <tr> <td>dfa</td><td>6</td></tr> <tr> <td>arp</td><td>9</td></tr> <tr> <td>log</td><td>24</td></tr> <tr> <td>pmanager</td><td>2</td></tr> <tr> <td>ssl</td><td>26</td></tr> <tr> <td>turbo</td><td>18</td></tr> <tr> <td>misc</td><td>31</td></tr> <tr> <td>ai</td><td>21</td></tr> <tr> <td>memory</td><td>6</td></tr> <tr> <td>proxy</td><td>63</td></tr> </table>	Name	#counters	kpp	47	reass	27	ids	24	flow	41	fragment	10	dfa	6	arp	9	log	24	pmanager	2	ssl	26	turbo	18	misc	31	ai	21	memory	6	proxy	63
Name	#counters																																
kpp	47																																
reass	27																																
ids	24																																
flow	41																																
fragment	10																																
dfa	6																																
arp	9																																
log	24																																
pmanager	2																																
ssl	26																																
turbo	18																																
misc	31																																
ai	21																																
memory	6																																
proxy	63																																

Table 149: Command Reference: scio var (*continued*)

Options	Usage and Example																																																		
get <i>class</i>	<p>Gets counter values for the specified class.</p> <pre>[root@defaulthost admin]# scio counter get ids</pre> <table> <thead> <tr> <th>Name</th><th>Value</th></tr> </thead> <tbody> <tr><td>sc_ids_tcp_fast_path</td><td>121</td></tr> <tr><td>sc_ids_l4_anomalies</td><td>0</td></tr> <tr><td>sc_ids_anomaly_hash_miss</td><td>14</td></tr> <tr><td>sc_ids_match_line</td><td>0</td></tr> <tr><td>sc_ids_match_stream256</td><td>0</td></tr> <tr><td>sc_ids_match_stream</td><td>0</td></tr> <tr><td>sc_ids_match_packet</td><td>0</td></tr> <tr><td>sc_ids_match_packet_header</td><td>0</td></tr> <tr><td>sc_ids_match_context</td><td>9</td></tr> <tr><td>sc_ids_match_regex</td><td>0</td></tr> <tr><td>sc_ids_tail_dfa</td><td>21</td></tr> <tr><td>sc_ids_exempt_attacks</td><td>0</td></tr> <tr><td>sc_ids_chain_out_of_order</td><td>0</td></tr> <tr><td>sc_ids_chain_partial_match</td><td>0</td></tr> <tr><td>sc_ids_device_fifo_size</td><td>0</td></tr> <tr><td>sc_ids_device_fifo_overflow</td><td>0</td></tr> <tr><td>sc_ids_brute_force_q_size</td><td>0</td></tr> <tr><td>sc_ids_cache_hit</td><td>0</td></tr> <tr><td>sc_ids_cache_miss</td><td>0</td></tr> <tr><td>sc_ids_shellcode_func</td><td>0</td></tr> <tr><td>sc_ids_linebreak_hw</td><td>0</td></tr> <tr><td>sc_ids_linebreak_sw</td><td>154</td></tr> <tr><td>ipblocker_dropped</td><td>0</td></tr> <tr><td>sc_ids_drop_pkts</td><td>0</td></tr> </tbody> </table>	Name	Value	sc_ids_tcp_fast_path	121	sc_ids_l4_anomalies	0	sc_ids_anomaly_hash_miss	14	sc_ids_match_line	0	sc_ids_match_stream256	0	sc_ids_match_stream	0	sc_ids_match_packet	0	sc_ids_match_packet_header	0	sc_ids_match_context	9	sc_ids_match_regex	0	sc_ids_tail_dfa	21	sc_ids_exempt_attacks	0	sc_ids_chain_out_of_order	0	sc_ids_chain_partial_match	0	sc_ids_device_fifo_size	0	sc_ids_device_fifo_overflow	0	sc_ids_brute_force_q_size	0	sc_ids_cache_hit	0	sc_ids_cache_miss	0	sc_ids_shellcode_func	0	sc_ids_linebreak_hw	0	sc_ids_linebreak_sw	154	ipblocker_dropped	0	sc_ids_drop_pkts	0
Name	Value																																																		
sc_ids_tcp_fast_path	121																																																		
sc_ids_l4_anomalies	0																																																		
sc_ids_anomaly_hash_miss	14																																																		
sc_ids_match_line	0																																																		
sc_ids_match_stream256	0																																																		
sc_ids_match_stream	0																																																		
sc_ids_match_packet	0																																																		
sc_ids_match_packet_header	0																																																		
sc_ids_match_context	9																																																		
sc_ids_match_regex	0																																																		
sc_ids_tail_dfa	21																																																		
sc_ids_exempt_attacks	0																																																		
sc_ids_chain_out_of_order	0																																																		
sc_ids_chain_partial_match	0																																																		
sc_ids_device_fifo_size	0																																																		
sc_ids_device_fifo_overflow	0																																																		
sc_ids_brute_force_q_size	0																																																		
sc_ids_cache_hit	0																																																		
sc_ids_cache_miss	0																																																		
sc_ids_shellcode_func	0																																																		
sc_ids_linebreak_hw	0																																																		
sc_ids_linebreak_sw	154																																																		
ipblocker_dropped	0																																																		
sc_ids_drop_pkts	0																																																		
reset <i>class</i>	<p>Resets the counters for the specified class.</p> <pre>[root@defaulthost admin]# scio counter reset ids [root@defaulthost admin]#</pre>																																																		

## scio getsystem

---

**Syntax**    `scio getsystem`

**Description**    Displays IDP Series device model information, software version, and license information.

The following example shows the output of the **scio getsystem** command:

```
[root@defaulthost admin]# scio getsystem
Product Name:  NS-IDP-1100C
Serial Number:  XXXXXXXXXXXXXXXX
Software Version:  5.1.136718
IDP Mode:  transparent
HA Mode:  Disabled
Detector Version:  5.1.110100823
Software License:  Permanent
Software Expiration Date:  never
```

**Options**    None



scio idp-cpu-utilization

**Syntax**    `scio [-c idp-engine] idp-cpu-utilization`

**Description**    For multicore platorms, displays CPU utilization of IDP engines. For single core platforms, use the Linux `top` command. For more information, see “[Viewing CPU Utilization](#)” on [page 586](#).



**NOTE:** Using `scio idp-cpu-utilization` can cause CPU usage to spike. We, therefore, recommend you use `scio idp-cpu-utilization` only for debugging.

**Options**    [Table 150 on page 525](#) describes `scio idp-cpu-utilization` options and arguments and provides examples of command syntax.

Table 150: Command Reference: `scio idp-cpu-utilization`

Options	Usage and Example
None	With no options, <code>scio idp-cpu-utilization</code> displays the CPU utilization of all IDP engines.  [root@defaulthost admin]# <code>scio idp-cpu-utilization</code> Current actual cpu utilization: 0
<code>-c <i>idp-engine</i></code>	Displays the CPU utilization for the specified IDP engine. The values for <i>idp-engine</i> are numbers. For IDP8200, the number is 0–5. For other platforms, number is 0.  [root@defaulthost admin]# <code>scio -c 0 idp-cpu-utilization</code> Current actual cpu utilization: 0

## scio logview

**Syntax** `scio logview logfile`

**Description** The purpose of the **scio logview** utility is to allow you to troubleshoot issues with logging features. On the IDP Series device, you can use the **scio logview** utility to view contents of log files before the logs are forwarded to NSM. This way, if you suspect a problem with logging features, you can compare the device-side logs with the NSM-side logs.

Note:

- Run the **scio logview** utility from the `/var/idp/device/logs/` directory.
- Logs that have been read from the NSM Log Viewer get deleted from the IDP Series device, so typically there is not a large collection of logs in `/var/idp/device/logs/`.
- Data includes only a subset of log columns: src ip, src port, dst ip, dst port, category, sub-category, attack id, severity, protocol, action, src interface, and details.
- Packet logs cannot be displayed.

**Additional Information** The following example commands show how to navigate to the logs directory, sort by date, and use the **scio logview** command to display contents of a recent log.

```
[root@defaultthost ~]# cd /var/idp/device/logs/
[root@defaultthost logs]# ls -lat | less
drwx----- 2 idp idp 69632 Aug  5 11:50 .
-rw----- 1 idp idp  2788 Aug  5 11:50 1281034151.log
-rw----- 1 idp idp   212 Aug  5 11:50 1281034242.log
-rw----- 1 idp idp    0 Aug  5 11:50 1281034242.wait
-rw----- 1 idp idp   384 Aug  5 11:49 1281034128.log
-rw----- 1 idp idp  1232 Aug  5 11:48 1281034081.log
-rw----- 1 idp idp  1680 Aug  5 11:47 1281034035.log
-rw----- 1 idp idp   744 Aug  5 11:47 1281033989.log
-rw----- 1 idp idp  1868 Aug  5 11:46 1281033942.log
-rw----- 1 idp idp   952 Aug  5 11:45 1281033916.log
-rw----- 1 idp idp   260 Aug  5 11:44 1281033804.log
-rw----- 1 idp idp   260 Aug  5 11:43 1281033699.log
-rw----- 1 idp idp   260 Aug  5 11:41 1281033590.log
-rw----- 1 idp idp   260 Aug  5 11:39 1281033484.log
-rw----- 1 idp idp   260 Aug  5 11:37 1281033386.log
-rw----- 1 idp idp   148 Aug  5 11:36 1281033138.log

[root@defaultthost logs]# scio logview 1281034242.log
Log :Time Generated : Thu Aug  5 11:50:41 2010
  Source IP 0.0.0.0 Source Port :0 -> Destination IP 0.0.0.0 Destination Port :0
Category Enum : attackid :805306379 Severity Enum :SC_LOG_SEVERITY_INFO
Protocol Enum :0 Action :SC_LOG_ACTION_NOT_SET
srcIface : , Details : Percentage of Control CPU usage last 5 minutes has
restored below threshold and is at 57 [Simulation Mode]
```

## scio napp sig list

**Syntax** `scio napp sig list`

**Description** Lists details of the nested application signatures that are relevant to the current policy. The application signatures are relevant if the application is specified or implicated by IDP rulebase or APE rulebase rules.

When verifying or troubleshooting features with the CLI, you might find this command useful for verifying expected behavior.

**Options** The list option is the only argument and is required.

```
[root@defaulthost ~]# scio napp sig list
Nested Application signatures: total 551 show 551
NESTEDAPPLICATION:PRICELINE, HTTP, index 0, nested service 1, max_trans 1, order
  33249, appl_id 794, n_members 1
NESTEDAPPLICATION:ICAST, HTTP, index 1, nested service 2, max_trans 1, order
  33118, appl_id 555, n_members 2
NESTEDAPPLICATION:GOOGLE-TRANSLATE, HTTP, index 2, nested service 3, max_trans
  1, order 32991, appl_id 467, n_members 1
NESTEDAPPLICATION:EBUDDY, HTTP, index 3, nested service 4, max_trans 1, order
  32906, appl_id 278, n_members 1
NESTEDAPPLICATION:TOPFRIENDS, HTTP, index 4, nested service 5, max_trans 1, order
  33299, appl_id 723, n_members 2
NESTEDAPPLICATION:MYSpace-GUARDIAN-ANGELS, HTTP, index 5, nested service 6,
  max_trans 1, order 33169, appl_id 619, n_members 2
NESTEDAPPLICATION:ALLMUSIC-LOOKUP, HTTP, index 6, nested service 7, max_trans 1,
  order 33043, appl_id 530, n_members 2
NESTEDAPPLICATION:HOTMAIL, HTTP, index 7, nested service 8, max_trans 1, order
  32832, appl_id 383, n_members 1
NESTEDAPPLICATION:RAGINGBULL-POST, HTTP, index 8, nested service 9, max_trans 1,
  order 32963, appl_id 354, n_members 2
NESTEDAPPLICATION:FACEBOOK-VISUALBOOKSHELF, HTTP, index 9, nested service 10,
  max_trans 1, order 33227, appl_id 602, n_members 2
NESTEDAPPLICATION:TRIPADVISOR, HTTP, index 10, nested service 11, max_trans 1,
  order 33099, appl_id 579, n_members 1
NESTEDAPPLICATION:SPANKWIRE, HTTP, index 11, nested service 12, max_trans 1, order
  32816, appl_id 505, n_members 1
NESTEDAPPLICATION:THECIRCLE, HTTP, index 12, nested service 13, max_trans 1, order
  32888, appl_id 261, n_members 1

...
```

## scio nic

**Syntax** `scio nic option argument`

**Description** Attach or release the attachment of the IDP OS kernel module to an IDP Series traffic interface network interface card (NIC). By default, the kernel module is attached to all traffic interfaces but not the management interface or high availability (HA) interface. We recommend you retain the defaults. These commands are provided for troubleshooting.

**Options** [Table 151 on page 528](#) describes **scio nic** options and arguments and provides example syntax.

**Table 151: Command Reference: scio nic**

Options	Usage and Examples
<code>attach <i>argument</i></code>	<p>Attaches the IDP OS kernel module to the specified NIC. To attach the kernel module to all NICs, do not specify an argument.</p> <pre>[root@defaulthost admin]# scio nic attach [root@defaulthost admin]#</pre>
<code>release <i>argument</i></code>	<p>Releases the attachment of the kernel module to the specified NIC. To release the attachment for all NICs, do not specify an argument.</p> <pre>[root@defaulthost admin]# scio nic release eth2 [root@defaulthost admin]#</pre>

## scio sri

**Syntax** `scio sri category [list | set] variable value`

**Description** Enables and configures system resource instrumentation options. When enabled, the system resource instrumentation feature collects and reports statistics to the device MIB and sends SNMP traps.

Commands take effect immediately. Upon restart, settings are saved to the `idp.cfg` file. Settings persist across restarts.

System resource instrumentation adds load to the control plane CPU and to the IDP engine. We recommend you disable reporting for statistics you do not use. If you encounter performance issues, you might choose to disable one or more categories.

**Options** [Table 152 on page 529](#) describes **scio sri** variables and provides examples of command syntax.

**Table 152: Command Reference: scio sri**

Category	Usage and Example
resource	<p>Enables or disables statistics reporting and configures trap thresholds for the resource category of statistics.</p> <pre>[root@defaulthost ~]# scio sri resource list SRI resource :   sc_enable_resource_stats      = 1      [ 0 - 1 ]   sc_enable_resource_traps      = 1      [ 0 - 1 ]   thrshld_idp_cpu               = 40     [ 0 - 100 ]   thrshld_session_rate          = 40     [ 0 - 100 ]   thrshld_free_pkt_bufs         = 40     [ 0 - 100 ]   thrshld_active_sessions       = 40     [ 0 - 100 ]</pre> <pre>[root@defaulthost ~]# scio sri resource set thrshld_idp_cpu 90</pre> <p>For additional information, see <a href="#">“Configuring SNMP Reporting for the Resource Category of Statistics” on page 428</a>.</p>
traffic	<p>Enables or disables statistics reporting and configures trap thresholds for the traffic category of statistics.</p> <pre>[root@defaulthost ~]# scio sri traffic list SRI traffic :   sc_enable_traffic_stats       = 1      [ 0 - 1 ]</pre> <pre>[root@defaulthost ~]# scio sri traffic set sc_enable_traffic_stats 0</pre> <p>For additional information, see <a href="#">“Configuring SNMP Reporting for the Traffic Category of Statistics” on page 438</a>.</p>

Table 152: Command Reference: scio sri (*continued*)

Category	Usage and Example
rule	<p>Enables or disables statistics reporting and configures trap thresholds for the rule category of statistics.</p> <pre>[root@default host ~]# scio sri rule list</pre> <p>SRI rule :</p> <pre>sc_enable_rule_stats          = 1      [ 0 - 1 ]</pre> <p>[root@default host ~]# scio sri rule set sc_enable_rule_stat 0</p> <p>For additional information, see <a href="#">"Configuring SNMP Reporting for the Rule Category of Statistics" on page 433</a>.</p>
interface	<p>Enables or disables statistics reporting and configures trap thresholds for the interface category of statistics.</p> <pre>[root@default host ~]# scio sri interface list</pre> <p>SRI interface :</p> <pre>sc_enable_interface_stats    = 1      [ 0 - 1 ] sc_enable_interface_traps    = 1      [ 0 - 1 ] thrshld_rx_pkt_drop_rate_per_if = 40    [ 0 - 100 ] thrshld_rx_pkt_drop_per_if   = 40    [ 0 - 100 ] thrshld_rx_pkt_drop_all_if   = 40    [ 0 - 100 ] thrshld_rx_pkt_drop_rate_all_if = 40    [ 0 - 100 ] thrshld_rx_pkt_drop_overflow = 40    [ 0 - 100 ] thrshld_tx_pkt_drop_per_if   = 40    [ 0 - 100 ] thrshld_tx_pkt_drop_all_if   = 40    [ 0 - 100 ]</pre> <p>[root@default host ~]# scio sri interface set thrshld_rx_pkt_drop_rate_all_if 80</p> <p>For additional information, see <a href="#">"Configuring SNMP Reporting for the Interface Category of Statistics" on page 418</a>.</p>
sensor	<p>Enables or disables statistics reporting and configures trap thresholds for the sensor category of statistics.</p> <pre>[root@default host ~]# scio sri sensor list</pre> <p>SRI sensor :</p> <pre>sc_enable_sensor_stats      = 1      [ 0 - 1 ] sc_enable_sensor_traps      = 1      [ 0 - 1 ] thrshld_memory              = 40     [ 0 - 100 ] thrshld_hard_disk           = 40     [ 0 - 100 ] thrshld_control_cpu         = 40     [ 0 - 100 ]</pre> <p>[root@default host ~]# scio sri sensor set thrshld_control_cpu 90</p> <p>For additional information, see <a href="#">"Configuring SNMP Reporting for the Sensor Category of Statistics" on page 435</a>.</p>
all	<p>Enables or disables statistics reporting and traps for all categories.</p> <pre>[root@default host admin]# [root@default host ~]# scio sri all list</pre> <p>SRI Enable all :</p> <pre>sc_enable_all_stats         = 1      [ 0 - 1 ] sc_enable_all_traps         = 1      [ 0 - 1 ]</pre> <p>[root@default host admin]# [root@default host ~]# scio sri all set sc_enable_all_traps 0</p>

## scio ssl

**Syntax** `scio ssl option argument`

**Description** Manages SSL server keys and certificate authorities (CA) used by the IDP Series device to inspect SSL traffic. Also manages the whitelist of destination servers you want to exempt from decryption and IDP processing.

**Options** [Table 153 on page 531](#) describes **scio ssl** options and arguments and provides examples of command syntax.

**Table 153: Command Reference: scio ssl**

Options	Usage and Examples
list all	<p>Lists all stored SSL keys. Each IDP Series device can store 100 server private keys and 100 servers per key.</p> <pre>[root@default host admin]# scio ssl list all [root@default host admin]#</pre>
list key <i>key-id</i>	<p>Lists all servers associated with a particular key.</p> <pre>[root@default host admin]# scio ssl list key Key-1 [root@default host admin]#</pre>
add key <i>key-path</i> [password <i>password-string</i> ] [server <i>server-ip</i> ]	<p>Adds a key with an optional password and an associated server.</p> <p>Use SCP or FTP to copy your SSL server private key file to the IDP Series device. The IDP Series device does not run an FTP server, so you have to initiate the FTP session from the IDP Series device.</p> <p>Keys must be based on RSA and be in PEM format. We have verified support for the following RSA private key lengths: 1024 bits, 2048 bits, 3072 bits, and 4096 bits.</p> <pre>[root@default host admin]# scio ssl add key /tmp/server.key password P@ss-Strong! server 10.1.1.1 [root@default host admin]#</pre>
add server <i>server-ip</i> key <i>key-id</i>	<p>Associates the specified server with the specified key.</p> <pre>[root@default host admin]# scio ssl add server 10.1.1.1 key server.key [root@default host admin]#</pre>
delete all	<p>Clears the SSL keystore.</p> <pre>[root@default host admin]# scio ssl delete all [root@default host admin]#</pre>
delete key <i>key-id</i> [server <i>server-ip</i> ]	<p>Deletes a particular SSL key from the SSL keystore. To delete a key-server association but not the key, use the server option.</p> <pre>[root@default host admin]# scio ssl delete key server.key server 10.1.1.1 [root@default host admin]#</pre>

Table 153: Command Reference: `scio ssl` (*continued*)

Options	Usage and Examples
<code>ca {create country-code state locality organization organization-unit common-name e-mail [nbits]   delete   export   show}</code>	<p>Use these options to configure the CA used by the SSL forward proxy feature.</p> <p>Command arguments correspond with the values you want to set for the CA:</p> <ul style="list-style-type: none"> <li>• <i>country-code</i>—A two-letter code. This is the C value in the certificate.</li> <li>• <i>state</i>—A string. This is the ST value in the certificate.</li> <li>• <i>locality</i>—A string. This is the L value in the certificate.</li> <li>• <i>organization</i>—A string. This is the O value in the certificate.</li> <li>• <i>organization-unit</i>—A string. This is the OU value in the certificate.</li> <li>• <i>common-name</i>—A string. This is the CN value in the certificate.</li> <li>• <i>e-mail</i>—An e-mail address. This should be an administrative e-mail address for the issuer.</li> <li>• <i>nbits</i> is the RSA private key length. We have verified support for the following RSA private key lengths: 1024 bits, 2048 bits, 3072 bits, and 4096 bits. If you do not specify this option, the key length defaults to 1024 bits.</li> </ul> <p><b>NOTE:</b> Enclose strings that are phrases in single quotation marks.</p> <p>The following example creates a root self-signed CA used by the SSL forward proxy feature:</p> <pre>[root@default host admin]# scio ssl ca create US CA Sunnyvale 'Juniper Networks Inc.' 'SSL Inspection policy' 'Juniper IT Services' 'admin@juniper.net' 1024</pre> <p>The following example displays the CA settings:</p> <pre>[root@default host admin]# scio ssl ca show serial=8E0012848A2D7CCD subject= /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks Inc./OU=SSL Inspection policy/CN=Juniper IT Services/emailAddress=admin@juniper.net issuer= /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks Inc./OU=SSL Inspection policy/CN=Juniper IT Services/emailAddress=admin@juniper.net notBefore=Jun 25 22:13:23 2009 GMT notAfter=Jun 23 22:13:23 2019 GMT</pre>



Table 153: Command Reference: scio ssl (*continued*)

Options	Usage and Examples
	<p>The following example prints to the screen the CA in PEM format. You can copy this to a file and then import this CA into SSL clients, enabling them to validate and trust certificates signed by the IDP Series device:</p> <pre data-bbox="570 493 1339 997">[root@default host admin]# scio ssl ca export -----BEGIN CERTIFICATE----- MIIC1TCCAj4CCQCOABKEi i 18zTANBgkqhki G9w0BAQUFADCB r jELMAkGA1UEBhMC VVMxCzAJBgNVBAGTAKNBMRIwEAYDVQQHEw1TdW5ueXZhbGUxHjAcBgNVBAoTFUp1 bm1wZXI gTmV0d29ya3MgSW5jLjEeMBwGA1UECXMVU1NMIE1uc3B1Y3Rpb24gcG9s aWN5MRwwGgYDVQQDExNKdw5pcGVyIE1UIFN1cnZpY2VzMSAwHgYJKoZIhvcNAQkB FhFhZG1pbk BqdW5pcGVyLm51dDAeFw0wOTA2MjUyMjEz MjNaFw0xOTA2MjMyMjEz MjNaMIGuMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExEjAQBgNVBACTCVN1bm55 dmFsZTEeMBwGA1UEChMVSnVuaXB1ciB0ZXR3b3JrcyBJbmMuMR4wHAYDVQQLExVT U0wgSW5zcGVjdG1vbiBwb2xpY3kxHDAaBgNVBAMTE0p1bm1wZXI gSVQgU2Vydm1j ZXMxIDAeBgkqhki G9w0BCEwEWFkbW1uQGp1bm1wZXI ubmV0MIGfMA0GCSqGSIb3 DQEBAQUAA4GNADCBiQKBgQDAsn2NFaXTrCpShf9sg+Ccn1rUYzPuVHTw1GUtnHHB o/oFXeNGETggLZ/jck+L2710x3IpGd67yyHs08sXWvgC3MJukb14kqyTyguy3/E9 wkiIey8W4XzyBXrCfW2YEgMc0cFExdm+C6DrAai1ddTQdgelxZ7nfIj24iiBhYYM GQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAFTrEz9DHcbohDJFqGWPjS+MDgsX9041 f/WzHXftak4ZHjOryYvVaRuyitEhMX1KvMPQjYXf+TE2vF9yYqmoCj6710Liu2ZJ Tw4gwy9E9p58krqvZu4F2/kVM+yEAKsUIjBme1RIL6Az3kLauHvkyAbMcSFZG2b0 7Z8WbQqn3o6s -----END CERTIFICATE-----</pre> <p>Deleting a CA effectively turns off the SSL forward proxy feature. The following example deletes the CA:</p> <pre data-bbox="570 1087 1052 1144">[root@default host admin]# scio ssl ca delete [root@default host admin]#</pre>

Table 153: Command Reference: `scio ssl` (*continued*)

Options	Usage and Examples
<code>whitelist {import <i>filepathname</i>   export }</code>	<p>Imports or exports a whitelist file. A whitelist file is a list of IP addresses and domain names for destination servers for which traffic should not be inspected. The file must be reachable by the <i>filepathname</i> you specify. We recommend you store the file in the IDP Series device <b>/tmp</b> directory.</p> <p>Traffic that matches a whitelist entry is passed through (not decrypted or inspected).</p> <p>The following example shows the format of a whitelist file:</p> <pre>10.0.0.1 1.0.0.0/8 70.34.21.82 trustedsite.com landing.trustedsearch.com</pre> <p>Each line in the whitelist file specifies the IP address or domain name for a destination server. To whitelist multiple sites with one entry, you can use an IP prefix to match address blocks and a domain suffix to include all subdomains.</p> <p>The domain name in your whitelist should match the common name entry in the certificate presented by the destination server. For example, suppose the certificate for the E-Trade HTTPS server contains the following subject:</p> <pre>C=US, ST=Georgia, L=Alpharetta, O=ETRADE FINANCIAL CORPORATION, OU=Global Information Security, CN=us.etrade.com</pre> <p>You can whitelist this site by adding either <b>us.etrade.com</b> or the domain suffix <b>etrade.com</b> to your whitelist file.</p> <p>The following example shows the syntax for the import option.</p> <pre>[root@default host admin]# scio ssl whitelist import /tmp/whitelist.txt [root@default host admin]#</pre> <p><b>NOTE:</b> To update the active whitelist, import an updated whitelist file. To clear the whitelist, import a file that contains only one empty line.</p> <p>The following example shows the syntax for the export option. The export option prints the active whitelist to the screen.</p> <pre>[root@default host admin]# scio ssl whitelist export 10.0.0.1 1.0.0.0/8 70.34.21.82 trustedsite.com landing.trustedsearch.com</pre>

## scio subs

**Syntax** `scio subs option argument`

**Description** Displays statistics for the IDP subscriber and enables you to manage subscriber settings. The IDP subscriber is a process that associates traffic with the IDP engine. By default, all virtual circuits belong to the subscriber named s0. We test and support only configurations where the default subscriber s0 is used.

