JUNIPER NETWORKS | Engineering Simplicity

# Junos® OS

# Authentication and Integrated User Firewalls Feature Guide

Published
2019-12-24

JUNOS

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, 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.

*Junos® OS Authentication and Integrated User Firewalls Feature Guide*
Copyright © 2019 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 https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

**About the Documentation | xxi**

Documentation and Release Notes | **xxi**

Using the Examples in This Manual | **xxi**

- Merging a Full Example | **xxii**
- Merging a Snippet | **xxiii**

Documentation Conventions | **xxiii**

Documentation Feedback | **xxvi**

Requesting Technical Support | **xxvi**

- Self-Help Online Tools and Resources | **xxvii**
- Creating a Service Request with JTAC | **xxvii**

1 **Overview**

**Understanding User Authentication for Security Devices | 31**

2 **Firewall User Authentication**

**Firewall User Authentication Overview | 35**

**Configure Client Groups | 37**

- Understanding Client Groups for Firewall Authentication | **37**
- Example: Configuring Local Users for Client Groups | **38**

**Customize the Firewall Authentication Banner | 41**

- Understanding Firewall Authentication Banner Customization | **41**
- Example: Customizing a Firewall Authentication Banner | **42**

**Configure External Authentication Servers | 44**

- Understanding External Authentication Servers | **44**
  - Understanding SecurID User Authentication | **45**
- Example: Configuring RADIUS and LDAP User Authentication | **46**
- Enabling LDAP Authentication with TLS/SSL for Secure Connections | **51**
- Example: Configuring SecurID User Authentication | **53**
- Example: Deleting the SecurID Node Secret File | **57**

**Configure User Authentication Methods | 59**

Understanding Pass-Through Authentication | **59**

Example: Configuring Pass-Through Authentication | **61**

Example: Configuring HTTPS Traffic to Trigger Pass-Through Authentication | **70**

Understanding Web Authentication | **78**

Example: Configuring Web Authentication | **80**

Example: Configuring HTTPS Traffic to Trigger Web Authentication | **89**

**Encrypt Traffic Using SSL Proxy and TLS | 95**

SSL Proxy Overview | **96**

Benefits of SSL Proxy | **97**

Logical Systems Support | **98**

Limitations | **98**

Configuring SSL Forward Proxy | **99**

SSL Proxy Configuration Overview | **99**

Configuring a Root CA Certificate | **100**

Generate a Root CA Certificate with CLI | **101**

Generate a Root CA Certificate with OpenSSL | **102**

Configuring a CA Profile Group | **103**

Importing a Root CA Certificate into a Browser | **105**

Applying an SSL Proxy Profile to a Security Policy | **106**

Configuring SSL Proxy Logging | **107**

Configuring Certificate Authority Profiles | **108**

Exporting Certificates to a Specified Location | **109**

Ignoring Server Authentication | **110**

Enabling Debugging and Tracing for SSL Proxy | **110**

Transport Layer Security (TLS) Overview | **112**

Benefits of TLS | **113**

TLS Versions | **113**

Three Essential Services of TLS | **113**

TLS Handshake | **114**

Encrypting Syslog Traffic with TLS | **114**

Configuring the TLS Syslog Protocol on SRX Series device | **114**

3

## Unified Access Control with IC Series UAC Appliance


**Configure Unified Access Control in Junos OS | 121**

Understanding UAC in a Junos OS Environment | **121**

Enabling UAC in a Junos OS Environment (CLI Procedure) | **123**

**Set Up Communication between Junos OS Enforcer and IC Series UAC Appliance | 124**

Understanding Communications Between the Junos OS Enforcer and the IC Series UAC Appliance | **124**

Understanding Communications Between Junos OS Enforcer and a Cluster of IC Series UAC Appliances | **125**

Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure) | **126**

Understanding Junos OS Enforcer Implementations Using IPsec | **128**

Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI) | **129**

**Enforce Policies and Configure Endpoint Security with Junos OS Enforcer | 138**

Understanding Junos OS Enforcer Policy Enforcement | **139**

Configuring Junos OS Enforcer Failover Options (CLI Procedure) | **140**

Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) | **141**

Verifying Junos OS Enforcer Policy Enforcement | **141**

Displaying IC Series UAC Appliance Authentication Table Entries from the Junos OS Enforcer | **142**

Displaying IC Series UAC Appliance Resource Access Policies from the Junos OS Enforcer | **142**

Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer | **142**

Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer | **143**

**Configure Captive Portal on Junos OS Enforcer | 143**

Understanding the Captive Portal on the Junos OS Enforcer | **144**

Understanding Captive Portal Configuration on the Junos OS Enforcer | **146**

Understanding the Captive Portal Redirect URL Options | **146**

Example: Creating a Captive Portal Policy on the Junos OS Enforcer | **148**

**Classify Traffic Based on User Roles | 151**

Understanding Unified Access Control | **151**

Acquiring User Role Information from an Active Directory Authentication Server | **152**

Obtaining Username and Role Information Through Firewall Authentication | **172**