**Options** [Table 154 on page 535](#) describes options and arguments to the **scio subs** command and provides examples of command syntax.

**Table 154: Command Reference: scio subs**

Options	Usage and Examples																																				
list	<p>Lists the virtual circuits and NICs associated with the subscriber s0.</p> <pre>[root@defaulthost admin]# scio subs list</pre> <p>Defined Subscribers:</p> <table><tr><th>Subscriber</th><th>V-Circuit</th><th>NIC</th></tr><tr><td>-----</td><td>-----</td><td>----</td></tr><tr><td>s0</td><td>eth11</td><td>eth11</td></tr><tr><td></td><td>eth10</td><td>eth10</td></tr><tr><td></td><td>eth9</td><td>eth9</td></tr><tr><td></td><td>eth8</td><td>eth8</td></tr><tr><td></td><td>eth7</td><td>eth7</td></tr><tr><td></td><td>eth6</td><td>eth6</td></tr><tr><td></td><td>eth5</td><td>eth5</td></tr><tr><td></td><td>eth4</td><td>eth4</td></tr><tr><td></td><td>eth3</td><td>eth3</td></tr><tr><td></td><td>eth2</td><td>eth2</td></tr></table>	Subscriber	V-Circuit	NIC	-----	-----	----	s0	eth11	eth11		eth10	eth10		eth9	eth9		eth8	eth8		eth7	eth7		eth6	eth6		eth5	eth5		eth4	eth4		eth3	eth3		eth2	eth2
Subscriber	V-Circuit	NIC																																			
-----	-----	----																																			
s0	eth11	eth11																																			
	eth10	eth10																																			
	eth9	eth9																																			
	eth8	eth8																																			
	eth7	eth7																																			
	eth6	eth6																																			
	eth5	eth5																																			
	eth4	eth4																																			
	eth3	eth3																																			
	eth2	eth2																																			
aggregatestatus <i>subscriber</i>	<p>For IDP8200, use this option instead of <b>scio subs stats s0</b> to display aggregated status statistics for the IDP Series device. The <b>scio subs stats s0</b> displays status per IDP engine.</p> <pre>[root@defaulthost admin]# scio subs aggregatestatus s0</pre> <p>Aggregate Status for subs 's0'</p> <table><tr><td>Packets/second:</td><td>54</td><td>peak: 4000</td></tr><tr><td>KBits/second:</td><td>360</td><td>peak: 15207</td></tr><tr><td>Packets received:</td><td colspan="2">icmp 63580, tcp 15663286, udp 15550659, other 16125996</td></tr><tr><td>Current flows:</td><td colspan="2">icmp 0, tcp 1680, udp 26104, other 8288</td></tr><tr><td>Current sessions:</td><td colspan="2">icmp 0, tcp 840, udp 8702, other 4144</td></tr><tr><td>Current bypassed flows :</td><td colspan="2">0</td></tr><tr><td>Current policy:</td><td colspan="2">Recommended v0</td></tr></table>	Packets/second:	54	peak: 4000	KBits/second:	360	peak: 15207	Packets received:	icmp 63580, tcp 15663286, udp 15550659, other 16125996		Current flows:	icmp 0, tcp 1680, udp 26104, other 8288		Current sessions:	icmp 0, tcp 840, udp 8702, other 4144		Current bypassed flows :	0		Current policy:	Recommended v0																
Packets/second:	54	peak: 4000																																			
KBits/second:	360	peak: 15207																																			
Packets received:	icmp 63580, tcp 15663286, udp 15550659, other 16125996																																				
Current flows:	icmp 0, tcp 1680, udp 26104, other 8288																																				
Current sessions:	icmp 0, tcp 840, udp 8702, other 4144																																				
Current bypassed flows :	0																																				
Current policy:	Recommended v0																																				
attach <i>subscriber vc-name</i>	<p>Associates a virtual circuit with the subscriber instance.</p> <pre>[root@defaulthost admin]# scio subs attach s0 eth2</pre>																																				
overflow [ <i>get subscriber   set subscriber overflow_module threshold   change subscriber overflow_module threshold</i> ]	<p>Gets or sets overflow parameters.</p> <pre>[root@defaulthost admin]# scio subs overflow get s0</pre> <p>subs overflow: subscriber=s0 mode=0 threshold=0</p>																																				

Table 154: Command Reference: scio subs (*continued*)

Options	Usage and Examples
qmodules <i>subscriber</i>	<p>Lists qmodules associated with a subscriber. A qmodule is a module of code related to an IDP Series function or feature.</p> <pre>[root@defaulthost admin]# scio subs qmodules s0 Qmodules for subs 's0'     flow - Performs flow lookups, flow/session creation and policy lookups     ape - Application Policy Enforcement     ipblocker - IDS ip action module     pre-ids filter - Weeds out unwanted sessions before entering the IDS modules      tsig - Performs Traffic Signature detection     seqack - Translates TCP SEQ/ACK numbers     syndef - Provides defense against SYN attack     portfaker - Fakes active ports on the network to catch hackers     reass - Tracks a TCP connection and reorders packets     ptype - Detects protocol type using content and statistical analysis      ids - Detects intrusion attempts based on a library of attack signatures     backdoor - Detects backdoor activity using statistical analysis     iprouter - Routes packets to the appropriate virtual circuit</pre>

Table 154: Command Reference: scio subs (*continued*)

Options	Usage and Examples																																																																																																																																																																																																				
qmodstats subscriber	<div>Displays statistics and counters aggregated by qmodule.</div> <div><pre>[root@defaultthost admin]# scio subs qmodstats s0</pre><p>Qmodules Statistics for subs 's0' (time in micro seconds)</p><table><thead><tr><th>Q-Module</th><th>Min.Time</th><th>Max.Time</th><th>Ave.Time</th><th>#Pkt.</th><th>#Pkt.Drop</th><th>#Pkt.Error</th></tr></thead><tbody><tr><td>flow</td><td>0</td><td>0</td><td>0</td><td>1373573</td><td>194</td><td>0</td></tr><tr><td>ape</td><td>0</td><td>0</td><td>0</td><td>97130</td><td>37288</td><td>0</td></tr><tr><td>ipblocker</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>pre-ids filter</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>tsig</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>seqack</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>syndef</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>portfaker</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>reass</td><td>0</td><td>0</td><td>0</td><td>1095300</td><td>0</td><td>0</td></tr><tr><td>pptype</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>ids</td><td>0</td><td>0</td><td>0</td><td>41882</td><td>0</td><td>0</td></tr><tr><td>backdoor</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>iprouter</td><td>0</td><td>0</td><td>0</td><td>1336112</td><td>0</td><td>0</td></tr></tbody></table><p>Qmodules Performance Monitor Counters for subs 's0' (average count per packet)</p><table><thead><tr><th>Q-Module</th><th>Cycles</th><th>Insts</th><th>CPI</th><th>Misses</th><th>Hits</th><th>#Pkt.</th></tr></thead><tbody><tr><td>flow</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>1373573</td></tr><tr><td>ape</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>97130</td></tr><tr><td>ipblocker</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>pre-ids filter</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>tsig</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>seqack</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>syndef</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>portfaker</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>reass</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>1095300</td></tr><tr><td>pptype</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>ids</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>41882</td></tr><tr><td>backdoor</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>0</td></tr><tr><td>iprouter</td><td>0</td><td>0</td><td>0.00</td><td>0</td><td>0</td><td>1336112</td></tr></tbody></table></div>	Q-Module	Min.Time	Max.Time	Ave.Time	#Pkt.	#Pkt.Drop	#Pkt.Error	flow	0	0	0	1373573	194	0	ape	0	0	0	97130	37288	0	ipblocker	0	0	0	0	0	0	pre-ids filter	0	0	0	0	0	0	tsig	0	0	0	0	0	0	seqack	0	0	0	0	0	0	syndef	0	0	0	0	0	0	portfaker	0	0	0	0	0	0	reass	0	0	0	1095300	0	0	pptype	0	0	0	0	0	0	ids	0	0	0	41882	0	0	backdoor	0	0	0	0	0	0	iprouter	0	0	0	1336112	0	0	Q-Module	Cycles	Insts	CPI	Misses	Hits	#Pkt.	flow	0	0	0.00	0	0	1373573	ape	0	0	0.00	0	0	97130	ipblocker	0	0	0.00	0	0	0	pre-ids filter	0	0	0.00	0	0	0	tsig	0	0	0.00	0	0	0	seqack	0	0	0.00	0	0	0	syndef	0	0	0.00	0	0	0	portfaker	0	0	0.00	0	0	0	reass	0	0	0.00	0	0	1095300	pptype	0	0	0.00	0	0	0	ids	0	0	0.00	0	0	41882	backdoor	0	0	0.00	0	0	0	iprouter	0	0	0.00	0	0	1336112
Q-Module	Min.Time	Max.Time	Ave.Time	#Pkt.	#Pkt.Drop	#Pkt.Error																																																																																																																																																																																															
flow	0	0	0	1373573	194	0																																																																																																																																																																																															
ape	0	0	0	97130	37288	0																																																																																																																																																																																															
ipblocker	0	0	0	0	0	0																																																																																																																																																																																															
pre-ids filter	0	0	0	0	0	0																																																																																																																																																																																															
tsig	0	0	0	0	0	0																																																																																																																																																																																															
seqack	0	0	0	0	0	0																																																																																																																																																																																															
syndef	0	0	0	0	0	0																																																																																																																																																																																															
portfaker	0	0	0	0	0	0																																																																																																																																																																																															
reass	0	0	0	1095300	0	0																																																																																																																																																																																															
pptype	0	0	0	0	0	0																																																																																																																																																																																															
ids	0	0	0	41882	0	0																																																																																																																																																																																															
backdoor	0	0	0	0	0	0																																																																																																																																																																																															
iprouter	0	0	0	1336112	0	0																																																																																																																																																																																															
Q-Module	Cycles	Insts	CPI	Misses	Hits	#Pkt.																																																																																																																																																																																															
flow	0	0	0.00	0	0	1373573																																																																																																																																																																																															
ape	0	0	0.00	0	0	97130																																																																																																																																																																																															
ipblocker	0	0	0.00	0	0	0																																																																																																																																																																																															
pre-ids filter	0	0	0.00	0	0	0																																																																																																																																																																																															
tsig	0	0	0.00	0	0	0																																																																																																																																																																																															
seqack	0	0	0.00	0	0	0																																																																																																																																																																																															
syndef	0	0	0.00	0	0	0																																																																																																																																																																																															
portfaker	0	0	0.00	0	0	0																																																																																																																																																																																															
reass	0	0	0.00	0	0	1095300																																																																																																																																																																																															
pptype	0	0	0.00	0	0	0																																																																																																																																																																																															
ids	0	0	0.00	0	0	41882																																																																																																																																																																																															
backdoor	0	0	0.00	0	0	0																																																																																																																																																																																															
iprouter	0	0	0.00	0	0	1336112																																																																																																																																																																																															
release subscriber vc-name	<div>Releases the association that was created with <code>scio subs attach</code>.</div> <div><pre>[root@defaultthost admin]# scio subs release s0 eth2</pre></div>																																																																																																																																																																																																				
reset subscriber	<div>Resets statistics.</div> <div><pre>[root@defaultthost admin]# scio subs reset s0</pre></div>																																																																																																																																																																																																				
rulestats subscriber	<div>Displays a counter security policy rules used in traffic processing. Each session match increments the counter for the rule.</div> <div><pre>[root@defaultthost admin]# scio subs rulestats s0</pre><table><thead><tr><th></th><th>ape</th><th>ids</th></tr></thead><tbody><tr><td>1</td><td>0</td><td>0</td></tr></tbody></table></div>		ape	ids	1	0	0																																																																																																																																																																																														
	ape	ids																																																																																																																																																																																																			
1	0	0																																																																																																																																																																																																			

Table 154: Command Reference: scio subs (*continued*)

Options	Usage and Examples																																													
service detail <i>subscriber</i>	<p>Displays the active and total session count, by service.</p> <pre>[root@defaulthost admin]# scio subs service detail s0</pre> <p>Service Session Count Table:</p> <table><tr><th>Service</th><th>Active</th><th>Total</th></tr><tr><td>FTP</td><td>86</td><td>86</td></tr><tr><td>RLOGIN</td><td>21</td><td>21</td></tr><tr><td>PORTMAPPER</td><td>100</td><td>100</td></tr><tr><td>HTTP</td><td>730</td><td>730</td></tr><tr><td>SMTP</td><td>38</td><td>38</td></tr><tr><td>POP3</td><td>76</td><td>76</td></tr><tr><td>IMAP</td><td>10</td><td>10</td></tr><tr><td>TELNET</td><td>52</td><td>52</td></tr><tr><td>ICMP</td><td>116</td><td>116</td></tr><tr><td>DNS</td><td>50</td><td>52</td></tr><tr><td>SSH</td><td>1</td><td>1</td></tr><tr><td>SNMP</td><td>11</td><td>11</td></tr><tr><td>DHCP</td><td>17</td><td>17</td></tr><tr><td>TFTP</td><td>21</td><td>21</td></tr></table>	Service	Active	Total	FTP	86	86	RLOGIN	21	21	PORTMAPPER	100	100	HTTP	730	730	SMTP	38	38	POP3	76	76	IMAP	10	10	TELNET	52	52	ICMP	116	116	DNS	50	52	SSH	1	1	SNMP	11	11	DHCP	17	17	TFTP	21	21
Service	Active	Total																																												
FTP	86	86																																												
RLOGIN	21	21																																												
PORTMAPPER	100	100																																												
HTTP	730	730																																												
SMTP	38	38																																												
POP3	76	76																																												
IMAP	10	10																																												
TELNET	52	52																																												
ICMP	116	116																																												
DNS	50	52																																												
SSH	1	1																																												
SNMP	11	11																																												
DHCP	17	17																																												
TFTP	21	21																																												
status <i>subscriber</i>	<p>Provides a summary of status and performance statistics.</p> <pre>[root@defaulthost admin]# scio subs status s0</pre> <p>Status for subs 's0'</p> <pre>up since - Thu Aug 12 17:18:53 2010 Packets/second: 11          peak: 27027 @ Thu Aug 12 17:20:01 2010 KBits/second: 25           peak: 99724 @ Thu Aug 12 17:20:01 2010 Packets received: icmp 30227, tcp 254924, udp 24019, other 0 Current flows: icmp 0, tcp 2, udp 34661, other 0 Current sessions: icmp 0, tcp 1, udp 11893, other 0 Current bypassed flows : 0 Current bypass mode : OFF Latency Statistics (time in micro seconds): Min: 0 Max: 0 Ave: 0 Performance statistics Average packet lifetime: Cycles: 0 Instructions: 0 CPI: 0.00 Cache misses: 0 hits: 0 Current policy: idpengine v0</pre> <p>For IDP8200, a summary is displayed for each IDP engine. To view an aggregate summary for IDP8200 devices, use <b>scio subs aggregatestatus s0</b>.</p>																																													

scio sysconf

**Syntax**    `scio sysconf option`

**Description**    Displays supported protocols, attacks, and contexts.

**Options**    [Table 155 on page 539](#) describes **scio sysconf** options and provides examples of command syntax.

Table 155: Command Reference: scio sysconf

Options	Usage and Examples
all	<div>Displays a complete list of supported protocols, attacks, and contexts.</div> <div><pre>[root@defaulthost admin]# scio sysconf all (sysconf   :model (     :type (NS-IDP-1100C)   )   :version (     :branch (idp51)     :major (5)     :minor (1)     :build (136809)   )   :interfaces (     : (eth2       :nic (eth2)       :vr (vr0)       :subs (s0)       :ipaddr ("n/a")       :netmask ("n/a")       :broadcast ("n/a")       :mac ("00:00:00:00:00:00")       :sniffer (true)       :ha_interface (false)       :external (false)     )   ) )</pre></div> <div>[...]</div>

Table 155: Command Reference: scio sysconf (*continued*)

Options	Usage and Examples
protocols	



Table 155: Command Reference: scio sysconf (continued)

Options	Usage and Examples				
	Displays protocols that can be decoded.				
	[root@default host admin]# scio sysconf protocols				
	Name	Proto	Port	Line Separator	Scope
	----	----	----	-----	-----
	ECHO	TCP	7	CRLF	session
	DISCARD	TCP	9	NONE	session
	CHARGEN	TCP	19	NONE	session
	FTP	TCP	21	CRLF or LF	session
	SSH	TCP	22	NONE	session
	TELNET	TCP	23	NONE	session
	SMTP	TCP	25	CRLF or LF	transaction
	DNS	TCP	53	NONE	transaction
	GOPHER	TCP	70	NONE	session
	FINGER	TCP	79	CRLF or LF	session
	HTTP	TCP	80	CRLF or LF	transaction
	HTTP	TCP	3128	CRLF or LF	transaction
	HTTP	TCP	8000	CRLF or LF	transaction
	HTTP	TCP	8080	CRLF or LF	transaction
	POP3	TCP	110	CRLF or LF	session
	PORTMAPPER	TCP	111	NONE	transaction
	IDENT	TCP	113	CRLF or LF	session
	SMB	TCP	139	NONE	session
	IMAP	TCP	143	CRLF or LF	session
	SMB	TCP	445	NONE	session
	REXEC	TCP	512	NONE	session
	RLOGIN	TCP	513	NONE	session
	RSH	TCP	514	/	session
	LPR	TCP	515	CRLF or LF	session
	RTSP	TCP	554	NONE	session
	NFS	TCP	2049	NONE	transaction
	IRC	TCP	6667	CRLF or LF	session
	YMSG	TCP	5050	NONE	session
	AIM	TCP	5190	NONE	session
	VNC	TCP	5800	NONE	session
	VNC	TCP	5900	NONE	session
	NNTP	TCP	119	CRLF	session
	MSN	TCP	1863	CRLF	session
	GNUTELLA	TCP	6346	NONE	session
	WHOIS	TCP	43	CRLF or LF	session
	LDAP	TCP	389	NONE	transaction
	SSL	TCP	443	NONE	session
	MSRPC	TCP	135	NONE	transaction
	MSSQL	TCP	1433	NONE	session
	MYSQL	TCP	3306	NONE	session
	BGP	TCP	0	NONE	session
	SIP	TCP	5060	NONE	session
	TNS	TCP	1521	NONE	session
	H225SGN	TCP	1720	NONE	session
	IEC104	TCP	2404	NONE	session
	MODBUS	TCP	502	NONE	transaction
	UNSPECIFIED	TCP	0	NONE	session
	ECHO	UDP	7	NONE	session
	DISCARD	UDP	9	NONE	session
	CHARGEN	UDP	19	NONE	session
	DNS	UDP	53	NONE	transaction
	DHCP	UDP	67	NONE	transaction

Table 155: Command Reference: scio sysconf (continued)

Options	Usage and Examples				
	DHCP	UDP	68	NONE	transaction
	TFTP	UDP	69	NONE	session
	PORTMAPPER	UDP	111	NONE	transaction
	SNMP	UDP	161	NONE	transaction
	SNMPTRAP	UDP	162	NONE	session
	IKE	UDP	500	NONE	session
	SYSLOG	UDP	514	NONE	session
	NFS	UDP	2049	NONE	transaction
	NTP	UDP	123	NONE	session
	NBNAME	UDP	137	NONE	session
	NBDS	UDP	138	NONE	session
	RADIUS	UDP	1812	NONE	transaction
	RADIUS	UDP	1813	NONE	transaction
	MSRPC	UDP	135	NONE	transaction
	SQLMON	UDP	1434	NONE	session
	UNSPECIFIED	UDP	0	NONE	session
	SIP	UDP	5060	NONE	session
	H225RAS	UDP	1718	NONE	session
	H225RAS	UDP	1719	NONE	session
	MGCP	UDP	2427	NONE	session
	MGCP	UDP	2727	NONE	session
	IEC104	UDP	2404	NONE	session
	RTP	UDP	0	NONE	transaction
	RTPVIDEO	UDP	0	NONE	transaction
	ICMP	ICMP	N/A	NONE	session
	RUSERS	TCP	RPC/100002	NONE	transaction
	RUSERS	UDP	RPC/100002	NONE	transaction
	NFS	TCP	RPC/100003	NONE	transaction
	NFS	UDP	RPC/100003	NONE	transaction
	NFS	TCP	RPC/100227	NONE	transaction
	NFS	UDP	RPC/100227	NONE	transaction
	PORTMAPPER	TCP	RPC/100000	NONE	transaction
	PORTMAPPER	UDP	RPC/100000	NONE	transaction

ptypes

Displays protocols the kernel can detect.

```
[root@default host admin]# scio sysconf ptypes
```

```
Name      ID
----      --
http      0
ssh       1
msn       2
ymsg      3
vnc       4
gnutella  5
gopher    6
```

Table 155: Command Reference: scio sysconf (continued)

Options	Usage and Examples																																																																																												
attacks	<div>Displays attacks that can be detected.</div> <div><pre>[root@defaulthost admin]# scio sysconf attacks</pre><table><thead><tr><th>Service</th><th>SvcID</th><th>Attack</th><th>AttackID</th></tr><tr><th>-----</th><th>-----</th><th>-----</th><th>-----</th></tr></thead><tbody><tr><td>NONE</td><td>0</td><td>ACCEPT</td><td>0</td></tr><tr><td></td><td></td><td>RULEBASE_DROP</td><td>1</td></tr><tr><td></td><td></td><td>NO_VCIRCUIT</td><td>2</td></tr><tr><td></td><td></td><td>NO_ROUTE</td><td>3</td></tr><tr><td></td><td></td><td>NO_ARP_ENTRY</td><td>4</td></tr><tr><td></td><td></td><td>ARP_PENDING</td><td>5</td></tr><tr><td></td><td></td><td>SHORT_READ</td><td>6</td></tr><tr><td></td><td></td><td>LINE_TOO_LONG</td><td>7</td></tr><tr><td></td><td></td><td>TTL_TIME_EXCEEDED</td><td>8</td></tr><tr><td></td><td></td><td>INVALID_IP_PROTOCOL</td><td>9</td></tr><tr><td></td><td></td><td>INVALID_VERSION</td><td>10</td></tr><tr><td></td><td></td><td>INVALID_CHECKSUM</td><td>11</td></tr><tr><td></td><td></td><td>TCP_SESSIONS_EXCEEDED</td><td>12</td></tr><tr><td></td><td></td><td>UDP_SESSIONS_EXCEEDED</td><td>13</td></tr><tr><td></td><td></td><td>ICMP_SESSIONS_EXCEEDED</td><td>14</td></tr><tr><td></td><td></td><td>IP_SESSIONS_EXCEEDED</td><td>15</td></tr><tr><td></td><td></td><td>SESSION_START</td><td>16</td></tr><tr><td></td><td></td><td>SESSION_END</td><td>17</td></tr><tr><td></td><td></td><td>MEMORY_LIMIT_EXCEEDED</td><td>18</td></tr><tr><td></td><td></td><td>OVERSIZED_TCP_SEGMENT</td><td>19</td></tr><tr><td></td><td></td><td>INVALID_TCP_HEADER_LENGTH</td><td>20</td></tr></tbody></table><div>[...]</div></div>	Service	SvcID	Attack	AttackID	-----	-----	-----	-----	NONE	0	ACCEPT	0			RULEBASE_DROP	1			NO_VCIRCUIT	2			NO_ROUTE	3			NO_ARP_ENTRY	4			ARP_PENDING	5			SHORT_READ	6			LINE_TOO_LONG	7			TTL_TIME_EXCEEDED	8			INVALID_IP_PROTOCOL	9			INVALID_VERSION	10			INVALID_CHECKSUM	11			TCP_SESSIONS_EXCEEDED	12			UDP_SESSIONS_EXCEEDED	13			ICMP_SESSIONS_EXCEEDED	14			IP_SESSIONS_EXCEEDED	15			SESSION_START	16			SESSION_END	17			MEMORY_LIMIT_EXCEEDED	18			OVERSIZED_TCP_SEGMENT	19			INVALID_TCP_HEADER_LENGTH	20
Service	SvcID	Attack	AttackID																																																																																										
-----	-----	-----	-----																																																																																										
NONE	0	ACCEPT	0																																																																																										
		RULEBASE_DROP	1																																																																																										
		NO_VCIRCUIT	2																																																																																										
		NO_ROUTE	3																																																																																										
		NO_ARP_ENTRY	4																																																																																										
		ARP_PENDING	5																																																																																										
		SHORT_READ	6																																																																																										
		LINE_TOO_LONG	7																																																																																										
		TTL_TIME_EXCEEDED	8																																																																																										
		INVALID_IP_PROTOCOL	9																																																																																										
		INVALID_VERSION	10																																																																																										
		INVALID_CHECKSUM	11																																																																																										
		TCP_SESSIONS_EXCEEDED	12																																																																																										
		UDP_SESSIONS_EXCEEDED	13																																																																																										
		ICMP_SESSIONS_EXCEEDED	14																																																																																										
		IP_SESSIONS_EXCEEDED	15																																																																																										
		SESSION_START	16																																																																																										
		SESSION_END	17																																																																																										
		MEMORY_LIMIT_EXCEEDED	18																																																																																										
		OVERSIZED_TCP_SEGMENT	19																																																																																										
		INVALID_TCP_HEADER_LENGTH	20																																																																																										
contexts	<div>Displays contexts that can be isolated in attack searches.</div> <div><pre>[root@defaulthost admin]# scio sysconf contexts</pre><table><thead><tr><th>Service</th><th>Context</th><th>OffID</th><th>Direction</th></tr><tr><th>-----</th><th>-----</th><th>-----</th><th>-----</th></tr></thead><tbody><tr><td>NONE</td><td>stream</td><td>0</td><td>ANY</td></tr><tr><td>NONE</td><td>normalized-stream</td><td>1</td><td>ANY</td></tr><tr><td>NONE</td><td>normalized-stream256</td><td>2</td><td>ANY</td></tr><tr><td>NONE</td><td>normalized-stream1k</td><td>3</td><td>ANY</td></tr><tr><td>NONE</td><td>normalized-stream8k</td><td>4</td><td>ANY</td></tr><tr><td>NONE</td><td>stream256</td><td>5</td><td>ANY</td></tr><tr><td>NONE</td><td>stream1k</td><td>6</td><td>ANY</td></tr><tr><td>NONE</td><td>stream8k</td><td>7</td><td>ANY</td></tr><tr><td>NONE</td><td>line</td><td>8</td><td>ANY</td></tr><tr><td>NONE</td><td>first-packet</td><td>9</td><td>ANY</td></tr><tr><td>NONE</td><td>first-data-packet</td><td>10</td><td>ANY</td></tr><tr><td>NONE</td><td>packet</td><td>11</td><td>ANY</td></tr><tr><td>HTTP</td><td>http-url</td><td>12</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-url-parsed</td><td>13</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-url-parsed-param</td><td>14</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-url-parsed-param-parsed</td><td>15</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-get-url-parsed-param-parsed</td><td>16</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-post-url-parsed-param-parsed</td><td>17</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-head-url-parsed-param-parsed</td><td>18</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-param-parsed</td><td>19</td><td>CTS</td></tr><tr><td>HTTP</td><td>http-get-url</td><td>20</td><td>CTS</td></tr></tbody></table><div>[...]</div></div>	Service	Context	OffID	Direction	-----	-----	-----	-----	NONE	stream	0	ANY	NONE	normalized-stream	1	ANY	NONE	normalized-stream256	2	ANY	NONE	normalized-stream1k	3	ANY	NONE	normalized-stream8k	4	ANY	NONE	stream256	5	ANY	NONE	stream1k	6	ANY	NONE	stream8k	7	ANY	NONE	line	8	ANY	NONE	first-packet	9	ANY	NONE	first-data-packet	10	ANY	NONE	packet	11	ANY	HTTP	http-url	12	CTS	HTTP	http-url-parsed	13	CTS	HTTP	http-url-parsed-param	14	CTS	HTTP	http-url-parsed-param-parsed	15	CTS	HTTP	http-get-url-parsed-param-parsed	16	CTS	HTTP	http-post-url-parsed-param-parsed	17	CTS	HTTP	http-head-url-parsed-param-parsed	18	CTS	HTTP	http-param-parsed	19	CTS	HTTP	http-get-url	20	CTS
Service	Context	OffID	Direction																																																																																										
-----	-----	-----	-----																																																																																										
NONE	stream	0	ANY																																																																																										
NONE	normalized-stream	1	ANY																																																																																										
NONE	normalized-stream256	2	ANY																																																																																										
NONE	normalized-stream1k	3	ANY																																																																																										
NONE	normalized-stream8k	4	ANY																																																																																										
NONE	stream256	5	ANY																																																																																										
NONE	stream1k	6	ANY																																																																																										
NONE	stream8k	7	ANY																																																																																										
NONE	line	8	ANY																																																																																										
NONE	first-packet	9	ANY																																																																																										
NONE	first-data-packet	10	ANY																																																																																										
NONE	packet	11	ANY																																																																																										
HTTP	http-url	12	CTS																																																																																										
HTTP	http-url-parsed	13	CTS																																																																																										
HTTP	http-url-parsed-param	14	CTS																																																																																										
HTTP	http-url-parsed-param-parsed	15	CTS																																																																																										
HTTP	http-get-url-parsed-param-parsed	16	CTS																																																																																										
HTTP	http-post-url-parsed-param-parsed	17	CTS																																																																																										
HTTP	http-head-url-parsed-param-parsed	18	CTS																																																																																										
HTTP	http-param-parsed	19	CTS																																																																																										
HTTP	http-get-url	20	CTS																																																																																										

## scio user

**Syntax** `scio user option argument`

**Description** Displays the status of communication with a Juniper Networks IC Series Unified Access Control (UAC) appliance to support the user role-based policy feature and coordinated threat control.

**Options** [Table 156 on page 544](#) describes **scio user** options and arguments and provides examples of command syntax.

**Table 156: Command Reference: scio ca**

Options	Usage and Examples
status	<p>Shows the status of the connection between the IDP Series appliance and IC Series appliance.</p> <pre>[root@default host admin]# scio user status</pre> <pre>IDP-IC Connectivity is.....[      Up  ] User Session Table Lookup.....[ Enabled ]</pre>
logs throttle {show   set value}	<p>Shows or sets the value for log throttling. Log throttling limits the number of logs per second the IDP Series device sends to the IC Series appliance.</p> <pre>[root@default host admin]# scio user logs throttle show</pre> <pre>5 Log(s)/Second.</pre> <pre>[root@default host admin]# scio user logs throttle set 10</pre> <pre>IC-Log Throttle limit set to '10'.</pre>
list [-u username   -i IP address]	<p>Shows the list of users and IP addresses in the user session table.</p> <pre>[root@default host ~]# scio user list</pre> <pre>1. IP[      8.0.0.1] USER[test] ROLES(1)[QA] 2. IP[      8.0.0.2] USER[test] ROLES(1)[QA1] 3. IP[      5.0.0.1] USER[test] ROLES(1)[QA3] 4. IP[      9.0.0.1] USER[test] ROLES(1)[QA2]</pre> <pre>===== Total Matches Found (4) =====</pre>

Table 156: Command Reference: scio ca (*continued*)

Options	Usage and Examples																																																																																				
counters <i>cmd type</i>	<p>Displays, resets, enables, or disables counters for the user session table. You use these counters for diagnostic purposes only—to verify the statistics increment when the IC Series appliance updates the user session table.</p> <p><i>cmd</i> is {list   reset   enable   disable}. Specifies a counter operation.</p> <p><i>type</i> is {all   add   delete   lookup}. Specifies the counter.</p> <p>The success and failure counters refer to the success and failure for user add, user delete, and user lookup operations.</p> <pre>[root@default host admin]# scio user counters list all</pre> <table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td> </td><td>SUCCESS</td><td>  FAILURE</td></tr><tr><td></td><td> </td><td></td><td> </td></tr><tr><td>Add</td><td> </td><td>0</td><td>  0</td></tr><tr><td></td><td> </td><td></td><td> </td></tr><tr><td>Delete</td><td> </td><td>0</td><td>  0</td></tr><tr><td></td><td> </td><td></td><td> </td></tr><tr><td>Lookup</td><td> </td><td>0</td><td>  0</td></tr><tr><td></td><td> </td><td></td><td> </td></tr></table> <pre>[root@default host admin]# scio user counters list lookup</pre> <table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td> </td><td>SUCCESS</td><td>  FAILURE</td></tr><tr><td></td><td> </td><td></td><td> </td></tr><tr><td>Lookup</td><td> </td><td>0</td><td>  0</td></tr><tr><td></td><td> </td><td></td><td> </td></tr></table> <pre>[root@default host admin]# scio user counters list add delete</pre> <table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td> </td><td>SUCCESS</td><td>  FAILURE</td></tr><tr><td></td><td> </td><td></td><td> </td></tr><tr><td>Add</td><td> </td><td>0</td><td>  0</td></tr><tr><td></td><td> </td><td></td><td> </td></tr><tr><td>Delete</td><td> </td><td>0</td><td>  0</td></tr><tr><td></td><td> </td><td></td><td> </td></tr></table>							SUCCESS	FAILURE					Add		0	0					Delete		0	0					Lookup		0	0											SUCCESS	FAILURE					Lookup		0	0											SUCCESS	FAILURE					Add		0	0					Delete		0	0				
		SUCCESS	FAILURE																																																																																		
Add		0	0																																																																																		
Delete		0	0																																																																																		
Lookup		0	0																																																																																		
		SUCCESS	FAILURE																																																																																		
Lookup		0	0																																																																																		
		SUCCESS	FAILURE																																																																																		
Add		0	0																																																																																		
Delete		0	0																																																																																		

## scio var

---

**Syntax** `scio var {-s subscriber | -v virtual router} [-f file][varname]`

**Description** Displays variables related to a subscriber or a virtual router.

**Options** [Table 157 on page 547](#) describes options and arguments to the **scio var** command and provides examples of command syntax.

Table 157: Command Reference: scio var

Options	Usage and Example
<code>-s subscriber [variable]</code>	<p>Lists variables related to subscribers. To display a summary list of all variables, do not specify a variable name.</p> <pre>[root@defaulthost admin]# scio var -s s0 Kernel variables for 'subscriber' 's0'  sc_icmp_session_table: Timed hash table sc_fragment_table: Timed hash table sc_gate_table: Timed hash table sc_tcp_session_table: Timed hash table sc_bypass_flow_table: Timed hash table sc_udp_session_table: Timed hash table sc_rpc_program_table: Timed hash table sc_icmp_flow_table: Timed hash table sc_session_table: Timed hash table sc_ip_session_table: Timed hash table sc_ipaction_table: Timed hash table sc_ids_cache: Hash table sc_bypass_counts: variable sc_rpc_xid_table: Timed hash table sc_attack_table: Hash table sc_ape_flow_table: Timed hash table sc_tcp_flow_table: Timed hash table sc_ip_flow_table: Timed hash table sc_udp_flow_table: Timed hash table</pre> <p>To display details of a particular variable, specify the variable name as an argument. The following example shows the UDP flow table:</p> <pre>[root@defaulthost ~]# scio var -s s0 sc_udp_flow_table sc_udp_flow_table:   Source IP   Port   Destination IP   Port   Application   FSt   Dir  Xtra info  VLAN   Timeout   [8.0.0.1   62091] [8.0.0.101   53]   DNS   R   CTS - 0 59/60 [8.0.0.101   53] [8.0.0.1   62091]   DNS   A   STC - 0 59/60 [8.0.0.1   58007] [8.0.0.101   69]   TFTP   R   CTS - 0 59/60 [8.0.0.101   69] [8.0.0.1   58007]   TFTP   A   STC - 0 59/60 [8.0.0.1   3812] [8.0.0.101   111]   PORTMAPPER   R   CTS - 0 59/60 [8.0.0.101   111] [8.0.0.1   3812]   PORTMAPPER   A   STC - 0 59/60 [88.143.88.25   59092] [8.0.0.101   0]   RTP   R   CTS - 0 59/60 [8.0.0.101   0] [88.143.88.25   59092]   RTP   A   STC - 0 59/60</pre>

Table 157: Command Reference: scio var (continued)

Options	Usage and Example
-v <i>virtual-router</i> [ <i>variable</i> ]	<p>Lists variables related to virtual routers. To display a summary list of all variables, do not specify a variable name.</p> <pre>[root@defaultthost admin]# scio var -v vr1</pre> <p>Kernel variables for 'virtual router' 'vr1'</p> <pre>sc_arp_table: Timed hash table sc_mac_table: Timed hash table</pre> <p>To display details of a particular variable, specify the variable name as an argument.</p> <pre>[root@defaultthost admin]# scio var -v vr1 sc_arp_table</pre> <pre>sc_arp_table: IP Address      MAC Address      Interface      (On Behalf Of) Timeout</pre>
[-f <i>file</i> ]	<p>Writes the output to the specified filename.</p> <pre>[root@defaultthost admin]# scio var -ssO -f /tmp/udp_flow_table.txt sc_udp_flow_table</pre> <pre>[root@defaultthost admin]#</pre> <pre>[root@defaultthost admin]# more /tmp/udp_flow_table.txt</pre> <pre>sc_udp_flow_table:   Source IP   Port   Destination IP   Port   Application   FSt   Dir  Xtra info  VLAN   Timeout   [8.0.0.1    62091] [8.0.0.101    53] DNS      R   CTS -      0    59/60 [8.0.0.101  53] [8.0.0.1    62091] DNS      A   STC -      0    59/60 [8.0.0.1    58007] [8.0.0.101    69] TFTP     R   CTS -      0    59/60 [8.0.0.101  69] [8.0.0.1    58007] TFTP     A   STC -      0    59/60 [8.0.0.1    3812] [8.0.0.101    111] PORTMAPPER R   CTS -      0    59/60 [8.0.0.101  111] [8.0.0.1    3812] PORTMAPPER A   STC -      0    59/60 [88.143.88.25 59092] [8.0.0.101    0] RTP      R   CTS -      0    59/60 [8.0.0.101  0] [88.143.88.25 59092] RTP      A   STC -      0    59/60</pre>



## scio vc

**Syntax** *scio vc option arguments*

**Description** Enables you to create and manage virtual circuits.

**Options** [Table 158 on page 549](#) describes **scio vc** options and arguments and provides examples of command syntax.

**Table 158: Command Reference: scio vc**

Options	Usage and Examples																												
list	<p>Lists virtual circuits.</p> <pre>[root@defaulthost admin]# scio vc list</pre> <p>Defined Virtual Circuits:</p> <table><tr><th>V-Circuit HA</th><th>NIC</th><th>V-Router</th><th>Subscriber</th><th>IP Address</th><th>Network Mask</th><th>Sniff</th></tr><tr><td>-----</td><td>----</td><td>-----</td><td>-----</td><td>-----</td><td>-----</td><td>-----</td></tr><tr><td>eth2 no</td><td>eth2</td><td>vr1</td><td>s0</td><td>n/a</td><td>n/a</td><td>yes</td></tr><tr><td>eth3 no</td><td>eth3</td><td>vr1</td><td>s0</td><td>n/a</td><td>n/a</td><td>yes</td></tr></table> <pre>[root@defaulthost admin]#</pre>	V-Circuit HA	NIC	V-Router	Subscriber	IP Address	Network Mask	Sniff	-----	----	-----	-----	-----	-----	-----	eth2 no	eth2	vr1	s0	n/a	n/a	yes	eth3 no	eth3	vr1	s0	n/a	n/a	yes
V-Circuit HA	NIC	V-Router	Subscriber	IP Address	Network Mask	Sniff																							
-----	----	-----	-----	-----	-----	-----																							
eth2 no	eth2	vr1	s0	n/a	n/a	yes																							
eth3 no	eth3	vr1	s0	n/a	n/a	yes																							
external <i>virtual-circuit</i> [set unset]	<p>Sets or unsets the external bit for the virtual circuit.</p> <pre>[root@defaulthost admin]# scio vc external eth2 set</pre> <pre>[root@defaulthost admin]#</pre>																												
sniff <i>virtual-circuit</i> [enable   disable]	<p>Sets or unsets sniffer mode for the specified virtual circuit.</p> <pre>[root@defaulthost admin]# scio vc sniff eth2 enable</pre> <pre>[root@defaulthost admin]#</pre>																												
define <i>virtual-circuit</i> <i>vc-type</i>	<p>Creates a new virtual circuit with the specified name and type.</p> <pre>[root@defaulthost admin]# scio vc define eth4 sniff</pre> <pre>[root@defaulthost admin]#</pre>																												
undef <i>virtual-circuit</i>	<p>Deletes the specified virtual circuit.</p> <pre>[root@defaulthost admin]# scio vc undef eth4</pre> <pre>[root@defaulthost admin]#</pre>																												

## scio version

---

**Syntax**    **scio version**

**Description**    Displays the version of the **scio** utility and IDP OS. You might need to display the precise version numbers in cases where you troubleshoot issues with Juniper Networks Technical Assistance Center (J-TAC).

The following example shows output of the **scio version** command:

```
[root@defaulthost ~]# scio -c 0 version
scio 5.1.136718
kernel 5.1.136718
```

**Options**    None

## scio vr

**Syntax** `scio vr option arguments`

**Description** Enables you to create and manage virtual routers.

**Options** [Table 159 on page 551](#) describes **scio vr** options and arguments and provides example command syntax.

**Table 159: Command Reference: scio vr**

Options	Usage and Examples
list	<p>Lists virtual routers.</p> <pre>[root@defaulthost admin]# scio vr list Attached Virtual Routers:  V-Router  V-Circuit  NIC -----  - vr0        eth3       eth3            eth2       eth2  vr1        eth5       eth5            eth4       eth4  vr2        eth7       eth7            eth6       eth6  vr3        eth9       eth9            eth8       eth8  vr4        eth11      eth11            eth10      eth10</pre>
define <i>virtual-router</i>	<p>Creates a virtual router.</p> <pre>[root@defaulthost admin]# scio vr define vr2</pre>
undef <i>virtual-circuit</i>	<p>Deletes a virtual router.</p> <pre>[root@defaulthost admin]# scio vr undef vr2  [root@defaulthost admin]# scio vr list Attached Virtual Routers: V-Router  V-Circuit  NIC -----  - vr1        eth3       eth3            eth2       eth2 [root@defaulthost admin]#</pre>
attach <i>virtual-router</i> <i>virtual-circuit</i>	<p>Associates a virtual circuit with a virtual router..</p> <pre>[root@defaulthost admin]# scio vr attach vr1 eth4 [root@defaulthost admin]#</pre>

Table 159: Command Reference: scio vr (*continued*)

Options	Usage and Examples
mode <i>virtual-router</i> {sniffer transparent}	Sets the deployment mode.  [root@defaulthost admin]# <b>scio vr mode vr1 transparent</b> [root@defaulthost admin]#
listmac <i>virtual-router</i>	Displays multicast addresses for a virtual router.  [root@defaulthost admin]# <b>scio vr listmac vr1</b> Mac addresses added to Virtual Router 'vr1' MAC ADDRESS                      V-Circuit ----- [root@defaulthost admin]#
addmac <i>virtual-router</i> mac-addr <i>virtual-circuit</i>	Assigns a multicast address to a virtual router.  [root@defaulthost admin]# <b>scio vr addmac vr1 00-0C-F1-56-98-AD eth4</b> [root@defaulthost admin]#
delmac <i>virtual-router</i> mac-addr <i>virtual-circuit</i>	Deletes the multicast address assigned to a virtual circuit.  [root@defaulthost admin]# <b>scio vr delmac vr1 00-0C-F1-56-98-AD eth4</b> [root@defaulthost admin]#
addstaticmac <i>virtual-router</i> mac-addr	Adds a MAC address to a the virtual router MAC table.  [root@defaulthost admin]# <b>scio vr addstaticmac vr1 00-0C-F1-56-98-AD</b> [root@defaulthost admin]#
delstaticmac <i>virtual-router</i> mac-addr	Deletes a MAC address from the virtual router MAC table.  [root@defaulthost admin]# <b>scio vr delstaticmac vr1 00-0C-F1-56-98-AD</b> [root@defaulthost admin]#
addarp <i>virtual-router</i> ip-addr mac-addr <i>virtual-circuit</i>	Adds an ARP entry to the virtual router ARP table.  [root@defaulthost admin]# <b>scio vr addarp vr1 10.1.1.1 00-0C-F1-56-98-AD eth4</b> [root@defaulthost admin]#
delarp <i>virtual-router</i> ip-addr	Releases the association between a virtual circuit and a virtual router.  [root@defaulthost admin]# <b>scio vr delarp vr1 10.1.1.1</b> [root@defaulthost admin]#
showspan <i>virtual-router</i>	Shows spanning tree protocol (STP) settings.  [root@defaulthost admin]# <b>scio vr showspan vr1</b> [root@defaulthost admin]#
reset <i>virtual-router</i> [ <i>virtual-circuit</i> {enable   disable}]	Resets the configuration.  [root@defaulthost admin]# <b>scio vr reset vr1 eth4 enable</b> [root@defaulthost admin]#

# IDP MIB Object ID Reference

- [IDP Series MIB Object ID Reference on page 553](#)

## IDP Series MIB Object ID Reference

Device MIB files are located in the `/usr/share/snmp/mibs/` directory. The **JUNIPER-IDP-MIB.txt** file contains the IDP Series MIB definition.

You can use SNMP query tools to retrieve data from the device MIB. SNMP trap receivers and SNMP query tools are widely available, and most network administrators already have a preferred tool. For information about using these tools, see the documentation provided by your vendor.

This reference includes the following tables:

- [Table 160 on page 553](#) shows query results using name-based or number-based display options with the onboard **snmpwalk** utility.
- [Table 161 on page 559](#) describes the MIB objects.
- [Table 162 on page 567](#) describes the SNMP traps.