4

## Integrated User Firewall

**Integrated User Firewall Overview | 177**

Overview of Integrated User Firewall | **177**

Integrated User Firewall and Authentication Sources | **177**

Benefits of Integrated User Firewall | **178**

How the Integrated User Firewall Works | **178**

Deployment Scenario for User Firewall Integration with Windows Active Directory | **179**

Limitations | **181**

Understanding Active Directory Authentication Tables | **182**

Active Directory Authentication as an Authentication Source | **182**

Active Directory Authentication Tables | **182**

State Information for Active Directory Authentication Table Entries | **185**

Active Directory Authentication Table Management | **186**

Timeout Interval for Table Entries | **188**

Understanding the Invalid Authentication Table Entry Timeout Setting | **190**

Timeout Setting for Invalid Authentication Entries | **190**

How the Invalid Authentication Entry Timeout Works for Windows Active Directory | **191**

How the Invalid Authentication Entry Timeout Works for SRX Series and NFX Series Aruba ClearPass | **192**

LDAP Functionality in Integrated User Firewall | **194**

Role of LDAP in Integrated User Firewall | **194**

LDAP Server Configuration and Base Distinguished Name | **194**

LDAP's Authentication Method | **195**

LDAP Server's Username, Password, and Server Address | **195**

Caching and Calculation of User-to-Group Mappings | **195**

Updating Group Information in the Authentication Entry Table | **196**

LDAP Server Status and Statistics | **196**

Active Directory Autodiscovery | **196**

**Configure Integrated User Firewall | 196**

Example: Configuring Integrated User Firewall on SRX Series | **197**

Configuring Integrated User Firewall on NFX Devices | **209**

Example: Configuring Integrated User Firewall on SRX Series devices to Use Web-Redirect for Unauthenticated and Unknown Users | **211**

Example: Configuring Integrated User Firewall on SRX Series devices to Use Web-Redirect-to-HTTPS to Authenticate Unauthenticated and Unknown Users | **216**

**Configure Captive Portal for Unauthenticated Browsers | 223**

Understanding SRX Series Assured Captive Portal Support for Unauthenticated Browser Users | **223**

Understanding the Forced Timeout Setting Assigned to Active Directory Authentication Entries for Users Authenticated Through Captive Portal | **226**

**Manage Event logs to Generate IP Address-to-User Mapping | 227**

Understanding How the WMIC Reads the Event Log on the Domain Controller | **228**

Windows Management Instrumentation Client | **228**

WMIC Reads the Event Log on the Domain Controller | **229**

Specifying IP Filters to Limit IP-to-User Mapping | **229**

Event Log Verification and Statistics | **230**

Using Firewall Authentication as an Alternative to WMIC | **230**

WMIC Limitations | **230**

Firewall Authentication as a Backup Method for IP Address-to-User Mappings | **231**

Understanding Integrated User Firewall Domain PC Probing | **231**

Overview of Domain PC Probing | **232**

Probing Domain PCs for User Information | **232**

Probe Response | **233**

Probe Configuration | **234**

Probe Rate and Statistics | **234**

**Logging User Identity Information Based on Zones | 235**

Understanding How to Include User Identity Information in the Session Log File Based on the Source Zone | **235**

Example: Configuring Integrated User Firewall to Write User Identity to the Session Log Based On the Source Zone | **237**

**Control Network Access Using Device Identity Authentication | 243**

Understanding Access Control to Network Resources Based on Device Identity Information | **244**

Why Use Device Identity Information to Control Access to Your Network | **244**

Background | **244**

Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature | **247**

Device Identity | **247**

Device Identity Profile Contents | **248**

Predefined Device Identity Attributes | **250**

Characteristics of Device Identity Profiles, and Attributes and Target Scaling | **250**

Understanding the Device Identity Authentication Table and Its Entries | **252**

The Device Identity Authentication Table | **252**

Why the Device Identity Authentication Table Content Changes | **253**

Security Policy Matching and Device Identity Profiles | **256**

Understanding How the SRX Series Obtains the Authenticated Device Identity Information From Windows Active Directory for Network Access Control | **257**

Understanding the Device Identity XML Solution for Third-Party NAC Authentication Systems | **258**

XML Web API Implementation on SRX Series and NFX Series Devices | **259**

Ensuring the Integrity of Data Sent from the NAC Service to the SRX Series or NFX Series Device | **259**

Data Size Restrictions and Other Constraints | **259**

Example: Configuring the SRX Series Device Identity Feature in an Active Directory Environment | **260**

5 **Identity Management User Firewall**

**Configure Juniper Identity Management Service to Obtain User Identity Information | 275**

Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | **275**

Overview | **275**

Establishing a Connection to JIMS to Obtain User Identity Information | **276**

Querying JIMS for User Identity Information | **277**

Filters | **278**

Caveats and Limitations | 279

Configuring Advanced Query Feature for Obtaining User Identity Information from JIMS | 280

Configuring the Web API Process | 280

Configuring the Advanced Query Feature for Obtaining User Identity Information from JIMS | 281