**Table 160: snmpwalk Results by Name and by Number**

OID (name)	OID (number)
<code>[cmd] # snmpwalk -v2c -c public localhost JUNIPER-IDP-MIB::jnxIdpSensor -O sq</code>	<code>[cmd] # snmpwalk -v2c -c public localhost 1.3.6.1.4.1.2636.3.9 -O n</code>
<code>jnxIdpSensorCpuUsage.0 39</code>	<code>.1.3.6.1.4.1.2636.3.9.1.1.0 = Gauge32: 100</code>
<code>jnxIdpSensorMemUsage.0 29</code>	<code>.1.3.6.1.4.1.2636.3.9.1.2.0 = Gauge32: 29</code>
<code>jnxIdpSensorSessAllocated.0 1</code>	<code>.1.3.6.1.4.1.2636.3.9.1.3.0 = Gauge32: 1</code>
<code>jnxIdpSensorSessMaximum.0 500000</code>	<code>.1.3.6.1.4.1.2636.3.9.1.4.0 = INTEGER: 500000</code>
<code>jnxIdpSensorFreeDiskSpace.0 46716 Megabytes</code>	<code>.1.3.6.1.4.1.2636.3.9.1.5.0 = Gauge32: 46719</code>
<code>jnxIdpSensorCpuThreshold.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.6.0 = INTEGER: 0</code>
<code>jnxIdpSensorMemThreshold.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.7.0 = INTEGER: 0</code>
<code>jnxIdpSensorSessThreshold.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.8.0 = INTEGER: 0</code>
<code>jnxIdpSensorDiskSpaceThreshold.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.9.0 = INTEGER: 0</code>
<code>jnxIdpSensorCpuUsageOneMin.0 105</code>	<code>.1.3.6.1.4.1.2636.3.9.1.10.0 = Gauge32: 66</code>
<code>jnxIdpSensorCpuUsageFiveMin.0 83</code>	<code>.1.3.6.1.4.1.2636.3.9.1.11.0 = Gauge32: 56</code>
<code>jnxIdpSensorFiveSecCpuID.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.12.1.1.0 = INTEGER: 0</code>
<code>jnxIdpSensorFiveSecCpuUtilPercent.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.12.1.2.0 = Gauge32: 0</code>
<code>jnxIdpSensorOneMinCpuID.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.13.1.1.0 = INTEGER: 0</code>
<code>jnxIdpSensorOneMinCpuUtilPercent.0 0</code>	<code>.1.3.6.1.4.1.2636.3.9.1.13.1.2.0 = Gauge32: 0</code>

Table 160: snmpwalk Results by Name and by Number (*continued*)

OID (name)	OID (number)
jnxIdpSensorFiveMinCpuID.0 0	.1.3.6.1.4.1.2636.3.9.1.14.1.1.0 = INTEGER: 0
jnxIdpSensorFiveMinCpuUtilPercent.0 0	.1.3.6.1.4.1.2636.3.9.1.14.1.2.0 = Gauge32: 0
jnxIdpSensorSessnCreateRateFiveSec.0 0	.1.3.6.1.4.1.2636.3.9.1.15.0 = INTEGER: 0
jnxIdpSensorTCPSessions.0 0	.1.3.6.1.4.1.2636.3.9.1.16.0 = INTEGER: 0
jnxIdpSensorUDPSessions.0 0	.1.3.6.1.4.1.2636.3.9.1.17.0 = INTEGER: 0
jnxIdpSensorICMPSessions.0 1	.1.3.6.1.4.1.2636.3.9.1.18.0 = INTEGER: 1
jnxIdpSensorOtherSessions.0 0	.1.3.6.1.4.1.2636.3.9.1.19.0 = INTEGER: 0
jnxIdpSensorFreePktBuffersFiveSec.0 486930	.1.3.6.1.4.1.2636.3.9.1.20.0 = INTEGER: 488586
jnxIdpSensorFreePktBuffersOneMin.0 486930	.1.3.6.1.4.1.2636.3.9.1.21.0 = INTEGER: 488586
jnxIdpSensorPacketsPerSec.0 2	.1.3.6.1.4.1.2636.3.9.1.22.0 = INTEGER: 2
jnxIdpSensorBytesPerSec.0 1	.1.3.6.1.4.1.2636.3.9.1.23.0 = INTEGER: 1
jnxIdpSensorIPv4PktsPerSec.0 2	.1.3.6.1.4.1.2636.3.9.1.24.0 = INTEGER: 2
jnxIdpSensorNonIPv4PktsPerSec.0 0	.1.3.6.1.4.1.2636.3.9.1.25.0 = INTEGER: 0
jnxIdpSensorTCPPktsPerSec.0 0	.1.3.6.1.4.1.2636.3.9.1.26.0 = INTEGER: 0
jnxIdpSensorUDPPktsPerSec.0 0	.1.3.6.1.4.1.2636.3.9.1.27.0 = INTEGER: 0
jnxIdpSensorICMPPktsPerSec.0 2	.1.3.6.1.4.1.2636.3.9.1.28.0 = INTEGER: 2
jnxIdpSensorOtherPktsPerSec.0 0	.1.3.6.1.4.1.2636.3.9.1.29.0 = INTEGER: 0
jnxIdpSensorPktsProcessed.0 204	.1.3.6.1.4.1.2636.3.9.1.31.0 = INTEGER: 204
jnxIdpSensorBytesProcessed.0 8763	.1.3.6.1.4.1.2636.3.9.1.32.0 = INTEGER: 8763
jnxIdpSensorTCPPktsProcessed.0 70	.1.3.6.1.4.1.2636.3.9.1.33.0 = INTEGER: 70
jnxIdpSensorUDPPktsProcessed.0 112	.1.3.6.1.4.1.2636.3.9.1.34.0 = INTEGER: 112
jnxIdpSensorICMPPktsProcessed.0 22	.1.3.6.1.4.1.2636.3.9.1.35.0 = INTEGER: 22
jnxIdpSensorOtherPktsProcessed.0 0	.1.3.6.1.4.1.2636.3.9.1.36.0 = INTEGER: 0
jnxIdpSensorFragmentsRxd.0 0	.1.3.6.1.4.1.2636.3.9.1.38.0 = INTEGER: 0
jnxIdpSensorFragmentsReassembled.0 0	.1.3.6.1.4.1.2636.3.9.1.39.0 = INTEGER: 0
jnxIdpSensorFragmentsDropped.0 0	.1.3.6.1.4.1.2636.3.9.1.40.0 = INTEGER: 0
jnxIdpSensorPktsDroppedToRule.0 0	.1.3.6.1.4.1.2636.3.9.1.41.0 = INTEGER: 0
jnxIdpSensorPktsDroppedToChksum.0 0	.1.3.6.1.4.1.2636.3.9.1.42.0 = INTEGER: 0
jnxIdpSensorPktsDroppedToAnomaly.0 0	.1.3.6.1.4.1.2636.3.9.1.43.0 = INTEGER: 0
jnxIdpSensorPktsDroppedToMisc.0 0	.1.3.6.1.4.1.2636.3.9.1.44.0 = INTEGER: 0
jnxIdpSensorPktsDroppedToNonRule.0 0	.1.3.6.1.4.1.2636.3.9.1.45.0 = INTEGER: 0
jnxIdpSensorTotalAlerts.0 0	.1.3.6.1.4.1.2636.3.9.1.46.0 = INTEGER: 0
jnxIdpSensorTotalLogs.0 58	.1.3.6.1.4.1.2636.3.9.1.47.0 = INTEGER: 52
jnxIdpSensorLogsPerSec.0 0	.1.3.6.1.4.1.2636.3.9.1.48.0 = INTEGER: 0
jnxIdpSensorIFTable1Index.1 1	.1.3.6.1.4.1.2636.3.9.1.49.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable1Index.2 2	.1.3.6.1.4.1.2636.3.9.1.49.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable1Index.3 3	.1.3.6.1.4.1.2636.3.9.1.49.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable1Index.4 4	.1.3.6.1.4.1.2636.3.9.1.49.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable1Index.5 5	.1.3.6.1.4.1.2636.3.9.1.49.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable1Index.6 6	.1.3.6.1.4.1.2636.3.9.1.49.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable1Index.7 7	.1.3.6.1.4.1.2636.3.9.1.49.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable1Index.8 8	.1.3.6.1.4.1.2636.3.9.1.49.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable1Index.9 9	.1.3.6.1.4.1.2636.3.9.1.49.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable1Index.10 10	.1.3.6.1.4.1.2636.3.9.1.49.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable1Index.11 11	.1.3.6.1.4.1.2636.3.9.1.49.1.1.11 = INTEGER: 11
jnxIdpSensorIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.1 = STRING: "eth1"
jnxIdpSensorIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.2 = STRING: "eth7"
jnxIdpSensorIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.3 = STRING: "eth6"
jnxIdpSensorIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.4 = STRING: "eth9"
jnxIdpSensorIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.5 = STRING: "eth8"
jnxIdpSensorIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.6 = STRING: "eth11"
jnxIdpSensorIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.7 = STRING: "eth10"
jnxIdpSensorIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.8 = STRING: "eth3"
jnxIdpSensorIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.9 = STRING: "eth2"
jnxIdpSensorIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.10 = STRING: "eth5"
jnxIdpSensorIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.49.1.2.11 = STRING: "eth4"
jnxIdpSensorNoOfPkts.1 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.1 = INTEGER: 0

Table 160: snmpwalk Results by Name and by Number (*continued*)

OID (name)	OID (number)
jnxIdpSensorNoOfPkts.2 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.2 = INTEGER: 0
jnxIdpSensorNoOfPkts.3 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.3 = INTEGER: 0
jnxIdpSensorNoOfPkts.4 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.4 = INTEGER: 0
jnxIdpSensorNoOfPkts.5 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.5 = INTEGER: 0
jnxIdpSensorNoOfPkts.6 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.6 = INTEGER: 0
jnxIdpSensorNoOfPkts.7 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.7 = INTEGER: 0
jnxIdpSensorNoOfPkts.8 5327	.1.3.6.1.4.1.2636.3.9.1.49.1.3.8 = INTEGER: 5195
jnxIdpSensorNoOfPkts.9 168818	.1.3.6.1.4.1.2636.3.9.1.49.1.3.9 = INTEGER: 168686
jnxIdpSensorNoOfPkts.10 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.10 = INTEGER: 0
jnxIdpSensorNoOfPkts.11 0	.1.3.6.1.4.1.2636.3.9.1.49.1.3.11 = INTEGER: 0
jnxIdpSensorIFTable2Index.1 1	.1.3.6.1.4.1.2636.3.9.1.50.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable2Index.2 2	.1.3.6.1.4.1.2636.3.9.1.50.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable2Index.3 3	.1.3.6.1.4.1.2636.3.9.1.50.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable2Index.4 4	.1.3.6.1.4.1.2636.3.9.1.50.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable2Index.5 5	.1.3.6.1.4.1.2636.3.9.1.50.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable2Index.6 6	.1.3.6.1.4.1.2636.3.9.1.50.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable2Index.7 7	.1.3.6.1.4.1.2636.3.9.1.50.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable2Index.8 8	.1.3.6.1.4.1.2636.3.9.1.50.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable2Index.9 9	.1.3.6.1.4.1.2636.3.9.1.50.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable2Index.10 10	.1.3.6.1.4.1.2636.3.9.1.50.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable2Index.11 11	.1.3.6.1.4.1.2636.3.9.1.50.1.1.11 = INTEGER: 11
jnxIdpSensorPktsRxRateIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.1 = STRING: "eth1"
jnxIdpSensorPktsRxRateIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.2 = STRING: "eth7"
jnxIdpSensorPktsRxRateIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.3 = STRING: "eth6"
jnxIdpSensorPktsRxRateIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.4 = STRING: "eth9"
jnxIdpSensorPktsRxRateIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.5 = STRING: "eth8"
jnxIdpSensorPktsRxRateIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.6 = STRING: "eth11"
jnxIdpSensorPktsRxRateIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.7 = STRING: "eth10"
jnxIdpSensorPktsRxRateIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.8 = STRING: "eth3"
jnxIdpSensorPktsRxRateIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.9 = STRING: "eth2"
jnxIdpSensorPktsRxRateIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.10 = STRING: "eth5"
jnxIdpSensorPktsRxRateIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.50.1.2.11 = STRING: "eth4"
jnxIdpSensorPktsRxdPerSec.1 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.1 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.2 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.2 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.3 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.3 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.4 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.4 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.5 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.5 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.6 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.6 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.7 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.7 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.8 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.8 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.9 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.9 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.10 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.10 = INTEGER: 0
jnxIdpSensorPktsRxdPerSec.11 0	.1.3.6.1.4.1.2636.3.9.1.50.1.3.11 = INTEGER: 0
jnxIdpSensorIFTable3Index.1 1	.1.3.6.1.4.1.2636.3.9.1.51.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable3Index.2 2	.1.3.6.1.4.1.2636.3.9.1.51.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable3Index.3 3	.1.3.6.1.4.1.2636.3.9.1.51.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable3Index.4 4	.1.3.6.1.4.1.2636.3.9.1.51.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable3Index.5 5	.1.3.6.1.4.1.2636.3.9.1.51.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable3Index.6 6	.1.3.6.1.4.1.2636.3.9.1.51.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable3Index.7 7	.1.3.6.1.4.1.2636.3.9.1.51.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable3Index.8 8	.1.3.6.1.4.1.2636.3.9.1.51.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable3Index.9 9	.1.3.6.1.4.1.2636.3.9.1.51.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable3Index.10 10	.1.3.6.1.4.1.2636.3.9.1.51.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable3Index.11 11	.1.3.6.1.4.1.2636.3.9.1.51.1.1.11 = INTEGER: 11
jnxIdpSensorRxIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.1 = STRING: "eth1"
jnxIdpSensorRxIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.2 = STRING: "eth7"
jnxIdpSensorRxIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.3 = STRING: "eth6"

Table 160: snmpwalk Results by Name and by Number (*continued*)

OID (name)	OID (number)
jnxIdpSensorRxIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.4 = STRING: "eth9"
jnxIdpSensorRxIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.5 = STRING: "eth8"
jnxIdpSensorRxIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.6 = STRING: "eth11"
jnxIdpSensorRxIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.7 = STRING: "eth10"
jnxIdpSensorRxIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.8 = STRING: "eth3"
jnxIdpSensorRxIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.9 = STRING: "eth2"
jnxIdpSensorRxIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.10 = STRING: "eth5"
jnxIdpSensorRxIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.51.1.2.11 = STRING: "eth4"
jnxIdpSensorRxPktsDropCount.1 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.1 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.2 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.2 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.3 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.3 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.4 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.4 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.5 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.5 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.6 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.6 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.7 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.7 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.8 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.8 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.9 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.9 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.10 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.10 = INTEGER: 0
jnxIdpSensorRxPktsDropCount.11 0	.1.3.6.1.4.1.2636.3.9.1.51.1.3.11 = INTEGER: 0
jnxIdpSensorIFTable4Index.1 1	.1.3.6.1.4.1.2636.3.9.1.52.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable4Index.2 2	.1.3.6.1.4.1.2636.3.9.1.52.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable4Index.3 3	.1.3.6.1.4.1.2636.3.9.1.52.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable4Index.4 4	.1.3.6.1.4.1.2636.3.9.1.52.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable4Index.5 5	.1.3.6.1.4.1.2636.3.9.1.52.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable4Index.6 6	.1.3.6.1.4.1.2636.3.9.1.52.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable4Index.7 7	.1.3.6.1.4.1.2636.3.9.1.52.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable4Index.8 8	.1.3.6.1.4.1.2636.3.9.1.52.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable4Index.9 9	.1.3.6.1.4.1.2636.3.9.1.52.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable4Index.10 10	.1.3.6.1.4.1.2636.3.9.1.52.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable4Index.11 11	.1.3.6.1.4.1.2636.3.9.1.52.1.1.11 = INTEGER: 11
jnxIdpSensorRxPktsDropRateIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.1 = STRING: "eth1"
jnxIdpSensorRxPktsDropRateIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.2 = STRING: "eth7"
jnxIdpSensorRxPktsDropRateIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.3 = STRING: "eth6"
jnxIdpSensorRxPktsDropRateIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.4 = STRING: "eth9"
jnxIdpSensorRxPktsDropRateIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.5 = STRING: "eth8"
jnxIdpSensorRxPktsDropRateIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.6 = STRING: "eth11"
jnxIdpSensorRxPktsDropRateIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.7 = STRING: "eth10"
jnxIdpSensorRxPktsDropRateIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.8 = STRING: "eth3"
jnxIdpSensorRxPktsDropRateIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.9 = STRING: "eth2"
jnxIdpSensorRxPktsDropRateIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.10 = STRING: "eth5"
jnxIdpSensorRxPktsDropRateIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.52.1.2.11 = STRING: "eth4"
jnxIdpSensorRxPktsDropRate.1 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.1 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.2 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.2 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.3 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.3 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.4 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.4 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.5 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.5 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.6 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.6 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.7 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.7 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.8 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.8 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.9 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.9 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.10 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.10 = INTEGER: 0
jnxIdpSensorRxPktsDropRate.11 0	.1.3.6.1.4.1.2636.3.9.1.52.1.3.11 = INTEGER: 0
jnxIdpSensorPktsRxdOnAllIntfc.0 174145	.1.3.6.1.4.1.2636.3.9.1.53.0 = INTEGER: 173881
jnxIdpSensorPktsDropOnAllIntfc.0 0	.1.3.6.1.4.1.2636.3.9.1.54.0 = INTEGER: 0
jnxIdpSensorPktsDropRateOnAllIntfc.0 0	.1.3.6.1.4.1.2636.3.9.1.55.0 = INTEGER: 0
jnxIdpSensorPktsDropDueToRxOverflowTable.0 0	.1.3.6.1.4.1.2636.3.9.1.56.0 = INTEGER: 0
jnxIdpSensorIFTable5Index.1 1	.1.3.6.1.4.1.2636.3.9.1.57.1.1.1 = INTEGER: 1



Table 160: snmpwalk Results by Name and by Number (*continued*)

OID (name)	OID (number)
jnxIdpSensorIFTable5Index.2 2	.1.3.6.1.4.1.2636.3.9.1.57.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable5Index.3 3	.1.3.6.1.4.1.2636.3.9.1.57.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable5Index.4 4	.1.3.6.1.4.1.2636.3.9.1.57.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable5Index.5 5	.1.3.6.1.4.1.2636.3.9.1.57.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable5Index.6 6	.1.3.6.1.4.1.2636.3.9.1.57.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable5Index.7 7	.1.3.6.1.4.1.2636.3.9.1.57.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable5Index.8 8	.1.3.6.1.4.1.2636.3.9.1.57.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable5Index.9 9	.1.3.6.1.4.1.2636.3.9.1.57.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable5Index.10 10	.1.3.6.1.4.1.2636.3.9.1.57.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable5Index.11 11	.1.3.6.1.4.1.2636.3.9.1.57.1.1.11 = INTEGER: 11
jnxIdpSensorTxIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.1 = STRING: "eth1"
jnxIdpSensorTxIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.2 = STRING: "eth7"
jnxIdpSensorTxIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.3 = STRING: "eth6"
jnxIdpSensorTxIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.4 = STRING: "eth9"
jnxIdpSensorTxIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.5 = STRING: "eth8"
jnxIdpSensorTxIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.6 = STRING: "eth11"
jnxIdpSensorTxIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.7 = STRING: "eth10"
jnxIdpSensorTxIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.8 = STRING: "eth3"
jnxIdpSensorTxIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.9 = STRING: "eth2"
jnxIdpSensorTxIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.10 = STRING: "eth5"
jnxIdpSensorTxIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.57.1.2.11 = STRING: "eth4"
jnxIdpSensorNoOfPktsTxd.1 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.1 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.2 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.2 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.3 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.3 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.4 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.4 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.5 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.5 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.6 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.6 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.7 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.7 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.8 2907	.1.3.6.1.4.1.2636.3.9.1.57.1.3.8 = INTEGER: 2775
jnxIdpSensorNoOfPktsTxd.9 2912	.1.3.6.1.4.1.2636.3.9.1.57.1.3.9 = INTEGER: 2780
jnxIdpSensorNoOfPktsTxd.10 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.10 = INTEGER: 0
jnxIdpSensorNoOfPktsTxd.11 0	.1.3.6.1.4.1.2636.3.9.1.57.1.3.11 = INTEGER: 0
jnxIdpSensorIFTable8Index.1 1	.1.3.6.1.4.1.2636.3.9.1.58.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable8Index.2 2	.1.3.6.1.4.1.2636.3.9.1.58.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable8Index.3 3	.1.3.6.1.4.1.2636.3.9.1.58.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable8Index.4 4	.1.3.6.1.4.1.2636.3.9.1.58.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable8Index.5 5	.1.3.6.1.4.1.2636.3.9.1.58.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable8Index.6 6	.1.3.6.1.4.1.2636.3.9.1.58.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable8Index.7 7	.1.3.6.1.4.1.2636.3.9.1.58.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable8Index.8 8	.1.3.6.1.4.1.2636.3.9.1.58.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable8Index.9 9	.1.3.6.1.4.1.2636.3.9.1.58.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable8Index.10 10	.1.3.6.1.4.1.2636.3.9.1.58.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable8Index.11 11	.1.3.6.1.4.1.2636.3.9.1.58.1.1.11 = INTEGER: 11
jnxIdpSensorPktsTxRateIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.1 = STRING: "eth1"
jnxIdpSensorPktsTxRateIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.2 = STRING: "eth7"
jnxIdpSensorPktsTxRateIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.3 = STRING: "eth6"
jnxIdpSensorPktsTxRateIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.4 = STRING: "eth9"
jnxIdpSensorPktsTxRateIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.5 = STRING: "eth8"
jnxIdpSensorPktsTxRateIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.6 = STRING: "eth11"
jnxIdpSensorPktsTxRateIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.7 = STRING: "eth10"
jnxIdpSensorPktsTxRateIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.8 = STRING: "eth3"
jnxIdpSensorPktsTxRateIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.9 = STRING: "eth2"
jnxIdpSensorPktsTxRateIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.10 = STRING: "eth5"
jnxIdpSensorPktsTxRateIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.58.1.2.11 = STRING: "eth4"
jnxIdpSensorPktsTxdPerSec.1 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.1 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.2 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.2 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.3 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.3 = INTEGER: 0

Table 160: snmpwalk Results by Name and by Number (*continued*)

OID (name)	OID (number)
jnxIdpSensorPktsTxdPerSec.4 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.4 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.5 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.5 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.6 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.6 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.7 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.7 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.8 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.8 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.9 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.9 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.10 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.10 = INTEGER: 0
jnxIdpSensorPktsTxdPerSec.11 0	.1.3.6.1.4.1.2636.3.9.1.58.1.3.11 = INTEGER: 0
jnxIdpSensorIFTable6Index.1 1	.1.3.6.1.4.1.2636.3.9.1.59.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable6Index.2 2	.1.3.6.1.4.1.2636.3.9.1.59.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable6Index.3 3	.1.3.6.1.4.1.2636.3.9.1.59.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable6Index.4 4	.1.3.6.1.4.1.2636.3.9.1.59.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable6Index.5 5	.1.3.6.1.4.1.2636.3.9.1.59.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable6Index.6 6	.1.3.6.1.4.1.2636.3.9.1.59.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable6Index.7 7	.1.3.6.1.4.1.2636.3.9.1.59.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable6Index.8 8	.1.3.6.1.4.1.2636.3.9.1.59.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable6Index.9 9	.1.3.6.1.4.1.2636.3.9.1.59.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable6Index.10 10	.1.3.6.1.4.1.2636.3.9.1.59.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable6Index.11 11	.1.3.6.1.4.1.2636.3.9.1.59.1.1.11 = INTEGER: 11
jnxIdpSensorTxdIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.1 = STRING: "eth1"
jnxIdpSensorTxdIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.2 = STRING: "eth7"
jnxIdpSensorTxdIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.3 = STRING: "eth6"
jnxIdpSensorTxdIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.4 = STRING: "eth9"
jnxIdpSensorTxdIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.5 = STRING: "eth8"
jnxIdpSensorTxdIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.6 = STRING: "eth11"
jnxIdpSensorTxdIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.7 = STRING: "eth10"
jnxIdpSensorTxdIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.8 = STRING: "eth3"
jnxIdpSensorTxdIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.9 = STRING: "eth2"
jnxIdpSensorTxdIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.10 = STRING: "eth5"
jnxIdpSensorTxdIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.59.1.2.11 = STRING: "eth4"
jnxIdpSensorTxPktsDropped.1 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.1 = INTEGER: 0
jnxIdpSensorTxPktsDropped.2 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.2 = INTEGER: 0
jnxIdpSensorTxPktsDropped.3 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.3 = INTEGER: 0
jnxIdpSensorTxPktsDropped.4 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.4 = INTEGER: 0
jnxIdpSensorTxPktsDropped.5 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.5 = INTEGER: 0
jnxIdpSensorTxPktsDropped.6 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.6 = INTEGER: 0
jnxIdpSensorTxPktsDropped.7 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.7 = INTEGER: 0
jnxIdpSensorTxPktsDropped.8 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.8 = INTEGER: 0
jnxIdpSensorTxPktsDropped.9 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.9 = INTEGER: 0
jnxIdpSensorTxPktsDropped.10 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.10 = INTEGER: 0
jnxIdpSensorTxPktsDropped.11 0	.1.3.6.1.4.1.2636.3.9.1.59.1.3.11 = INTEGER: 0
jnxIdpSensorTxPktsOnAllIntfc.0 5819	.1.3.6.1.4.1.2636.3.9.1.60.0 = INTEGER: 5555
jnxIdpSensorTxPktsDropOnAllIntfc.0 0	.1.3.6.1.4.1.2636.3.9.1.61.0 = INTEGER: 0
jnxIdpSensorIFTable7Index.1 1	.1.3.6.1.4.1.2636.3.9.1.62.1.1.1 = INTEGER: 1
jnxIdpSensorIFTable7Index.2 2	.1.3.6.1.4.1.2636.3.9.1.62.1.1.2 = INTEGER: 2
jnxIdpSensorIFTable7Index.3 3	.1.3.6.1.4.1.2636.3.9.1.62.1.1.3 = INTEGER: 3
jnxIdpSensorIFTable7Index.4 4	.1.3.6.1.4.1.2636.3.9.1.62.1.1.4 = INTEGER: 4
jnxIdpSensorIFTable7Index.5 5	.1.3.6.1.4.1.2636.3.9.1.62.1.1.5 = INTEGER: 5
jnxIdpSensorIFTable7Index.6 6	.1.3.6.1.4.1.2636.3.9.1.62.1.1.6 = INTEGER: 6
jnxIdpSensorIFTable7Index.7 7	.1.3.6.1.4.1.2636.3.9.1.62.1.1.7 = INTEGER: 7
jnxIdpSensorIFTable7Index.8 8	.1.3.6.1.4.1.2636.3.9.1.62.1.1.8 = INTEGER: 8
jnxIdpSensorIFTable7Index.9 9	.1.3.6.1.4.1.2636.3.9.1.62.1.1.9 = INTEGER: 9
jnxIdpSensorIFTable7Index.10 10	.1.3.6.1.4.1.2636.3.9.1.62.1.1.10 = INTEGER: 10
jnxIdpSensorIFTable7Index.11 11	.1.3.6.1.4.1.2636.3.9.1.62.1.1.11 = INTEGER: 11
jnxIdpSensorNICIntfcName.1 "eth1"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.1 = STRING: "eth1"
jnxIdpSensorNICIntfcName.2 "eth7"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.2 = STRING: "eth7"
jnxIdpSensorNICIntfcName.3 "eth6"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.3 = STRING: "eth6"

Table 160: snmpwalk Results by Name and by Number (*continued*)

OID (name)	OID (number)
jnxIdpSensorNICIntfcName.4 "eth9"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.4 = STRING: "eth9"
jnxIdpSensorNICIntfcName.5 "eth8"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.5 = STRING: "eth8"
jnxIdpSensorNICIntfcName.6 "eth11"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.6 = STRING: "eth11"
jnxIdpSensorNICIntfcName.7 "eth10"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.7 = STRING: "eth10"
jnxIdpSensorNICIntfcName.8 "eth3"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.8 = STRING: "eth3"
jnxIdpSensorNICIntfcName.9 "eth2"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.9 = STRING: "eth2"
jnxIdpSensorNICIntfcName.10 "eth5"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.10 = STRING: "eth5"
jnxIdpSensorNICIntfcName.11 "eth4"	.1.3.6.1.4.1.2636.3.9.1.62.1.2.11 = STRING: "eth4"
jnxIdpSensorNICStatus.1 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.1 = STRING: "Down"
jnxIdpSensorNICStatus.2 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.2 = STRING: "Down"
jnxIdpSensorNICStatus.3 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.3 = STRING: "Down"
jnxIdpSensorNICStatus.4 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.4 = STRING: "Down"
jnxIdpSensorNICStatus.5 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.5 = STRING: "Down"
jnxIdpSensorNICStatus.6 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.6 = STRING: "Down"
jnxIdpSensorNICStatus.7 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.7 = STRING: "Down"
jnxIdpSensorNICStatus.8 "Up"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.8 = STRING: "Up"
jnxIdpSensorNICStatus.9 "Up"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.9 = STRING: "Up"
jnxIdpSensorNICStatus.10 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.10 = STRING: "Down"
jnxIdpSensorNICStatus.11 "Down"	.1.3.6.1.4.1.2636.3.9.1.62.1.3.11 = STRING: "Down"
jnxIdpSensorRuleIndex.1 1	.1.3.6.1.4.1.2636.3.9.1.63.1.1.1 = INTEGER: 1
jnxIdpSensorRuleIndex.2 2	.1.3.6.1.4.1.2636.3.9.1.63.1.1.2 = INTEGER: 2
jnxIdpSensorRulebaseName.1 "tsig"	.1.3.6.1.4.1.2636.3.9.1.63.1.2.1 = STRING: "tsig"
jnxIdpSensorRulebaseName.2 "ids"	.1.3.6.1.4.1.2636.3.9.1.63.1.2.2 = STRING: "ids"
jnxIdpSensorRuleID.1 1	.1.3.6.1.4.1.2636.3.9.1.63.1.3.1 = INTEGER: 1
jnxIdpSensorRuleID.2 1	.1.3.6.1.4.1.2636.3.9.1.63.1.3.2 = INTEGER: 1
jnxIdpSensorRuleHits.1 0	.1.3.6.1.4.1.2636.3.9.1.63.1.4.1 = INTEGER: 0
jnxIdpSensorRuleHits.2 0	.1.3.6.1.4.1.2636.3.9.1.63.1.4.2 = INTEGER: 0
jnxIdpSensorTopTenRuleIndex.1 1	.1.3.6.1.4.1.2636.3.9.1.65.1.1.1 = INTEGER: 1
jnxIdpSensorTopTenRuleIndex.2 2	.1.3.6.1.4.1.2636.3.9.1.65.1.1.2 = INTEGER: 2
jnxIdpSensorTopTenRulebaseName.1 "tsig"	.1.3.6.1.4.1.2636.3.9.1.65.1.2.1 = STRING: "tsig"
jnxIdpSensorTopTenRulebaseName.2 "ids"	.1.3.6.1.4.1.2636.3.9.1.65.1.2.2 = STRING: "ids"
jnxIdpSensorTopTenRuleID.1 1	.1.3.6.1.4.1.2636.3.9.1.65.1.3.1 = INTEGER: 1
jnxIdpSensorTopTenRuleID.2 1	.1.3.6.1.4.1.2636.3.9.1.65.1.3.2 = INTEGER: 1
jnxIdpSensorTopTenRuleHits.1 0	.1.3.6.1.4.1.2636.3.9.1.65.1.4.1 = INTEGER: 0
jnxIdpSensorTopTenRuleHits.2 0	.1.3.6.1.4.1.2636.3.9.1.65.1.4.2 = INTEGER: 0

Table 161 on page 559 describes the objects in the JUNIPER-IDP-MIB.txt file.

Table 161: IDP Series MIB Objects

Object Name	Object Identifier	Description
jnxIdpMIB	1.3.6.1.4.1.2636.3.9.0	Structure of Juniper Networks IDP Series MIBs.
jnxIdpSensor	1.3.6.1.4.1.2636.3.9.1.0	Object identifier.
jnxIdpSensorCpuUsage	1.3.6.1.4.1.2636.3.9.1.1.0	Control plane CPU utilization (percent).
jnxIdpSensorMemUsage	1.3.6.1.4.1.2636.3.9.1.2.0	Memory utilization (percent).
jnxIdpSensorSessAllocated	1.3.6.1.4.1.2636.3.9.1.3.0	Sessions currently allocated (count).

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorSessMaximum	1.3.6.1.4.1.2636.3.9.1.4.0	Maximum sessions supported (static number).
jnxIdpSensorFreeDiskSpace	1.3.6.1.4.1.2636.3.9.1.5.0	Available disk space (megabytes).
jnxIdpSensorCpuThreshold	1.3.6.1.4.1.2636.3.9.1.6.0	Threshold of CPU utilization when the device sends an up or down trap (percent).
jnxIdpSensorMemThreshold	1.3.6.1.4.1.2636.3.9.1.7.0	Threshold of memory utilization when the device sends an up or down trap (percent).
jnxIdpSensorSessThreshold	1.3.6.1.4.1.2636.3.9.1.8.0	Threshold of session/capacity when the device sends an up or down trap (percent).
jnxIdpSensorDiskSpaceThreshold	1.3.6.1.4.1.2636.3.9.1.9.0	Threshold of disk space utilization when the device sends an up or down trap (percent).
jnxIdpSensorCpuUsageOneMin	1.3.6.1.4.1.2636.3.9.1.10.0	Average control plane CPU utilization in the last 1 minute (percent).
jnxIdpSensorCpuUsageFiveMin	1.3.6.1.4.1.2636.3.9.1.11.0	Average control plane CPU utilization in the last 5 minutes (percent).
jnxIdpSensorCpuUtilFiveSecTable	1.3.6.1.4.1.2636.3.9.1.12	Table that holds CPU utilization per IDP engine in the last 5 seconds. Each table contains a pair of rows for each IDP engine reported. In each pair, one row is for the CPU ID and the other reports the value.
jnxIdpSensorCpuUtilFiveSecEntry	1.3.6.1.4.1.2636.3.9.1.12.1.0	Table row that holds CPU utilization per IDP engine in the last 5 seconds.
jnxIdpSensorFiveSecCpuID	1.3.6.1.4.1.2636.3.9.1.12.1.1.0	IDP engine CPU ID for the CPU data reported next.
jnxIdpSensorFiveSecCpuUtilPercent	1.3.6.1.4.1.2636.3.9.1.12.1.2.0	Average IDP engine CPU utilization in the last 5 seconds (percent).
jnxIdpSensorCpuUtilOneMinTable	1.3.6.1.4.1.2636.3.9.1.13	Table that holds CPU utilization per IDP engine in the last 1 minute. Each table contains a pair of rows for each IDP engine reported. In each pair of rows, one row is for CPU ID and the other reports the value.
jnxIdpSensorCpuUtilOneMinEntry	1.3.6.1.4.1.2636.3.9.1.13.1.0	Table row that holds CPU utilization per IDP engine in the last 1 minute.
jnxIdpSensorOneMinCpuID	1.3.6.1.4.1.2636.3.9.1.13.1.1.0	The IDP engine CPU ID for the CPU data reported next.

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorOneMinCpuUtilPercent	1.3.6.1.4.1.2636.3.9.1.13.1.2.0	Average IDP engine CPU utilization in the last 1 minute (percent).
jnxIdpSensorCpuUtilFiveMinTable	1.3.6.1.4.1.2636.3.9.1.14	Table that holds CPU utilization per IDP engine in the last 5 minutes. Each table contains a pair of rows for each IDP engine reported. In each pair of rows, one row is for CPU ID and the other reports the value.
jnxIdpSensorCpuUtilFiveMinEntry	1.3.6.1.4.1.2636.3.9.1.14.1.0	Table row that holds CPU utilization per IDP engine in the last 5 minutes.
jnxIdpSensorFiveMinCpuID	1.3.6.1.4.1.2636.3.9.1.14.1.1.0	IDP engine CPU ID for the CPU data reported next.
jnxIdpSensorFiveMinCpuUtilPercent	1.3.6.1.4.1.2636.3.9.1.14.1.2.0	Average IDP engine CPU utilization in the last 5 minutes (percent).
jnxIdpSensorSessnCreateRateFiveSec	1.3.6.1.4.1.2636.3.9.1.15.0	Rate of connections created per second (cps). The value reported is an average per second over the last 5 seconds.
jnxIdpSensorTCPSessions	1.3.6.1.4.1.2636.3.9.1.9.16.0	Active TCP sessions (count).
jnxIdpSensorUDPSessions	1.3.6.1.4.1.2636.3.9.1.9.17.0	Active UDP sessions (count).
jnxIdpSensorICMPSessions	1.3.6.1.4.1.2636.3.9.1.9.18.0	Active ICMP sessions (count).
jnxIdpSensorOtherSessions	1.3.6.1.4.1.2636.3.9.1.9.19.0	Active sessions that are not TCP, UDP, or ICMP (count).
jnxIdpSensorFreePktBuffersFiveSec	1.3.6.1.4.1.2636.3.9.1.9.20.0	Free packet buffers in the last 5 seconds (count).
jnxIdpSensorFreePktBuffersOneMin	1.3.6.1.4.1.2636.3.9.1.9.21.0	Free packet buffers in the last 1 minute (count).
jnxIdpSensorPacketsPerSec	1.3.6.1.4.1.2636.3.9.1.9.22.0	Rate of packets per second (pps) received.
jnxIdpSensorBytesPerSec	1.3.6.1.4.1.2636.3.9.1.9.23.0	Rate of bytes per second (bps) received.
jnxIdpSensorIPv4PktsPerSec	1.3.6.1.4.1.2636.3.9.1.9.24.0	Rate of IPv4 packets per second (pps) received.
jnxIdpSensorNonIPv4PktsPerSec	1.3.6.1.4.1.2636.3.9.1.25.0	Rate of non-IPv4 packets received per second (pps).
jnxIdpSensorTCPPktsPerSec	1.3.6.1.4.1.2636.3.9.1.9.26.0	Rate of TCP packets per second (pps) received.

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorUDPPktsPerSec	1.3.6.1.4.1.2636.3.9.1.9.27.0	Rate of UDP packets per second (pps) received.
jnxIdpSensorICMPktsPerSec jnxIdpSensorTotalLogs	1.3.6.1.4.1.2636.3.9.1.9.28	Rate of ICMP packets per second (pps) received.
jnxIdpSensorOtherPktsPerSec	1.3.6.1.4.1.2636.3.9.1.9.29.0	Rate of packets per second (pps) received for traffic other than TCP, UDP, and ICMP.
jnxIdpSensorPktsProcessed	1.3.6.1.4.1.2636.3.9.1.9.31.0	Total packets processed (count).
jnxIdpSensorBytesProcessed	1.3.6.1.4.1.2636.3.9.1.9.32.0	Total bytes processed (count).
jnxIdpSensorTCPPktsProcessed	1.3.6.1.4.1.2636.3.9.1.9.33.0	Total TCP packets processed (count).
jnxIdpSensorUDPPktsProcessed	1.3.6.1.4.1.2636.3.9.1.9.34.0	Total UDP packets processed (count).
jnxIdpSensorICMPktsProcessed	1.3.6.1.4.1.2636.3.9.1.9.35.0	Total ICMP packets processed (count).
jnxIdpSensorOtherPktsProcessed	1.3.6.1.4.1.2636.3.9.1.9.36.0	Total packets processed for traffic that is not TCP, UDP, or ICMP (count).
jnxIdpSensorFragmentsRxd	1.3.6.1.4.1.2636.3.9.1.9.38.0	Fragments received (count).
jnxIdpSensorFragmentsReassembled	1.3.6.1.4.1.2636.3.9.1.9.39.0	Fragments reassembled (count).
jnxIdpSensorFragmentsDropped	1.3.6.1.4.1.2636.3.9.1.9.40.0	Fragments dropped (count).
jnxIdpSensorPktsDroppedToRule	1.3.6.1.4.1.2636.3.9.1.9.41.0	Packets dropped because of a security policy rule being applied (count).
jnxIdpSensorPktsDroppedToChecksum	1.3.6.1.4.1.2636.3.9.1.9.42.0	Packets dropped because of a checksum error (count).
jnxIdpSensorPktsDroppedToAnomaly	1.3.6.1.4.1.2636.3.9.1.9.43.0	Packets dropped because of a protocol anomaly detected (count).
jnxIdpSensorPktsDroppedToMisc	1.3.6.1.4.1.2636.3.9.1.9.44.0	Packets dropped because of other reasons (count).
jnxIdpSensorPktsDroppedToNonRule	1.3.6.1.4.1.2636.3.9.1.9.45.0	Packets dropped because of a nonpolicy reason (count).
jnxIdpSensorTotalAlerts	1.3.6.1.4.1.2636.3.9.1.9.46.0	Total alerts generated (count).
jnxIdpSensorTotalLogs	1.3.6.1.4.1.2636.3.9.1.9.47.0	Total logs generated (count).
jnxIdpSensorLogsPerSec	1.3.6.1.4.1.2636.3.9.1.9.48.0	Rate of logs per second.

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorPktsRxdPerIntfcTable	1.3.6.1.4.1.2636.3.9.1.49	Table that holds the count of packets received per interface.
jnxIdpSensorPktsRxdPerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.49.1.0	Table row that holds the interface index number, interface name, and count.
jnxIdpSensorIFTable1Index	1.3.6.1.4.1.2636.3.9.1.49.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorIntfcName	1.3.6.1.4.1.2636.3.9.1.49.1.2.11 ...	Name of the interface (string).
jnxIdpSensorNoOfPkts	1.3.6.1.4.1.2636.3.9.1.49.1.3.11 ...	Packets received by the interface (count).
jnxIdpSensorPktsRxRatePerIntfcTable	1.3.6.1.4.1.2636.3.9.1.50	Table that holds the rate of packets received per interface.
jnxIdpSensorPktsRxRatePerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.50.1.0	Table row that holds the interface index number, interface name, and rate.
jnxIdpSensorIFTable2Index	1.3.6.1.4.1.2636.3.9.1.50.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorPktsRxRateIntfcName	1.3.6.1.4.1.2636.3.9.1.50.1.2.11 ...	Name of the interface (string).
jnxIdpSensorPktsRxdPerSec	1.3.6.1.4.1.2636.3.9.1.50.1.3.11 ...	Packets received by the interface (packets per second or pps).
jnxIdpSensorRxPktsDropPerIntfcTable	1.3.6.1.4.1.2636.3.9.1.51	Table that holds the count of packets dropped at the receiving interface.
jnxIdpSensorRxPktsDropPerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.51.1.0	Table row that holds the interface index number, interface name, and rate.
jnxIdpSensorIFTable3Index	1.3.6.1.4.1.2636.3.9.1.51.1.1.11 ...	Index number for the interface (integer).
jnxIdpSensorRxIntfcName	1.3.6.1.4.1.2636.3.9.1.51.1.2.11 ...	Name of the interface (string).
jnxIdpSensorRxPktsDropCount	1.3.6.1.4.1.2636.3.9.1.51.1.3.11 ...	Packets dropped at the receiving interface (count).
jnxIdpSensorRxPktsDropRatePerIntfcTable	1.3.6.1.4.1.2636.3.9.1.52	Table that holds the rate of packets dropped at the receiving interface.
jnxIdpSensorRxPktsDropRatePerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.52.1.0	Table row that holds the interface index number, interface name, and rate.
jnxIdpSensorIFTable4Index	1.3.6.1.4.1.2636.3.9.1.52.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorRxPktsDropRateIntfcName	1.3.6.1.4.1.2636.3.9.1.52.1.2.11 ...	Name of the interface (string).

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorRxPktsDropRate	1.3.6.1.4.1.2636.3.9.1.52.1.3.11 ...	Packets dropped by the interface (packets per second or pps).
jnxIdpSensorPktsRxdOnAllIntfc	1.3.6.1.4.1.2636.3.9.1.53.0	Sum of packets received on all interfaces (count).
jnxIdpSensorPktsDropOnAllIntfc	1.3.6.1.4.1.2636.3.9.1.54.0	Sum of packets dropped on all interfaces (count).
jnxIdpSensorPktsDropRateOnAllIntfc	1.3.6.1.4.1.2636.3.9.1.55.0	Packets dropped by all interfaces (packets per second or pps).
jnxIdpSensorPktsDropDueToRxOverflowTable	1.3.6.1.4.1.2636.3.9.1.56	Table that holds the count of packets dropped because of receiver overflow per interface.
jnxIdpSensorPktsDropDueToRxOverflowEntry	1.3.6.1.4.1.2636.3.9.1.56.1.0	Table row that holds the interface index number, interface name, and count.
jnxIdpSensorIFTable9Index	1.3.6.1.4.1.2636.3.9.1.56.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorRxOverflowIntfcName	1.3.6.1.4.1.2636.3.9.1.56.1.2.11 ...	Name of the interface (string).
jnxIdpSensorNoOfPktsTxd	1.3.6.1.4.1.2636.3.9.1.56.1.3.11 ...	Packets dropped because of receiver overflow (count).
jnxIdpSensorPktsTxdPerIntfcTable	1.3.6.1.4.1.2636.3.9.1.57	Table that holds the count of packets transmitted per interface.
jnxIdpSensorPktsTxdPerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.57.1.0	Table row that holds the interface index number, interface name, and count.
jnxIdpSensorIFTable5Index	1.3.6.1.4.1.2636.3.9.1.57.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorTxIntfcName	1.3.6.1.4.1.2636.3.9.1.57.1.2.11 ...	Name of the interface (string).
jnxIdpSensorNoOfPktsTxd	1.3.6.1.4.1.2636.3.9.1.57.1.3.11 ...	Packets transmitted (count).
jnxIdpSensorPktsTxRatePerIntfcTable	1.3.6.1.4.1.2636.3.9.1.58	Table that holds the rate of packets transmitted per interface.
jnxIdpSensorPktsTxRatePerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.58.1.0	Table row that holds the interface index number, interface name, and rate.
jnxIdpSensorIFTable8Index	1.3.6.1.4.1.2636.3.9.1.58.1.1.11...	An index number for the interface (integer).
jnxIdpSensorPktsTxRateIntfcName	1.3.6.1.4.1.2636.3.9.1.58.1.2.11 ...	Name of the interface (string).



Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorPktsTxdPerSec	1.3.6.1.4.1.2636.3.9.1.58.1.3.11 ...	Packets transmitted (packets per second or pps).
jnxIdpSensorTxPktsDropPerIntfcTable	1.3.6.1.4.1.2636.3.9.1.59	Table that holds the count of packets dropped at the transmit interface.
jnxIdpSensorTxPktsDropPerIntfcEntry	1.3.6.1.4.1.2636.3.9.1.59.1.0	Table row that holds the interface index number, interface name, and count.
jnxIdpSensorIFTable6Index	1.3.6.1.4.1.2636.3.9.1.59.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorTxdIntfcName	1.3.6.1.4.1.2636.3.9.1.59.1.2.11 ...	Name of the interface (string).
jnxIdpSensorTxPktsDropped	1.3.6.1.4.1.2636.3.9.1.59.1.3.11 ...	Packets dropped (count).
jnxIdpSensorTxPktsOnAllIntfc	1.3.6.1.4.1.2636.3.9.1.60.0	Sum of packets transmitted by all transit interfaces (count).
jnxIdpSensorTxPktsDropOnAllIntfc	1.3.6.1.4.1.2636.3.9.1.61.0	Sum of packets dropped by all transit interfaces (count).
jnxIdpSensorNICStatusTable	1.3.6.1.4.1.2636.3.9.1.62	Table that holds the report of interface status.
jnxIdpSensorNICStatusEntry	1.3.6.1.4.1.2636.3.9.1.62.1.0	Table row that holds the interface index number, interface name, and status.
jnxIdpSensorIFTable7Index	1.3.6.1.4.1.2636.3.9.1.62.1.1.11 ...	An index number for the interface (integer).
jnxIdpSensorNICIntfcName	1.3.6.1.4.1.2636.3.9.1.62.1.2.11 ...	Name of the interface (string).
jnxIdpSensorNICStatus	1.3.6.1.4.1.2636.3.9.1.62.1.3.11 ...	Status of the interface (up or down).
jnxIdpSensorRuleStatsTable	1.3.6.1.4.1.2636.3.9.1.63	Table that holds the report of rule statistics.
jnxIdpSensorRuleStatsEntry	1.3.6.1.4.1.2636.3.9.1.63.1.0	Table row that holds the rule index number, rulebase name, rule ID, and matches (count).
jnxIdpSensorRuleIndex	1.3.6.1.4.1.2636.3.9.1.63.1.1.11 ...	An index number for the rule.
jnxIdpSensorRulebaseName	1.3.6.1.4.1.2636.3.9.1.63.1.2.10 ...	Name of the rulebase.
jnxIdpSensorRuleID	1.3.6.1.4.1.2636.3.9.1.63.1.3.10 ...	Rule number.
jnxIdpSensorRuleHits	1.3.6.1.4.1.2636.3.9.1.63.1.4.10 ...	Matches (count).
jnxIdpSensorSignatureStatsTable	1.3.6.1.4.1.2636.3.9.1.64	Table that holds the report of attack object match statistics.

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorSignatureStatsEntry	1.3.6.1.4.1.2636.3.9.1.64.1.0	Table row that holds the following sequence: attack object index number, database ID, name, and matches (count).
jnxIdpSensorSignatureIndex	1.3.6.1.4.1.2636.3.9.1.64.1.1.10	An index number for the attack object.
jnxIdpSensorSignatureID	1.3.6.1.4.1.2636.3.9.1.64.1.2.10	Attack object database ID.
jnxIdpSensorSignatureName	1.3.6.1.4.1.2636.3.9.1.64.1.3.10	Attack object name.
jnxIdpSensorSignatureHits	1.3.6.1.4.1.2636.3.9.1.64.1.4.10	Matches (count).
jnxIdpSensorTopTenRuleStatsTable	1.3.6.1.4.1.2636.3.9.1.65	The table listing the top ten matching rules.
jnxIdpSensorTopTenRuleStatsEntry	1.3.6.1.4.1.2636.3.9.1.65.1.0	Table row that holds the following sequence: rule index number, rulebase name, rule ID, and matches (count).
jnxIdpSensorTopTenRuleIndex	1.3.6.1.4.1.2636.3.9.1.65.1.1.10	An index number for the rule.
jnxIdpSensorTopTenRulebaseName	1.3.6.1.4.1.2636.3.9.1.65.1.2.10 ...	Name of the rulebase.
jnxIdpSensorTopTenRuleID	1.3.6.1.4.1.2636.3.9.1.65.1.3.10 ...	Rule number.
jnxIdpSensorTopTenSignatureStatsTable	1.3.6.1.4.1.2636.3.9.1.65.1.4.10 ...	Matches (count).
jnxIdpSensorTopTenSignatureStatsTable	1.3.6.1.4.1.2636.3.9.1.66	The table listing the top ten matching attack objects.
jnxIdpSensorTopTenSignatureStatsEntry	1.3.6.1.4.1.2636.3.9.1.66.1.0	Table row that holds the following sequence: report index number, attack object database ID, attack object name, and matches (count).
jnxIdpSensorTopTenSignatureIndex	1.3.6.1.4.1.2636.3.9.1.66.1.1.10 ...	An index number for the attack object.
jnxIdpSensorTopTenSignatureID	1.3.6.1.4.1.2636.3.9.1.66.1.2.10 ...	Attack object database ID.
jnxIdpSensorTopTenSignatureName	1.3.6.1.4.1.2636.3.9.1.66.1.3.10 ...	Attack object name.
jnxIdpSensorTopTenSignatureHits	1.3.6.1.4.1.2636.3.9.1.66.1.4.10 ...	Matches (count).
jnxIdpSensorCpuld	1.3.6.1.4.1.2636.3.9.1.67.0	Not accessible. This object is used in trap notification objects.
jnxIdpSensorSessnCreateRateThreshold	1.3.6.1.4.1.2636.3.9.1.68.0	Not accessible. This object is used in trap notification objects.

Table 161: IDP Series MIB Objects (*continued*)

Object Name	Object Identifier	Description
jnxIdpSensorFreePktThreshold	1.3.6.1.4.1.2636.3.9.1.69	Not accessible. This object is used in trap notification objects.
jnxIdpSensorPktsDropOnAllIntfcThreshold	1.3.6.1.4.1.2636.3.9.1.70	Not accessible. This object is used in trap notification objects.
jnxIdpSensorPktsDropDueToRxOverflowThreshold	1.3.6.1.4.1.2636.3.9.1.71	Not accessible. This object is used in trap notification objects.
jnxIdpSensorPktsDropRateOnAllIntfcThreshold	1.3.6.1.4.1.2636.3.9.1.72	Not accessible. This object is used in trap notification objects.
jnxIdpSensorRxPktsDropPerIntfcThreshold	1.3.6.1.4.1.2636.3.9.1.73	Not accessible. This object is used in trap notification objects.
jnxIdpSensorRxPktsDropRatePerIntfcThreshold	1.3.6.1.4.1.2636.3.9.1.74	Not accessible. This object is used in trap notification objects.
jnxIdpSensorTxPktsDropAllIntfcThreshold	1.3.6.1.4.1.2636.3.9.1.75	Not accessible. This object is used in trap notification objects.
jnxIdpSensorTxPktsDropPerIntfcThreshold	1.3.6.1.4.1.2636.3.9.1.76	Not accessible. This object is used in trap notification objects.
jnxIdpSensorPktDropCount	1.3.6.1.4.1.2636.3.9.1.77	Not accessible. This object is used in trap notification objects.
jnxIdpSensorIfName	1.3.6.1.4.1.2636.3.9.1.78	Not accessible. This object is used in trap notification objects.
jnxIdpSensorIfStatus	1.3.6.1.4.1.2636.3.9.1.79	Not accessible. This object is used in trap notification objects.

[Table 162 on page 567](#) describes the MIB objects used to implement SNMP trap notifications.

Table 162: IDP Series Traps

Object Name	Object ID	Description
jnxIdpSessionCountNotify	jnxIdpTrapsPrefix 1	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorSessAllocated, jnxIdpSensorSessThreshold

Table 162: IDP Series Traps (*continued*)

Object Name	Object ID	Description
jnxIdpSessionCountLimitRestored	jnxIdpTrapsPrefix 2	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorSessAllocated
jnxIdpCPUUtilizationNotify	jnxIdpTrapsPrefix 3	Alarm is triggered when the control plane CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId, jnxIdpSensorCpuThreshold
jnxIdpCPUUtilizationLimitRestored	jnxIdpTrapsPrefix 4	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId
jnxIdpMemoryNotify	jnxIdpTrapsPrefix 5	Alarm is triggered when utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorMemUsage, jnxIdpSensorMemThreshold
jnxIdpMemoryLimitRestored	jnxIdpTrapsPrefix 6	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorMemUsage
jnxIdpDiskUtilizationNotify	jnxIdpTrapsPrefix 7	Alarm is triggered when utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorFreeDiskSpace, jnxIdpSensorDiskSpaceThreshold
jnxIdpDiskUtilizationLimitRestored	jnxIdpTrapsPrefix 8	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorFreeDiskSpace
jnxIdpControlCpuUsgFiveMinNotify	jnxIdpTrapsPrefix 9	Alarm is triggered when control plane CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId, jnxIdpSensorCpuThreshold

Table 162: IDP Series Traps (*continued*)

Object Name	Object ID	Description
jnxIdpControlCpuUsgFiveMinRestored	jnxIdpTrapsPrefix 10	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId
jnxIdpEngineCpuUsgOneMinNotify	jnxIdpTrapsPrefix 11	Alarm is triggered when IDP engine CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId, jnxIdpSensorCpuThreshold
jnxIdpEngineCpuUsgOneMinRestored	jnxIdpTrapsPrefix 12	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId
jnxIdpEngineCpuUsgFiveMinNotify	jnxIdpTrapsPrefix 13	Alarm is triggered when IDP engine CPU utilization exceeds the configured threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId, jnxIdpSensorCpuThreshold
jnxIdpEngineCpuUsgFiveMinRestored	jnxIdpTrapsPrefix 14	Notification is triggered when utilization falls below the alarm threshold.  Objects referenced: jnxIdpSensorCpuUsage, jnxIdpSensorCpuId
jnxIdpEngineSessnCreateRateNotify	jnxIdpTrapsPrefix 15	Alarm is triggered when the rate exceeds the configured threshold.  Objects referenced: jnxIdpSensorSessnCreateRateFiveSec, jnxIdpSensorSessnCreateRateThreshold
jnxIdpEngineSessnCreateRateRestored	jnxIdpTrapsPrefix 16	Notification is triggered when the rate falls below the alarm threshold.  Objects referenced: jnxIdpSensorSessnCreateRateFiveSec
jnxIdpFreePktLastFiveSecNotify	jnxIdpTrapsPrefix 17	Alarm is triggered when the count falls below the configured threshold.  Objects referenced: jnxIdpSensorFreePktBuffersFiveSec, jnxIdpSensorFreePktThreshold

Table 162: IDP Series Traps (*continued*)

Object Name	Object ID	Description
jnxIdpFreePktLastFiveSecRestored	jnxIdpTrapsPrefix 18	Notification is triggered when the count is restored above the alarm threshold.  Objects referenced: jnxIdpSensorFreePktBuffersFiveSec
jnxIdpFreePktLastOneMinNotify	jnxIdpTrapsPrefix 19	Alarm is triggered when the count falls below the configured threshold.  Objects referenced: jnxIdpSensorFreePktBuffersOneMin, jnxIdpSensorFreePktThreshold
jnxIdpFreePktLastOneMinRestored	jnxIdpTrapsPrefix 20	Notification is triggered when the count is restored to above the alarm threshold.  Objects referenced: jnxIdpSensorFreePktBuffersOneMin
jnxIdpTotalRxPktsDropNotify	jnxIdpTrapsPrefix 21	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktsDropOnAllIntfc, jnxIdpSensorPktsDropOnAllIntfcThreshold
jnxIdpTotalRxPktsDropRestored	jnxIdpTrapsPrefix 22	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktsDropOnAllIntfc
jnxIdpTotalRxPktsDropRateNotify	jnxIdpTrapsPrefix 23	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktsDropRateOnAllIntfc, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpTotalRxPktsDropRateRestored	jnxIdpTrapsPrefix 24	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktsDropRateOnAllIntfce
jnxIdpPerifRxOverflowNotify	jnxIdpTrapsPrefix 25	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold

Table 162: IDP Series Traps (*continued*)

Object Name	Object ID	Description
jnxIdpPerIfRxOverflowRestored	jnxIdpTrapsPrefix 26	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount
jnxIdpPerIfRxDropNotify	jnxIdpTrapsPrefix 27	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpPerIfRxDropRestored	jnxIdpTrapsPrefix 28	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount
jnxIdpPerIfRxDropRateNotify	jnxIdpTrapsPrefix 29	Alarm is triggered when the rate exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpPerIfRxDropRateRestored	jnxIdpTrapsPrefix 30	Notification is triggered when the rate falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount
jnxIdpPerIfTxDropNotify	jnxIdpTrapsPrefix 31	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpPerIfTxDropRestored	jnxIdpTrapsPrefix 32	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount

Table 162: IDP Series Traps (*continued*)

Object Name	Object ID	Description
jnxIdpTotalTxDropNotify	jnxIdpTrapsPrefix 33	Alarm is triggered when the count exceeds the configured threshold.  Objects referenced: jnxIdpSensorPktDropCount, jnxIdpSensorIfName, jnxIdpSensorPktsDropRateOnAllIntfcThreshold
jnxIdpTotalTxDropRestored	jnxIdpTrapsPrefix 34	Notification is triggered when the count falls below the alarm threshold.  Objects referenced: jnxIdpSensorPktDropCount
jnxIdpIntfcDownNotify	jnxIdpTrapsPrefix 35	Alarm is triggered when status is down.  Objects referenced: jnxIdpSensorIfStatus, jnxIdpSensorIfName
jnxIdpIntfcStatusRestored	jnxIdpTrapsPrefix 36	Notification is triggered when a down interface is restored (up).  Objects referenced: jnxIdpSensorIfStatus, jnxIdpSensorIfName

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring an SNMP Agent \(NSM Procedure\) on page 299](#)
- [Configuring SNMP Reporting for the Sensor Category of Statistics on page 435](#)
- [Configuring SNMP Reporting for the Resource Category of Statistics on page 428](#)
- [Configuring SNMP Reporting for the Interface Category of Statistics on page 418](#)
- [Configuring SNMP Reporting for the Traffic Category of Statistics on page 438](#)
- [Configuring SNMP Reporting for the Rule Category of Statistics on page 433](#)
- [Example: Querying the IDP Series Device MIB on page 151](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Logs Overview on page 24](#)



## PART 6

# Troubleshooting

- [Troubleshooting References on page 575](#)
- [Simulation Mode on page 579](#)
- [Troubleshooting Feature Implementation on page 581](#)



# Troubleshooting References

- Troubleshooting Tools Overview on page 575
- IDP Processes Reference on page 577
- Troubleshooting NSM Log Collection Issues on page 578

## Troubleshooting Tools Overview

The best troubleshooting tips and troubleshooting workflows are published in the Juniper Networks Technical Assistance Center (JTAC) knowledge base at <http://kb.juniper.net>. For example, <http://kb.juniper.net/index?page=content&id=KB9777> provides a workflow for diagnosing dropped traffic.

Table 163 on page 575 provides a summary of IDP Series troubleshooting tools.

Table 163: IDP Series Troubleshooting Tools

Tool	Description
tech-support	<p>The <b>tech-support</b> utility runs the following commands in the background and saves the output to a zipped temporary file you can e-mail to JTAC:</p> <ul style="list-style-type: none"><li>• <b>getplatform</b></li><li>• <b>ps</b></li><li>• <b>df</b></li><li>• <b>lsof</b></li><li>• <b>du</b></li><li>• <b>ifconfig</b></li><li>• <b>netstat</b></li><li>• <b>scio sysconf all</b></li><li>• <b>scio const list</b></li><li>• <b>scio vr list</b></li><li>• <b>scio vc list</b></li><li>• <b>ping</b></li><li>• <b>tcpdump</b></li></ul> <p>If you want to view the contents of the zip files, use the <b>bunzip2</b> command.</p>

Table 163: IDP Series Troubleshooting Tools (*continued*)

Tool	Description
<b>tcpdump</b>	<p>The <b>tcpdump</b> utility captures traffic and saves it to a file. For example, to perform a packet capture and save SMTP packets on interface eth1 to a file, use the following command:</p> <pre><b>tcpdump -i eth1 -s 0 -w /tmp/smtp.pcap tcp port 25</b></pre> <p>For more information, see <a href="#">“Using tcpdump to Capture Packets” on page 481</a>.</p>
<b>jnetTcpdump</b>	<p>An IDP OS Release 5.1 utility. Capable of capturing both Rx and Tx packets. The following example starts listening on eth4 for packets with destination IP address 4.0.0.4:</p> <pre>[root@localhost ~]# jnetTcpdump -i eth4 -f 4.0.0.4 dst jnetPassiveAttach done jnet tcpdump Started on eth4 for both Receive &amp; Transmit side Filter enabled - Host:4.0.0.4 as dst 0 50 56 a4 21 6c 0 50 56 a4 d 9 8 0 45 0 0 54 0 0 40 0 40 1 32 a3 4 0 0 3 4 0 0 4 8 0 55 8e 8e 4f 0 0 ba 9f 3e 4d 21 32 f 0 8 9 a b c d e f 10 11 12 13 14 15 0 50 56 a4 21 6c 0 50 56 a4 d 9 8 0 45 0 0 54 0 0 40 0 40 1 32 a3 4 0 0 3 4 0 0 4 8 0 97 88 8e 4f 0 1  bb 9f 3e 4d de 36 f 0 8 9 a b c d e f 10 11 12 13 14 15 Done...No of Packet Captured is 2 No of Packets filtered-out 2</pre> <p>For more information, see <a href="#">“Using jnetTcpdump to Capture Packets” on page 481</a>.</p>
<b>scio ccap all</b>	<p>In some cases, packet captures might be helpful to reproduce an issue so that it can be analyzed and resolved. The following command captures all services and contexts from all sessions:</p> <pre><b>scio ccap all</b></pre> <p>IDP8200 has multiple IDP engines. For IDP8200, <b>scio ccap all</b> returns data for idpengine_0. To capture data from other engines, use the <b>-c</b> option and specify the engine number (0 through 5). For example, <b>scio -c 1 ccap all</b> returns data for idpengine_1, <b>scio -c 2 ccap all</b> returns data for idpengine_2, and so forth.</p> <p><b>NOTE:</b> The <b>scio ccap all</b> command captures the same contexts as Profiler. You cannot use <b>scio ccap all</b> when Profiler is running.</p>
<b>scio pcap</b>	<p>You can use <b>scio pcap</b> to replay traffic that was previously captured by tools like <b>tcpdump</b> and <b>scio ccap</b>.</p> <p>For examples of using <b>scio ccap</b> and <b>scio pcap</b>, see the <a href="#">IDP Series Custom Attack Object Reference and Examples Guide</a>.</p> <p><b>NOTE:</b> You can not use <b>scio pcap</b> on a host where virtual routers are configured in a mix of sniffer and transparent mode (mixed mode).</p>
<b>tcpreplay</b>	<p>You can use <b>tcpreplay</b> to edit and replay network traffic that was previously captured by tools like <b>tcpdump</b> and <b>scio ccap</b>. Refer to the Linux man pages for details.</p>
<b>ethtool</b>	<p>You can use <b>ethtool</b> to query and configure network interfaces. Refer to the Linux man pages for details.</p> <p><b>NOTE:</b> Interface management with <b>mii-tool</b> is not supported.</p>
<b>IDP debug build</b>	<p>In some cases, JTAC might recommend you run a special build of the IDP OS to generate debugging information that can be used to determine the root cause of an issue.</p>

Table 163: IDP Series Troubleshooting Tools (*continued*)

Tool	Description
Reimaging	If necessary, you can revert to the factory image of the IDP Series device. For information, see the <a href="#">installation guide</a> for your IDP appliance.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [IDP Processes Reference on page 577](#)

## IDP Processes Reference

Specific IDP processes generate error messages. Knowing the process that encountered the error can often help you isolate and resolve the issue.

[Table 164 on page 577](#) provides a reference of IDP processes.

Table 164: Troubleshooting: IDP Processes Reference

Process	Function
agent	Establishes the Transport Layer Security (TLS) channel to Network and Security Manager (NSM). Sends IDP status, logs, and profiled data to NSM. Receives policy, detector, and configuration commands from NSM.
idpengine	The core IDP engine.
idpHMD	Health monitoring daemon. Monitors control plane and data plane resource utilization. Communicates status to the SNMP agent and SNMP trap service.
idpLogReader	Reads IDP logs and writes them to local hard disk.
nicBypass	Controls the internal bypass feature.
peerPortModulator	Controls peer port modulation.
pkid	Inspects SSL traffic, if SSL inspection is turned on.
profiler	Profiles network and application data collected by the device.
recover.sh	Used in the auto-recovery process.
sciod	Handles policy push, information retrieval, Profiler status, and so on.
sessionFetcher	When packet logging is enabled, retrieves session data and sends it to NSM.
slogd	Logs packet captures to the IDP Series device hard disk.
snmpd	SNMP agent.

Table 164: Troubleshooting: IDP Processes Reference (*continued*)

Process	Function
snmptrapd	SNMP trap service.



**NOTE:** You can use the Linux `ps` commands to display the process ID and other status information about IDP processes.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [idp.sh Command Reference on page 385](#)

## Troubleshooting NSM Log Collection Issues

**Problem** You can use the `scio logview` utility to view contents of log files before the logs are forwarded to NSM. This way, if you suspect a problem with logging features, you can compare the device-side logs with the NSM-side logs.

**Solution** The following example commands show how to navigate to the logs directory, sort by date, and use the `scio logview` command to display contents of a recent log.

```
[root@defaulthost ~]# cd /var/idp/device/logs/
[root@defaulthost logs]# ls -lat | less
drwx----- 2 idp idp 69632 Aug  5 11:50 .
-rw----- 1 idp idp 2788 Aug  5 11:50 1281034151.log
-rw----- 1 idp idp 212 Aug  5 11:50 1281034242.log
-rw----- 1 idp idp 0 Aug  5 11:50 1281034242.wait
-rw----- 1 idp idp 384 Aug  5 11:49 1281034128.log
-rw----- 1 idp idp 1232 Aug  5 11:48 1281034081.log
-rw----- 1 idp idp 1680 Aug  5 11:47 1281034035.log
-rw----- 1 idp idp 744 Aug  5 11:47 1281033989.log
-rw----- 1 idp idp 1868 Aug  5 11:46 1281033942.log
-rw----- 1 idp idp 952 Aug  5 11:45 1281033916.log
-rw----- 1 idp idp 260 Aug  5 11:44 1281033804.log
-rw----- 1 idp idp 260 Aug  5 11:43 1281033699.log
-rw----- 1 idp idp 260 Aug  5 11:41 1281033590.log
-rw----- 1 idp idp 260 Aug  5 11:39 1281033484.log
-rw----- 1 idp idp 260 Aug  5 11:37 1281033386.log
-rw----- 1 idp idp 148 Aug  5 11:36 1281033138.log

[root@defaulthost logs]# scio logview 1281034242.log
Log :Time Generated : Thu Aug  5 11:50:41 2010
Source IP 0.0.0.0 Source Port :0 -> Destination IP 0.0.0.0 Destination Port :0
Category Enum : attackid:805306379 Severity Enum :SC_LOG_SEVERITY_INFO
Protocol Enum :0 Action :SC_LOG_ACTION_NOT_SET
srcIface : , Details : Percentage of Control CPU usage last 5 minutes has
restored below threshold and is at 57 [Simulation Mode]
```

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [scio logview on page 526](#)

# Simulation Mode

- [Example: Using Simulation Mode to Maximize Uptime on page 579](#)

## Example: Using Simulation Mode to Maximize Uptime

---

The primary use case for simulation mode is for evaluating whether to adopt the IDP Series device as the intrusion prevention device for your network. You might also find simulation mode useful after you adopt the IDP Series as your IDP solution when you want to maximize network availability while you tune a security policy update or troubleshoot traffic outages when IDP processing results in crashes.

Suppose you discover that the device is dropping traffic and that early indicators suggest a likely false positive and that the traffic probably can be trusted. This situation might happen, for example, after an attack object database update when new attack signatures are added to a dynamic attack group that is specified in your IDP rulebase rules.

In cases like this, your choices are:

- Continue to drop traffic while you investigate.
- Change the rule action to allow traffic while you investigate. This requires you to reload the security policy with the changed rule action.
- Shut down the IDP Series device while you investigate. If you enable internal bypass, traffic passes through the device.
- Use simulation mode.

In cases like this, simulation mode is a good choice if you are an experienced IDP security administrator who suspects a false positive and are inclined to maximize uptime while you investigate. If you later conclude that it is not a false positive, you can disable simulation mode and return to active management without having to reload the security policy. You can use the logs collected during simulation mode to follow up on any subsequent security actions to take. If, on the other hand, your investigation confirms your hunch that it is a false positive, you can make iterations of modifications to your policy, load the changed policy, and observe the results. When you are satisfied with the results, you can disable simulation mode.

Simulation mode is not a good choice if you are not an experienced IDP security administrator or when you suspect a critical security risk. In these cases, we recommend that you continue to drop traffic while you investigate.

You might also switch to simulation mode on your live network when you are troubleshooting traffic outages due to IDP processing crashes. Before the IDP Series supported simulation mode, your customer support representative might have advised you to deploy the device in sniffer mode while you were waiting for a detector engine update or service patch to resolve the root cause of a crash. With IDP OS Release 5.1 and later, simulation mode is a good choice if you want to leave the device physically in path (you do not want to reconfigure and reconnect your traffic interfaces as required for the out-of-path, sniffer mode deployment). However, in these situations, sniffer mode is a better choice if you want the device to send TCP resets to close connections when a security policy rule matches.

**Related Documentation**

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [Simulation Mode Overview on page 33](#)
- [Sniffer Mode Overview](#)
- [Example: Fine-Tuning a Security Policy on page 48](#)

The following related topic is included in the *IDP Series Administration Guide*:

- [Enabling Simulation Mode on page 201](#)



# Troubleshooting Feature Implementation

- [Tuning the JNET Driver Failure Count on page 581](#)
- [Viewing Auto-Recovery Logs on page 583](#)
- [Disabling the Auto-Recovery Feature on page 584](#)
- [Tuning the Auto-Recovery Policy Reload Setting on page 584](#)
- [Troubleshooting SNMP Statistic Reporting on page 585](#)
- [Viewing CPU Utilization on page 586](#)
- [Troubleshooting High CPU Usage on page 588](#)
- [Troubleshooting Erroneous CPU Utilization Reports on page 590](#)
- [Displaying Service Session Count on page 592](#)
- [Troubleshooting Configuration Push Errors \(NSM Procedure\) on page 593](#)
- [Troubleshooting Security Policy Validation Errors \(NSM Procedure\) on page 594](#)
- [Troubleshooting Application Identification on page 595](#)
- [Disabling the APE Rulebase on page 598](#)
- [Disabling the User Role-Based Policy Feature on page 599](#)
- [Disabling Support for Jumbo Frames on page 599](#)
- [Troubleshooting SSL Inspection on page 600](#)
- [Disabling SSL Inspection on page 600](#)
- [Disabling MPLS Decapsulation on page 601](#)

## Tuning the JNET Driver Failure Count

---

**Problem** If you configure internal bypass for virtual routers, traffic will bypass processing if the IDP engine is unavailable or JNET driver fails to transmit packets as expected.

In the latter case, internal bypass is triggered when the JNET driver receive queue failure count reaches 18 (about three minutes). After you have analyzed and resolved the cause of the JNET driver failure, you can return the interfaces to normal state by restarting the IDP engine.

We have provided a tunable setting in case you prefer the IDP Series device to enter bypass sooner (decrease the count). If you want to give the JNET driver more time to self-correct, you can increase the failure count.

To tune the default, modify the `max_intf_rcv_failed_cnt_nicbypass` value in the `user_funcs` file. You must run **idp.sh restart** to restart the IDP engine and initialize any changes.

**Solution** To tune the JNET driver failure count:

1. Log into the CLI as **admin** and enter **su -** to switch to root.
2. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as **vi**.
3. Locate the following line in the Variables section of the file:

```
#####
VARIABLES
#####
...
# 'max_intf_rcv_failed_cnt_nicbypass' - The maximum count value for any
# data interface indicating the number of times the packet could not
# be received by that interface. If the count for any interface reaches
# this value nicBypass gets triggered.
# **WARNING**: Changing the value would require running 'idp.sh restart'.

export max_intf_rcv_failed_cnt_nicbypass=18
```



**NOTE:** If your upgrade path was from 5.0r1 to 5.0r2, your `user_funcs` files does not include the excerpt shown above. If this line is not present, the default is used. If you want to change the default, add the export statement shown to the Variables section of the `user_funcs` file.

4. Modify the default value (18) as you prefer. The value 18 is approximately equivalent to 180 seconds. A value of 6 would be approximately 60 seconds.
5. Save the file and exit the editor.
6. Restart the IDP engine:

```
[root@default host admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

#### Related Documentation

The following related topic is included in the *IDP Series Administration Guide*:

- [Configuring Virtual Routers \(ACM Procedure\) on page 192](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)
- [IDP Series Network Interfaces Overview on page 11](#)

## Viewing Auto-Recovery Logs

**Purpose** Use recovery-related logs as cues to examine the surrounding traffic logs. The surrounding logs can indicate the type of traffic that caused the failure condition.

In addition, the auto-recovery process is unable to identify the application for buffered sessions. As a result, in processing buffered traffic, the application identification feature is unavailable and application rate limiting cannot be applied. In addition, the latest interval of application volume tracking data is discarded. Keep this in mind when examining application-related logs generated immediately after restart.

Table 165 on page 583 describes recovery-related logs.

**Table 165: Auto-Recovery Logs**

Log Message	Description
IDP instance <i>number</i> successfully recovered.	Verifies successful restart. Success indicates that the IDP engine was restarted with the same device configuration, feature configuration, and security policy that were in place before the restart.
IDP instance <i>number</i> is detected to be terminated.	Indicates a particular IDP engine has encountered a condition that requires the IDP engine to be terminated and restarted.
Failed to recover IDP instance <i>number</i> .	Indicates a failure restarting.
Restarting IDP instance for the <i>n</i> th time.	Indicates that a particular IDP engine has been terminated and restarted a total of <i>number</i> times.
IDP is being stopped since IDP instance <i>number</i> restarted for <i>n</i> th time.	Indicates that the auto-recovery feature has reached its maximum number of restart attempts before shutting down the IDP Series device. The auto-recovery process makes up to six attempts to restart the failed IDP engine. After six failed attempts, the IDP auto-recovery process issues an <b>idp.sh stop</b> command. If you have enabled internal bypass, the IDP Series device enters bypass.

**Action** Use the NSM Log Viewer filtering features to identify auto-recovery events and examine the surrounding traffic. Auto-recovery logs belong to category Alarm and subcategory Others.

You can also find auto-recovery logs on the IDP Series device in `/var/idp/device/sysinfo/logs/idpinit.date`.

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [IDP Series Logs and Reports in NSM Task Summary on page 295](#)
- [Tuning the Auto-Recovery Bypass Setting](#)
- [Tuning the Auto-Recovery Policy Reload Setting on page 584](#)
- [Disabling the Auto-Recovery Feature on page 584](#)
- [idp.sh Command Reference on page 385](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)

---

## Disabling the Auto-Recovery Feature

**Problem** If you encounter a problem with the auto-recovery feature, you can disable it.

**Solution** To disable the auto-recovery feature:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Stop the IDP engine:  

```
[root@defaulthost admin]# idp.sh stop
```

Restarting the IDP engine can take several moments.
3. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as **vi**.
4. Locate the following line:  

```
export autorecovery_state=1
```
5. Change the value to **0** to disable the feature.
6. Save the file and exit the editor.
7. Restart the IDP engine:  

```
[root@defaulthost admin]# idp.sh start
```

Starting the IDP engine can take several moments.

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing Auto-Recovery Logs on page 583](#)
- [Tuning the Auto-Recovery Bypass Setting](#)
- [Tuning the Auto-Recovery Policy Reload Setting on page 584](#)

---

## Tuning the Auto-Recovery Policy Reload Setting

**Problem** The auto-recovery feature detects failure of an IDP engine and buffers packets while it attempts to restart the IDP engine. The auto-recovery process reloads the device configuration, including the security policy. The larger the security policy, the longer it takes to complete the auto-recovery process. By default, packet processing resumes only after the security policy has been reloaded. If your deployment requires faster resumption of traffic flow, you can change this setting so that the IDP engine begins processing traffic before the security policy has been loaded. However, the packets that are processed before the security policy has been loaded are uninspected.

**Solution** To set packet processing to resume before the security policy has been loaded:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Open the `/usr/idp/device/bin/user_funcs` file in a text editor, such as **vi**.
3. Locate the following line:  

```
export pktprocess_afterpolicyload=1
```
4. Change the value to **0** so that packet processing resumes before the security policy has been loaded.
5. Save the file and exit the editor.
6. Restart the IDP engine:  

```
[root@default host admin]# idp.sh restart
```

Restarting the IDP engine can take several moments.

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing Auto-Recovery Logs on page 583](#)
- [Tuning the Auto-Recovery Bypass Setting](#)
- [Disabling the Auto-Recovery Feature on page 584](#)

## Troubleshooting SNMP Statistic Reporting

**Problem** Many SNMP query tools indicate the cause of SNMP reporting failures, such as an unreachable host. If necessary, you can investigate the status of the health monitoring daemon (HMD) and snmpd processes on the IDP Series device.

**Solution** SNMP statistic and trap reporting depend on the health of the HMD and the SNMP reporting daemon (snmpd) that run on the IDP Series device. [Figure 160 on page 585](#) illustrates collection via HMD and reporting via SNMP.

**Figure 160: SNMP Statistic Report Processes**

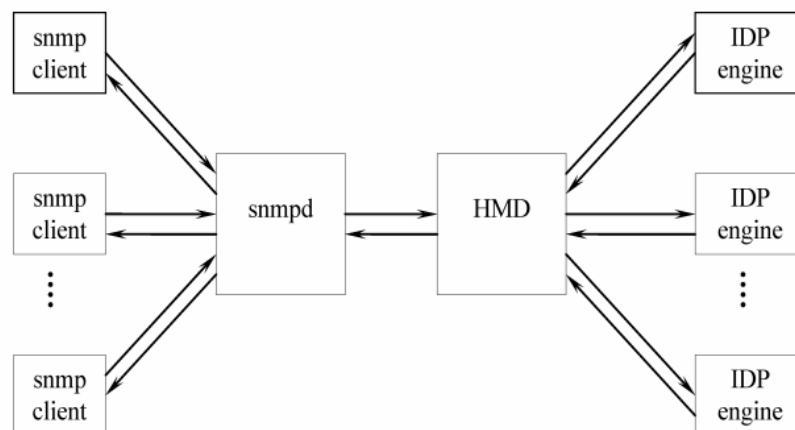


Table 166 on page 586 describes troubleshooting steps.

**Table 166: Diagnosing Problems with SNMP Reporting**

Symptom	Troubleshooting Steps
SNMP query times out.	<ol style="list-style-type: none"> <li>Use the following command to query the status of the health monitoring daemon: <pre>[root@default host ~]# idpHMD.sh status</pre> <pre>idpHMD (pid 13691).....ON</pre> <p>If necessary, use the following command to restart the service:</p> <pre>[root@default host ~]# idpHMD.sh restart</pre> </li> <li>Use the following command to query the status of the SNMP agent: <pre>[root@default host ~]# /sbin/service snmpd status</pre> <pre>snmpd (pid 4729) is running...</pre> <p>If necessary, use the following command to restart the service:</p> <pre>[root@default host ~]# /sbin/service snmpd restart</pre> </li> <li>Use <b>scio sri</b> commands to verify that the category of statistics is enabled.</li> <li>In the NSM Device Manager, check the report settings to verify that SNMP reporting is enabled.</li> </ol>
Traps not received.	<ol style="list-style-type: none"> <li>Use the following command to query the status of the SNMP trap service: <pre>[root@default host ~]# /sbin/service snmptrapd status</pre> <pre>snmpd (pid 4729) is running...</pre> <p>If necessary, use the following command to restart the service:</p> <pre>[root@default host ~]# /sbin/service snmptrapd restart</pre> </li> <li>Use <b>scio sri</b> commands to verify that the category of statistics is enabled and the trap thresholds are expected values.</li> <li>In the NSM Device Manager, check the report settings to verify that SNMP reporting is enabled and that it specifies the expected SNMP manager IP address.</li> </ol>

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [SNMP Statistic Reporting and Traps Task Summary on page 417](#)

## Viewing CPU Utilization

**Problem** In earlier releases of the IDP OS, the CPU utilization reported for the system was simply the results from the Linux **top** command. This has not changed for single core platforms (IDP75, IDP200, IDP600): the reported CPU utilization continues to be the results of **top**. For multicore platforms (IDP250, IDP800, IDP1100, IDP8200), the results of the Linux **top** command are not helpful. Because of JNET driver polling behavior on multicore platforms, the **top** command shows constant 100% utilization. To address this issue, we developed the **scio idp-cpu-utilization** utility for multicore platforms. We do not claim that the CPU utilization reported by the **scio idp-cpu-utilization** utility to be as precise as the Linux **top** command, but the results are helpful in tracking changes in CPU utilization

and gauging whether utilization is normal or abnormal (high or low). [Table 167 on page 587](#) summarizes how CPU utilization is reported for IDP Series platforms.

**Table 167: CPU Monitoring Tools**

	CLI	SNMP	NSM	IDP Reporter
Single core (IDP75, IDP200, IDP600)	<b>top</b>	<b>top</b>	Do not use: PR 434539	<b>top</b>
Dual core (IDP250, IDP800, IDP1100)	<b>scio idp-cpu-utilization</b>	<b>scio idp-cpu-utilization</b>	Do not use: PR 434539	<b>scio idp-cpu-utilization</b>
Multicore (IDP8200)	Aggregate CPU utilization of all IDP8200 IDP engines): <b>scio idp-cpu-utilization</b>	<b>scio idp-cpu-utilization</b>	Do not use: PR 434539	<b>scio idp-cpu-utilization</b>
	CPU utilization per IDP engine:  <b>scio -c 0 idp-cpu-utilization</b> <b>scio -c 1 idp-cpu-utilization</b> <b>scio -c 2 idp-cpu-utilization</b> <b>scio -c 3 idp-cpu-utilization</b> <b>scio -c 4 idp-cpu-utilization</b> <b>scio -c 5 idp-cpu-utilization</b>	No	No	No



**NOTE:** Using the **scio idp-cpu-utilization** utility can cause CPU usage to spike. We, therefore, recommend you use **scio idp-cpu-utilization** only for debugging.

**Solution** To display CPU utilization for IDP75, IDP200, and IDP600:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the Linux **top** command to display resource utilization statistics for IDP OS processes:

```
[root@defaultthost ~]# top
top - 01:39:36 up 14 days, 10:27, 1 user, load average: 2.33, 2.77, 2.84
Tasks: 82 total, 1 running, 81 sleeping, 0 stopped, 0 zombie
Cpu(s): 7.9%us, 38.4%sy, 0.0%ni, 53.0%id, 0.3%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 3007268k total, 2831804k used, 175464k free, 254104k buffers
Swap: 4192928k total, 29608k used, 4163320k free, 2081808k cached
```

```

    PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     TIME+  COMMAND
  27348 root      -31 -20    0    0    0  D   17.9   0.0   3636:39  kjnetd
  26188 root       10 -10 1972m 333m 132m  S   2.0  11.3   202:10.56  idpengine
  29222 root        0 -20  8992 1436  940  S   1.7   0.0   280:55.81  nicBypass
  28772 root       20  0  9000 1372  868  S   1.0   0.0   277:06.88  peerPortModulat
 16012 root       20  0 65936  988  824  S   0.3   0.0    6:33.82  idpprocesschk
 27877 idp        20  0 18604 3516 1800  S   0.3   0.1   14:18.50  agent
 28013 idp        20  0 13352 3844 1528  S   0.3   0.1   11:32.45  idpHMD
      1 root       20  0 10388  624  524  S   0.0   0.0    0:09.08  init
      2 root       20  0    0    0    0  S   0.0   0.0    0:00.00  kthreadd
```

```
3 root      20    0    0    0    0 S  0.0  0.0  10:25.39 ksoftirqd/0
4 root      20    0    0    0    0 S  0.0  0.0   0:00.00 kworker/0:0
```

To display CPU utilization for IDP250, IDP800, IDP1100, and IDP8200:

1. Log into the CLI as **admin** and enter **su** - to switch to **root**.
2. Use the following command to display CPU usage for the CPU core running the IDP engine:

```
[root@defaulthost admin]# scio idp-cpu-utilization
Current actual cpu utilization: 0
```

The IDP8200 has multiple IDP engines. Specify the **-c idp-engine** option to display CPU usage for a particular CPU core (0–5):

```
[root@defaulthost admin]# scio -c 0 idp-cpu-utilization
Current actual cpu utilization: 0
```

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring SNMP Reporting for the Sensor Category of Statistics on page 435](#)
- [Troubleshooting High CPU Usage on page 588](#)
- [Troubleshooting Erroneous CPU Utilization Reports on page 590](#)
- [scio idp-cpu-utilization on page 525](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)

---

## Troubleshooting High CPU Usage

**Problem** Security policy rules processing has an impact on CPU usage. You can use **scio** signature statistic tool to understand the CPU utilization used to match particular attack objects or attack groups. You can then take a better informed approach to tuning your security policy.



**NOTE:** There is a slight performance impact to using this feature. Enable this feature while you are investigating high CPU; otherwise, disable it.

To use the **scio** signature statistic tool:

1. Log into the CLI as **admin** and enter **su** - to switch to **root**.
2. Enter the following command to enable the signature statistic tool:

```
[root@defaulthost ~]# scio const -s s0 set sc_enable_sigstat 1
```



### 3. Configure a threshold to target attack objects that use a lot of CPU.

- a. Enter the following command to get started:

```
[root@default host ~]# scio const -s s0 set sc_sigstat_nano_secs 1000000
```

In this example, the value 1000000 is a filter to log only attack objects that use more than 1,000,000 CPU cycles. The default is 10,000 (0x2710).

- b. Search the log file to examine its contents. Logs are saved in the `/usr/idp/device/var/sysinfo/logs` directory. Files are named `idpengine_num.date`.

Logs have the following format:

```
sigmatch stat: [function name] [attack/group: attack_name/group_name] [cpu
cycles: cpu cycles consumed]
for the flow : srcIP:srcPort -> dst IP: dst Port [protocol number]
```

For example:

```
sigmatch stat: [sc_ids_run_packet] [attack: DNS:OVERFLOW:BIN] [cpu cycles:
12462695]
for the flow : 192.168.1.100:59773 -> 192.168.1.24:53 [17]
sigmatch stat: [sc_ids_run_stream256] [group: dg_c11e-1f0d0p17s0n00_10] [cpu
cycles: 31026226]
for the flow : 192.168.1.100:1502 -> 192.168.1.24:8080 [6]
```

In the above example, note that the attack object DNS:OVERFLOW:BIN and the attack group dg\_c11e-1f0d0p17s0n00\_10 are using a large number of CPU cycles.

- c. If you get too many matches, you can adjust your filter for `sc_sigstat_nano_secs` (upwards to narrow the set that matches, downwards to broaden the set that matches).
- d. Make a note of attack objects and groups that generate a lot of matches. You probably do not need to be concerned about occasional matches. You want to identify patterns of high CPU cycles.

Use the following command syntax to display the attacks contained in the attack group:

```
[root@default host ~]# scio policy attacks subscriber policy.set-path detector-path
group-name
```

For example:

```
[root@default host ~]# scio policy attacks s0 /usr/idp/device/state/s0/policy.set
/usr/idp/device/lkm/detector.o.gz.v dg_c11e-1f0d0p17s0n00_10
```

The following is example output:

Attacks in the group 'dg\_c11e-1f0d0p17s0n00\_10' are... [10 attacks]

```
APP:UPNP:APPLE-MDNS_1
MS-RPC:MESSENGER-OF2_1
APP:UPNP:APPLE-MDNS_2
RPC:RPC.SADMIN:METHOD-TRAV-UDP
APP:UPNP:APPLE-MDNS_3
MS-RPC:MESSENGER-OF2_2
MS-RPC:MESSENGER-OF
MS-RPC:MESSENGER-OF2_3
```

SCAN:CANVAS:LinuxSNMP  
TROJAN:BACKORIFICE:BOPING-WIN

4. If necessary, use the suggestions in the next section to refine your security policy rules. We recommend creating a separate rule for an attack object or group that has a pattern of high CPU usage. Later, if you encounter an issue with high CPU usage, you have the option of setting the rule action temporarily to **Ignore** while you investigate the CPU spike.

**Solution** If you observe high CPU usage, consider the following adjustments to your security policy rules:

- In general, it is better to create more rules with a few attack objects each than to create few rules with many attack objects in each. Being specific with rules might also facilitate network analysis and troubleshooting.
- Ensure your rule includes only the attack objects applicable to the service and destination server identified in the rule. Each attack object has a performance cost.
- Include only the attack objects related to the traffic direction. For example, in most cases you want to exclude server-to-client attacks from rules protecting destination servers and to exclude client-to-server attacks from rules where you monitor server response.

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [Viewing CPU Utilization on page 586](#)
- [Attack Objects Task Summary on page 246](#)

The following related topics are included in the *IDP Series Concepts and Examples Guide*:

- [IDP Series Operating System Overview on page 7](#)
- [Using Attack Objects on page 60](#)

---

## Troubleshooting Erroneous CPU Utilization Reports

---

**Problem** Accurate measurement of CPU utilization for an operating system must be calibrated. IDP Series devices have model-specific default CPU utilization calibration values that are set when the **lkmstart.sh** utility is run. The **lkmstart.sh** utility is run automatically when the system is restarted. In IDP OS 5.1r2, the implementation for multicore CPU platforms includes a self-correction mechanism that recalibrates the CPU utilization measurement from time-to-time. Recalibration is initialized after traffic load falls to zero and then starts processing new traffic (the new load must be nontrivial).

For multicore platforms, we recommend you use SNMP to monitor CPU utilization and use the **scio idp-cpu-utilization** utility to investigate unexpected high or low CPU utilization. If the CPU utilization does not make sense given the traffic load, the CPU measurement might need to be recalibrated. Wait until traffic load subsides to zero and rises again to see if the self-correction mechanism handles the problem. If this does not work, you can use the **scio calibration** command to change the calibration values manually.



**NOTE:** The recalibration method described here is not applicable to single core platforms. The CPU utilization reported for single core platforms is the results of the Linux `top` command.

**Solution** While troubleshooting CPU utilization reporting, you use the `idpengine` log files to review the history of CPU utilization calibration values. If the current CPU utilization that is reported seems inaccurate and on the low side, you can try to correct by increasing the calibration count. If the CPU utilization % is inaccurate and on the high side, you can try to correct by decreasing the calibration count.

The following commands show how to find the CPU calibration entries in the logs:

```
[root@defaulthost ~]# cd /usr/idp/device/var/sysinfo/logs
[root@defaulthost logs]# ls -l
total 13548
-rw-r--r-- 1 idp idp 13922 Apr 29 17:04 agent.20110429
-rw--w---- 1 idp idp 2700 Apr 30 11:34 agent.20110430
-rw--w---- 1 idp idp 91 May 2 00:41 agent.20110502
-rw-r--r-- 1 root root 3474 Apr 29 03:40 idpengine_0.20110429
-rw-r----- 1 root root 3752123 Apr 30 11:34 idpengine_0.20110430
-rw-r--r-- 1 root root 1750667 May 2 01:42 idpengine_0.20110502
-rw-r--r-- 1 idp idp 795595 Apr 29 23:59 idpHMD.20110429
-rw--w---- 1 idp idp 925840 Apr 30 23:59 idpHMD.20110430
-rw--w---- 1 idp idp 916416 May 1 23:59 idpHMD.20110501
-rw--w---- 1 idp idp 426893 May 2 11:10 idpHMD.20110502
-rw-r--r-- 1 root root 504 Apr 29 03:41 idpinit.20110429
-rw-r----- 1 root root 649 Apr 30 10:51 idpinit.20110430
-rwxr-x--- 1 idp idp 12405 Apr 29 02:21 idpInstallLog.20110429021955
-rw-r--r-- 1 idp idp 5668 Apr 29 03:42 idpLogReader.20110429
-rw-r----- 1 idp idp 12282 Apr 30 12:25 idpLogReader.20110430
-rw-r--r-- 1 idp idp 4063 May 2 01:42 idpLogReader.20110502
-rw-r--r-- 1 root root 1038066 Apr 29 23:59 lkmStart.20110429
-rw-r----- 1 root root 1182731 Apr 30 23:59 lkmStart.20110430
-rw-r--r-- 1 root root 1173772 May 1 23:59 lkmStart.20110501
-rw-r--r-- 1 root root 546132 May 2 11:10 lkmStart.20110502
-rw-r--r-- 1 root root 1421 Apr 29 03:40 loadNics.20110429
-rw-r--r-- 1 root root 2986 Apr 29 03:42 migrateLog.20110430034003
-rw-r--r-- 1 root root 1230 Apr 30 10:51 nicBypass.20110429
-rw-r--r-- 1 root root 44 Apr 30 10:52 nicBypass.20110430
-rw-r--r-- 1 root root 156 Apr 29 23:59 peerPortModulator.20110429
-rw-r--r-- 1 root root 104 Apr 30 23:59 peerPortModulator.20110430
-rw-r--r-- 1 root root 0 May 1 23:59 peerPortModulator.20110501
-rw-r--r-- 1 root root 0 May 2 11:10 peerPortModulator.20110502
-rw-r--r-- 1 root root 6740 Apr 29 03:31 reimage-shar.20110429032917
-rw-r--r-- 1 idp idp 1048 Apr 29 03:41 scio.20110429
-rw-r--r-- 1 idp idp 739 Apr 30 10:51 scio.20110430
-rw-r--r-- 1 idp idp 489 May 2 11:00 scio.20110502
-rw-r--r-- 1 idp idp 1133737 Apr 29 03:43 sciiod.20110429
-rw-r----- 1 idp idp 18243 Apr 30 11:34 sciiod.20110430
-rw-r--r-- 1 idp idp 2094 May 2 00:43 sciiod.20110502
-rw-rw-rw- 1 idp idp 144 Apr 29 03:42 statview.20110429
```

You want to base your adjustments on the latest reported value, so start with the most recent log file and work backwards. In this example, `idpengine_0.20110502` has no entries for CPU calibration, so we examine the previous log file, `idpengine_0.20110430`.

```
[root@defaulthost ~]# viidengine_0.20110430
[10:51:40] The initial calibrated delay count is 193340708 in 1000063
micro-seconds
[10:51:40] resource_util_loop initied
[10:51:40] SRI: Device count : 10
[10:51:40] SRI: Total free packets : 508369
[10:51:40] Stats enabled : 31
[10:51:41] calibration_count updated from 110395859 to 141787094 and
calibration_time from 1000000 to 1000247
[10:51:47] calibration_count updated from 141787094 to 141787882 and
calibration_time from 1000247 to 1000199
[10:51:53] calibration_count updated from 141787882 to 141789474 and
calibration_time from 1000199 to 1000189
[10:52:11] calibration_count updated from 141789474 to 141790962 and
calibration_time from 1000189 to 1000197
[11:06:23] sc_emul_request_module:Trying to open dll.. path:./detector1304186783
name:detector1304186783 ...
```

If the calibration time is exactly 1000000 (microseconds), you can infer that the calibration values probably have not yet been recalibrated with the self-correction mechanism. In the above example, the latest calibration time is 1000197 which indicates that self-correction has run. After self-correction has run, the CPU utilization percent reported tends to be accurate. If self-correction has not been run, and you cannot wait for the occurrence of the traffic conditions that initiate self-correction, you can perform the following steps to initiate recalibration.

To set a calibration value with the **scio calibration** command:

1. Log into the CLI.
2. Use the **scio calibration get** command to show the current calibration values:

```
[root@defaulthost ~]# scio calibration get
CPU Calibration detail of CPU0:
Calibration count: 141790962
Calibration time : 1000197
```

3. Use **scio calibration set** command to set a new calibration count:

```
[root@defaulthost ~]# scio calibration set calibration-count
```

The manual calibration values remain in effect until the next opportunity for the self-correction mechanism to run.

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Viewing CPU Utilization on page 586](#)
- [Configuring SNMP Reporting for the Sensor Category of Statistics on page 435](#)
- [scio idp-cpu-utlization on page 525](#)
- [Troubleshooting High CPU Usage on page 588](#)

---

## Displaying Service Session Count

**Problem** You can use the command-line interface (CLI) to verify proper functioning of the protocol decoders without waiting for corresponding attack logs to be generated. The **scio subs**

**service detail** command displays the current active and total service session counters. The counters increment when the IDP engine identifies the application.

**Solution** To display the service session count:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to display the service session count:

```
[root@defaulthost admin]# scio subs service detail s0
Service Session Count Table:
| Service | Active | Total |
|-----+-----+-----|
| FTP     | 86     | 86     |
| RLOGIN  | 21     | 21     |
| PORTMAPPER | 100    | 100    |
| HTTP    | 730    | 730    |
| SMTP    | 38     | 38     |
| POP3    | 76     | 76     |
| IMAP    | 10     | 10     |
| TELNET  | 52     | 52     |
| ICMP    | 116    | 116    |
| DNS     | 50     | 50     |
| SSH     | 1      | 1      |
| SNMP    | 11     | 11     |
| DHCP    | 17     | 17     |
| TFTP    | 21     | 21     |
```

**Related Documentation** • [scio subs on page 535](#)

## Troubleshooting Configuration Push Errors (NSM Procedure)

**Problem** [Table 168 on page 593](#) provides tips for troubleshooting errors related to NSM configuration push jobs.

**Table 168: Troubleshooting: Configuration Push Errors**

Error	Description
Timeout	<p>The default timeout for IDP security policy is 2,400,000 milliseconds (40 minutes).</p> <p>When you first push a policy to a newly deployed IDP Series device, NSM must send a lot of information (mostly attack definitions). In some cases, the update job can time out before it completes.</p> <p>To modify the timeout setting:</p> <ol style="list-style-type: none"> <li>1. On the NSM Device Server, open the following file in a text editor:           <pre>/usr/netscreen/DevSvr/var/devSvr.cfg</pre> </li> <li>2. Modify the following setting:           <pre>devSvrDirectiveHandler.idpPolicyPush.timeout 2400000</pre> </li> </ol>

Table 168: Troubleshooting: Configuration Push Errors (*continued*)

Error	Description
The following attacks/groups cannot be updated. Not supported for version.	<p>Different IDP platforms use different detector engines. Not all attack objects are valid for all versions of the detector engine. This message indicates which attack objects in the security policy were not valid for the loaded detector engine and, therefore, not loaded.</p> <p>This message is for information purposes only and does not indicate a problem with the device or the policy.</p>
No firewall rules can be updated for device in assigned policy policyName.	<p>You try to load a policy that contains a firewall rulebase onto a IDP Series device.</p> <p>This message means that the IDP Series device cannot process the firewall rulebase. The IDP security policy rulebases are still processed normally, assuming no other errors.</p>
Rule #: Packet logging with any/any rule has serious performance implications.	Setting the rule to log packets causes the device to save packets until it is sure that they will not be needed for a log entry. A rule that has any in the Source IP column and any in the Destination IP column examines all traffic. So, the device has to save a lot of packets all the time, which impacts performance.
Policy has not changed and hence will not be updated.	For performance reasons, the device does not spend resources recompiling a security policy that has not changed.
Failed to update device. Failed to compile policy.	Something has gone wrong with the policy compilation. Other error messages might indicate why.
No license for idp.	The device does not have a valid license. Unlicensed devices do not accept policy uploads.

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Pushing Security Policy Updates to an IDP Series Device \(NSM Procedure\)](#) on page 340

## Troubleshooting Security Policy Validation Errors (NSM Procedure)

**Problem** Table 169 on page 594 describes security policy validation errors and how to resolve them.

Table 169: Troubleshooting: Security Policy Validation Errors

Error	Description
Rule duplication	<p>Rule appears more than once.</p> <p>To resolve this problem, delete the duplicate.</p>
Rule shadowing	<p>Rule shadowing occurs when two rules are designed to detect the same attack, and the first rule is either a terminal match rule or contains a more severe action than the second rule. In these cases, the second rule will never be applied.</p> <p>To resolve this problem, modify or delete one of the rules.</p>

Table 169: Troubleshooting: Security Policy Validation Errors (*continued*)

Error	Description
Protocol mismatches	<p>Protocol mismatches occur when a service object that is specified in the Service column of the security policy uses a different protocol from that specified by the default service binding of the attack object for that rule. Remember that the service binding specifies the service and port that the attack uses. Because two different protocols are specified, the IDP engine cannot match attacks for the attack object.</p> <p>To resolve this problem, set Service to <b>Default</b>.</p>
Any-Any-None rules	<p>Look everywhere for nothing: any source, any destination, and no attacks. This rule can cause severe performance penalties.</p> <p>To resolve this problem, specify network objects for the destination and attack objects for the attacks.</p>
Any-Any-One rules	<p>Look everywhere for one thing: any source, any destination and one attack object. This rule can cause severe performance penalties.</p> <p>To resolve this problem, specify network objects for the destination.</p>
Unsupported options	<p>Rule contains options that are not supported on the target device.</p> <p>To resolve this problem, upgrade the target device or remove the option from the rule.</p>

**Related Documentation**

The following related topic is included in the *IDP Series Administration Guide*:

- [Validating a Security Policy \(NSM Procedure\) on page 336](#)

## Troubleshooting Application Identification

**Problem** If you encounter issues with the application identification feature, you might want to change the default behavior. The following features are enabled by default:

- Application identification
- Application identification of extended applications
- Caching of application identification matches
- Caching of extended application identification matches

For normal use, we recommend that you maintain these defaults.

**Solution** During experimentation and troubleshooting, you might do the following:

- View details of the application signatures that are relevant to the current policy. The application signatures are relevant if the application is specified or implicated by IDP rulebase or APE rulebase rules.
- View a list of application signatures cached. The application identification feature caches signatures it has detected.

- Clear the application signature or nested application signature cache.
- Disable application identification or the application identification cache.

To display the application signatures and nested application signatures that are relevant to the current policy:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to display a list of relevant application signatures:

```
[root@defaulthost ~]# scio app sig list
Application signatures: total 165 show 165
APPLICATIONID:ARES, index 0, service 77, mindata 7, order 61, tcp 0-65535, no
udp
APPLICATIONID:ICCP, index 1, service 106, mindata 2, order 147, tcp 102-102,
no udp
APPLICATIONID:HALF-LIFE, index 2, service 113, mindata 4, order 127, no tcp,
udp 1024-65535
APPLICATIONID:ICQ, index 3, service 172, mindata 10, order 22, tcp 0-65535,
no udp
APPLICATIONID:PPTP, index 4, service 128, mindata 11, order 173, tcp 1723-1723,
no udp
APPLICATIONID:X11, index 5, service 144, mindata 6, order 85, tcp 0-65535, no
udp
APPLICATIONID:GNUTELLA-CONNECT, index 6, service 151, mindata 16, order 40,
tcp 0-65535, no udp
APPLICATIONID:VMWARE-WEBUI, index 7, service 96, mindata 180, order 183, tcp
8333-8333, no udp
APPLICATIONID:FREecast, index 8, service 81, mindata 50, order 142, no tcp,
udp 0-65535
APPLICATIONID:MSRPC, index 9, service 55, mindata 20, order 42, tcp 135-135
137-139 445-445 1024-65535, udp 135-135 137-139 445-445 1024-65535
APPLICATIONID:MSN, index 10, service 41, mindata 20, order 107, tcp 0-65535,
no udp
APPLICATIONID:DRDA, index 11, service 121, mindata 20, order 52, tcp 0-65535,
no udp
APPLICATIONID:GNUTELLA-URN-DOWNLOAD, index 12, service 131, mindata 26, order
133, tcp 0-65535, no udp
APPLICATIONID:GNUTELLA-FIREWALLED, index 13, service 86, mindata 70, order 34,
tcp 0-65535, no udp
APPLICATIONID:IRC, index 14, service 32, mindata 32, order 46, tcp 0-65535,
no udp
APPLICATIONID:QQ, index 15, service 125, mindata 3, order 105, tcp 80-80
443-443, udp 0-65535
APPLICATIONID:LOTUSNOTES, index 16, service 135, mindata 21, order 134, tcp
0-65535, no udp
```

3. Enter the following command to display a list of relevant nested application signatures:

```
[root@defaulthost ~]# scio napp sig list
Nested Application signatures: total 551 show 551
NESTEDAPPLICATION:PRICELINE, HTTP, index 0, nested service 1, max_trans 1,
order 33249, appl_id 794, n_members 1
NESTEDAPPLICATION:ICAST, HTTP, index 1, nested service 2, max_trans 1, order
33118, appl_id 555, n_members 2
NESTEDAPPLICATION:GOOGLE-TRANSLATE, HTTP, index 2, nested service 3, max_trans
1, order 32991, appl_id 467, n_members 1
NESTEDAPPLICATION:EBUDDY, HTTP, index 3, nested service 4, max_trans 1, order
32906, appl_id 278, n_members 1
NESTEDAPPLICATION:TOPFRIENDS, HTTP, index 4, nested service 5, max_trans 1,
```



```

order 33299, appl_id 723, n_members 2
NESTEDAPPLICATION:MYSpace-GUARDIAN-ANGELS, HTTP, index 5, nested service 6,
max_trans 1, order 33169, appl_id 619, n_members 2
NESTEDAPPLICATION:ALLMUSIC-LOOKUP, HTTP, index 6, nested service 7, max_trans
1, order 33043, appl_id 530, n_members 2
NESTEDAPPLICATION:HOTMAIL, HTTP, index 7, nested service 8, max_trans 1, order
32832, appl_id 383, n_members 1
NESTEDAPPLICATION:RAGINGBULL-POST, HTTP, index 8, nested service 9, max_trans
1, order 32963, appl_id 354, n_members 2
NESTEDAPPLICATION:FACEBOOK-VISUALBOOKSHELF, HTTP, index 9, nested service 10,
max_trans 1, order 33227, appl_id 602, n_members 2
NESTEDAPPLICATION:TRIPADVISOR, HTTP, index 10, nested service 11, max_trans
1, order 33099, appl_id 579, n_members 1
NESTEDAPPLICATION:SPANKWIRE, HTTP, index 11, nested service 12, max_trans 1,
order 32816, appl_id 505, n_members 1
NESTEDAPPLICATION:THECIRCLE, HTTP, index 12, nested service 13, max_trans 1,
order 32888, appl_id 261, n_members 1

...

```

To display the application signature and the nested application signature cache:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to display the application signature cache:

```

[root@defaulthost ~]# scio app cache list
Application system cache: total 3 show 3

```

Index	VLAN	IP	Port	Proto	Application
0	0	9.0.0.101	21	6	FTP
1	0	8.0.0.101	21	6	FTP
2	0	8.0.0.1 22	6	SSH	

3. Enter the following command to display the nested application signature cache:

```

[root@defaulthost ~]# scio napp cache list
Application system cache: total 3 show 3

```

Index	VLAN	IP	Port	Proto	Application
0	0	9.0.0.101	21	6	FTP
1	0	8.0.0.101	21	6	FTP
2	0	8.0.0.1 22	6	SSH	

To clear application the signature and the nested application signature cache:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
  2. Enter the following command to clear the application signature cache:
- ```
[root@defaulthost ~]# scio app cache clear
```
3. Enter the following command to clear the nested application signature cache:

```
[root@defaulthost ~]# scio napp cache clear
```

To disable application identification features:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
  2. Enter the following command to disable application protocol identification:
- ```
[root@defaulthost ~]# scio const set sc_ai_enable 0
```
3. Enter the following command to disable extended application identification:

```
[root@defaulthost ~]# scio const set sc_ai_ext_enable 0
```

4. Enter the following command to disable caching of application protocol identification results:

```
[root@defaulthost ~]# scio const set sc_asc_enable 0
```

5. Enter the following command to disable caching of extended application protocol identification results:

```
[root@defaulthost ~]# scio const set sc_ext_asc_enable 0
```

Changes take effect immediately, but the settings do not persist across restarts and policy pushes.

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [scio app sig list on page 504](#)
- [scio napp sig list on page 527](#)
- [scio app cache on page 502](#)
- [scio const on page 505](#)
- [Application Objects Task Summary on page 286](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Using Application Objects on page 73](#)

---

## Disabling the APE Rulebase

---

**Problem** If the APE rulebase does not behave as expected, you can use the IDP OS CLI to disable it.

**Solution** To disable the APE rulebase:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Use the following command to show the current value:

```
[root@defaulthost admin]# scio const get sc_ape_enable  
scio: sc_ape_enable = 0x1
```

A 1 (0x1) indicates the APE is enabled; a 0 (0x0) indicates the APE rulebase is disabled.

3. Change this setting with the corresponding **set** command. The following example disables the APE rulebase:

```
[root@defaulthost admin]# scio const set sc_ape_enable 0  
scio: setting sc_ape_enable to 0x0
```

**Related Documentation**

The following related topics are included in the *IDP Series Administration Guide*:

- [Configuring the APE Rulebase \(NSM Procedure\) on page 228](#)
- [Verifying the APE Rulebase on page 491](#)

## Disabling the User Role-Based Policy Feature

**Problem** If the user role-based policy feature does not behave as expected, you can use the IDP OS command-line interface (CLI) to disable it.

**Solution** To disable the user role-based policy feature:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following command to show the current value:

```
[root@defaultthost admin]# scio const -s s0 get sc_enable_user_policy
scio: sc_enable_user_policy = 0x1
```

A 1 (0x1) indicates the feature is enabled; a 0 (0x0) indicates the feature is disabled.

3. Enter the following command to disable the feature:

```
[root@defaultthost admin]# scio const -s s0 set sc_enable_user_policy 0
scio: setting sc_enable_user_policy to 0x0
```

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying Integration with an IC Series Unified Access Control Appliance on page 493](#)
- [scio const on page 505](#)

## Disabling Support for Jumbo Frames

By default, the IDP engine processes traffic as large as the maximum transmission unit (MTU) required to support jumbo frames (9014 bytes = 9000 bytes of payload + 14 bytes for the Ethernet header). The IDP engine drops frames that are larger than 9014 bytes.

If you encounter issues processing jumbo frames, you can change the default maximum frame size to what is required to support the Ethernet MTU (1514 bytes = 1500 bytes of payload + 14 bytes for the Ethernet header). The greatest value supported for maximum frame size is 16,014 bytes.

To change the MTU:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following commands to view and change the MTU:

```
[root@defaultthost admin]# scio const -s s0 get sc_max_frame_size
scio: sc_max_frame_size = 0x2336
```

```
[root@defaultthost admin]# scio const -s s0 set sc_max_frame_size 1514
scio: sc_max_frame_size = 0x5EA
```

**Related Documentation** The following related topic is included in the *IDP Series Administration Guide*:

- [scio const on page 505](#)

## Troubleshooting SSL Inspection

---

**Problem** This topic lists a few areas to investigate if the IDP Series device hangs during HTTPS inspection or fails to inspect HTTPS traffic as expected.

**Solution** Investigate the following issues:

- In case of HTTPS traffic hanging in a laboratory environment, be sure your test traffic includes “background” traffic in addition to the HTTPS sessions. Background traffic can be a simple ping across the IDP Series device. In a production environment, this is not an issue.
- If the IDP Series device fails to detect a specified HTTP anomaly, examine the security policy to make sure it includes at least one SSL attack object. We recommend SSL: SERVER-CERT-FAILS-VALIDATION or any Recommended SSL attack object.

**Related Documentation** The following related topics are included in the *IDP Series Administration Guide*:

- [Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic on page 311](#)
- [Disabling SSL Inspection on page 600](#)
- [scio const on page 505](#)
- [scio ssl on page 531](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of SSL Traffic Overview on page 113](#)

## Disabling SSL Inspection

---

**Problem** If necessary, you can disable SSL inspection so that HTTPS sessions are passed through the IDP Series device uninspected. If you would rather drop such sessions, you must create a security policy rule that matches the HTTPS traffic and uses the drop action.

**Solution** Follow the procedure indicated to disable the particular SSL inspection method:

- Inspection using the internal server private key
- Inspection using the forward proxy feature

To disable the method that uses the internal server private key:

1. Log into the CLI as **admin** and enter **su** - to switch to root.
2. Enter the following command to disable decryption:

```
[root@default host admin]# scio const -s s0 set sc_ssl_decryption 0
scio: setting sc_ssl_decryption to 0x0
```



**TIP:** To make your setting persistent across restarts, modify the `user_funcs` file; or modify the setting in NSM and push the update to the IDP Series device.

To disable inspection using the forward proxy feature:

1. Log into the CLI as **admin** and enter **su -** to switch to root.
2. Delete the certificate authority:

```
[root@defaulthost admin]# scio ssl ca delete
[root@defaulthost admin]#
```

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Using the SSL Private Server Key to Enable Inspection of SSL Traffic on page 308](#)
- [Using the SSL Forward Proxy Feature to Enable Inspection of HTTPS Traffic on page 311](#)
- [scio const on page 505](#)
- [scio ssl on page 531](#)

The following related topic is included in the *IDP Series Concepts and Examples Guide*:

- [Inspection of SSL Traffic Overview on page 113](#)

## Disabling MPLS Decapsulation

If you encounter issues with the Multiprotocol Label Switching (MPLS) decapsulation feature, you can disable it.

To disable MPLS decapsulation:

1. Log into the CLI as **admin** and enter **su -** to switch to **root**.
2. Enter the following commands to disable MPLS decapsulation and verify your settings:

```
[root@defaulthost admin]# scio const -s s0 set sc_mpls_decapsulation 0
scio: sc_mpls_decapsulation 0x0

[root@defaulthost admin]# scio const -s s0 get sc_mpls_decapsulation
scio: sc_mpls_decapsulation = 0x0
```

#### Related Documentation

The following related topics are included in the *IDP Series Administration Guide*:

- [Verifying MPLS Decapsulation on page 494](#)
- [scio const on page 505](#)



## PART 7

# Index

- [Index on page 605](#)





# Index

## Symbols

#, comments in configuration statements.....	xxvii
( ), in syntax descriptions.....	xxvii
< >, in syntax descriptions.....	xxvi
[ ], in configuration statements.....	xxvii
{ }, in configuration statements.....	xxvii
(pipe), in syntax descriptions.....	xxvii

## A

ACM.....	189
ACM reference.....	191
actions.....	63
IDP rulebase.....	218
IP actions.....	220
Recommended.....	219
activating devices in NSM.....	348
adding a device to NSM.....	344
agent process.....	10
antispoof settings.....	353
APE rulebase.....	69, 228
troubleshooting.....	598
verifying implementation of.....	491
application groups	
viewing .....	292
application identification.....	43
application objects	
task list.....	286
viewing predefined.....	287
application policy enforcement.....	69
application volume tracking.....	22
enabling in NSM.....	205
preference settings.....	210
reports.....	471
viewing logs in Profiler Viewer.....	463
attack logs, viewing.....	454
attack objects.....	60, 243
creating compound.....	273
creating signature.....	253
groups.....	251
specifying in Exempt rulebase.....	228
specifying in IDP rulebase rules.....	216

task list.....	246
updating predefined.....	336, 392
viewing predefined.....	247
attack reports.....	469
Audit Log Viewer.....	460
auto-recovery.....	9
autorecovery	
disabling.....	584
autorecovery logs.....	583

## B

backdoor rulebase.....	85, 233
bandwidth rate limiting.....	228
Boolean expressions in compound attack	
objects.....	276
braces, in configuration statements.....	xxvii
brackets	
angle, in syntax descriptions.....	xxvi
square, in configuration statements.....	xxvii
bypassStatus reference.....	484
bypassStatus utility.....	483

## C

CLI.....	189, 307, 411, 491
clusters, adding to NSM.....	347
comments, in configuration statements.....	xxvii
compound attack objects	
Boolean expressions.....	276
creating.....	273, 274
protocol anomalies.....	276
compound attack objects, creating.....	273, 274
configuration updates.....	350
context check constraint.....	262
context, specifying.....	266
control plane.....	7
conventions	
text and syntax.....	xxvi
CPU usage.....	586
curly braces, in configuration statements.....	xxvii
custom application objects	
viewing.....	292
customer support.....	xxvii
contacting JTAC.....	xxvii

## D

data plane.....	7
DDoS attack prevention.....	97
default configuration.....	194
detector engine, updating.....	336, 392

device status.....	448
DFA expressions	
syntax.....	263
direction	
specifying.....	266
disabling rules.....	342
documentation	
comments on.....	xxvii
DoS attack prevention.....	91
dynamic groups.....	251

## E

eth0 interface.....	13
eth1 interface.....	13
eth2 interface.....	13
Exempt rulebase.....	67, 227
extended application identification See application identification	
extended application objects	
viewing .....	290
viewing predefined.....	292
external bypass.....	15
external bypass, setting.....	192, 401

## F

fail close.....	14
false positives.....	50, 162
feature list.....	3
flow bypass.....	10
enabling.....	319, 402
verifying implementation of.....	495
flow table	
resetting.....	360
viewing.....	487
flow, specifying.....	266
font conventions.....	xxvi
fragment handling.....	362

## G

GRE inspection.....	313
GRE traffic.....	111
GTP inspection.....	315
GTP traffic.....	111
guiSvrCli.sh.....	339, 395

## H

HA interface.....	13
HTTPS inspection.....	308, 311

## I

I/O modules.....	397
IC Series Appliance	
verifying integration with.....	493
ICMP packet header matching.....	267
IDP detector engine, updating.....	336, 392
IDP Reporter.....	31, 411
IDP rulebase.....	55, 213
idp.sh.....	385
idp.sh reference.....	385
idpengine process.....	10
idpHMD process.....	10
idpLogReader process.....	10
interface aliasing.....	296
internal bypass.....	14
internal bypass, setting.....	192, 401
IP Action	
resetting the block table.....	361
IP actions.....	220
IP packet header matching.....	267
IP spoof attack prevention.....	109
IPsec ESP NULL traffic.....	112
IPsec VPN inspection.....	317

## J

J-Security Center.....	19, 336, 392
JNET driver.....	7
jnetTcpdump.....	481
jumbo frames, disabling.....	599

## L

Log Investigator.....	459
log storage limits.....	297
log suppression.....	298
logging.....	65
logs.....	24, 295, 447

## M

management interface.....	13
manuals	
comments on.....	xxvii
MIB.....	553
mixed deployment mode, setting.....	192, 401
MPLS inspection	
disabling.....	601
enabling.....	318
verifying implementation of.....	494
MPLS traffic.....	112
multicore architecture.....	7

**N**

nested application identification See application identification

Network and Security Manager See NSM overview

Network Honeypot rulebase.....103, 240

NIC cards.....397

notification options.....221

NSM.....189, 411

- custom reports.....472
- managing device configuration with .....343
- modifying device configuration with.....351
- updating IDP detector engine with.....336, 392
- updating predefined attack objects with.....336, 392
- updating software with.....390
- viewing logs and reports with.....295, 447
- viewing logs with.....453
- viewing reports with.....469

NSM Audit Log Viewer.....460

NSM Log Investigator.....459

NSM Log Viewer.....454

NSM overview.....18

NSM reports.....29, 469

**O**

OS fingerprinting

- enabling.....205

**P**

packet header matching.....267

packet logging.....145, 475

- enabling collection in NSM logs.....303
- enabling in IDP rulebase rules.....221
- log storage limits.....297
- using jnetTcpdump.....481
- using tcpdump.....481

parameters.....365

parentheses, in syntax descriptions.....xxvii

peer port modulation.....16

peer port modulation, setting.....192, 401

pkid process.....10

policies.....41

PPM.....16

Profiler.....21

- alerts.....209
- Application Profiler tab.....134, 463
- database settings.....210
- managing the database.....328
- Network Profiler tab.....466

- options.....204
- process.....11
- Protocol Profiler tab.....465
- reports.....30, 471
- starting and stopping.....328
- using.....203, 327
- Violation Viewer tab.....468

protocol anomalies in compound attack

- objects.....276

protocol decoders.....592

protocol handling parameters.....367

pulling configuration updates.....350

pushing configuration updates.....350

**R**

reboot command.....384

Recommended action.....63, 156, 219

Recommended attack.....155

Recommended attack objects.....216, 252

Recommended policy.....46, 49, 160

Recommended security policy.....197

reports.....295, 447

rule match conditions.....215

rule severity.....224

rule target.....223

runtime parameters.....355

**S**

scio

- agentconfig.....500
- ca.....544
- const.....505
- getsystem.....524
- nic.....528
- ssl.....531
- subs.....535
- sysconf.....539
- vc.....549
- version.....550
- vr.....551

scio utility.....307, 491

sciod process.....11

sctop.....487

security policies.....41

- assigning.....335
- creating.....195
- exporting.....342
- pushing to IDP Series devices.....340
- validating.....336

service binding.....	260
severity, rule.....	224
shutdown command.....	384
signature attack objects	
creating.....	253
signature attack objects, creating.....	253
sniffer mode, setting.....	192, 401
SNMP.....	299, 411
SSL inspection.....	308, 311
static groups.....	252
support, technical See technical support	
SYN Protector rulebase.....	91, 235
configurable thresholds.....	237, 363
syntax conventions.....	xxvi
syslog.....	301

## T

TCP packet header matching.....	267
tcpdump.....	481
tech-support script.....	575
technical support	
contacting JTAC.....	xxvii
tech-support script.....	575
templates, security policy.....	197
terminal match.....	215
time binding.....	261
Traffic Anomalies rulebase.....	97, 237
traffic interfaces.....	13
I/O modules.....	397
transparent mode.....	33
transparent mode, setting.....	192, 401
troubleshooting utilities.....	575

## U

UDP packet header matching.....	267
updating software	
CLI procedure.....	389
NSM procedure.....	390
updating the configuration.....	350
user-role-based policy.....	58
advanced features.....	322
troubleshooting.....	599
verifying implementation of.....	493

## V

virtual router.....	13
virtual routers, configuring.....	192, 401
VLAN	
match in security policy rules.....	223

## W

within bytes constraint.....	262
within packets constraint.....	262