Configuring Device Identity Authentication Source, and Security Policy to Match the User Identity Information Obtained from JIMS | 283

Example: Configuring the Advanced Query Feature for Obtaining User Identity Information from JIMS | 285

Example: Configuring Filter for Advanced Query Feature | 291

6 User Authentication and Enforcement with Clearpass

Integrated ClearPass Authentication and Enforcement Overview | 299

Understanding the Integrated ClearPass Authentication and Enforcement Feature | 299

Why You Need to Protect Your Environment With the Integrated ClearPass Authentication and Enforcement Feature | 300

How the Integrated ClearPass Authentication and Enforcement Feature Can Protect Your Network Environment | 300

Understanding the Invalid Authentication Table Entry Timeout Setting | 302

Timeout Setting for Invalid Authentication Entries | 302

How the Invalid Authentication Entry Timeout Works for Windows Active Directory | 303

How the Invalid Authentication Entry Timeout Works for SRX Series and NFX Series Aruba ClearPass | 304

Configure Integrated ClearPass Authentication and Enforcement | 306

Understanding How ClearPass Initiates a Session and Communicates User Authentication Information Using the Web API | 306

Web API | 307

ClearPass Authentication Table | 307

Using HTTPS or HTTP for the Connection Protocol Between ClearPass and the Device | 308

Ensuring the Integrity of Data Sent from ClearPass to the Device | 308

Data Size Restrictions and Other Constraints | 308

Posture States and the Posture Group | 309

Example: Configuring the SRX Series Integrated ClearPass Feature to Allow the Device to Receive User Authentication Data from ClearPass | 310

Understanding the Integrated ClearPass Authentication and Enforcement User Query Function | 321

Example: Configuring the Integrated ClearPass Authentication and Enforcement User Query Function | 324

**Enforce Security Policies using ClearPass | 334**

Understanding Enforcement of ClearPass User and Group Authentication | 335

Understanding How the Device Manages the ClearPass Authentication Table | 335

User Authentication Entries in the ClearPass Authentication Table | 336

Communication Between ClearPass and the SRX Series or NFX Series Device | 339

Understanding Domains and Interested Groups | 342

When a User Has Already Been Authenticated By Another Source | 345

Example: Enforcing SRX Series Security Policies Using Aruba ClearPass as the Authentication Source | 345

**Filter and Transmit Threat and Attack Logs to ClearPass | 366**

Understanding How the Integrated ClearPass Feature Detects Threats and Attacks and Notifies the CPPM | 367

SRX Series Threat and Attack Logs Sent to Aruba ClearPass | 369

Example: Configuring Integrated ClearPass to Filter and Rate-limit Threat and Attack Logs | 371

**Configure ClearPass and JIMS at the Same Time | 376**

Understanding How ClearPass and JIMS Works at the Same Time | 376

How ClearPass and JIMS Works at the Same Time? | 376

| 377

Example: Configure ClearPass and JIMS at the Same Time | 379

7 **Configuration Statements**

**actions (Services SSL Proxy) | 397**

**active-directory-access | 400**

**active-directory-authentication-table | 402**

**address (Services) | 403**

**address (Services User Identification) | 404**

address (Identity Management Advanced Query Primary) | 405

address (Identity Management Advanced Query Secondary) | 407

admin-search | 409

allow-reverse-ecmp | 410

application (Security Policies) | 411

application-services (Security Policies) | 413

assemble | 415

auth-only-browser | 416

auth-user-agent | 418

authentication-entry-timeout (Services User Identification) | 419

authentication-entry-timeout (Identity Management Advanced Query) | 420

authentication-source (Services User Identification ClearPass) | 422

authentication-source (Services User Identification Device Identity) | 424

batch query | 426

banner (Access FTP HTTP Telnet Authentication) | 429

banner (Access Web Authentication) | 430

base-distinguished-name | 431

ca-certificate (Services User Identification) | 432

ca-certificate (Identity Management Advanced Query Primary) | 433

ca-certificate (Identity Management Advanced Query Secondary) | 435

ca-profile (Services) | 437

captive-portal (Services UAC) | 438

captive-portal (Services UAC Policy) | 439

certficate (System Services) | 440

certificate-key (System Services) | 441

certificate-verification | 442

client (System Services) | 444

client-id (Services User Identification) | 445

client-id (Identity Management Advanced Query Primary) | 446

client-id (Identity Management Advanced Query Secondary) | 448

client-group | 450

client-idle-timeout (Access Profile) | 451

client-name-filter | 452

client-secret (Services User Identification) | 453

client-secret (Identity Management Advanced Query Primary) | 454

client-secret (Identity Management Advanced Query Secondary) | 456

client-session-timeout (Access Profile) | 458

configuration-file | 459

connection (Identity Management Advanced Query) | 460

connect-method (Identity Management Advanced Query) | 464

connect-method (Services User Identification) | 466

count | 467

custom-ciphers | 468

debug-level (System Services) | 471

debug-log (System Services) | 472

default-certificate (System Services) | 473

default-profile | 474

delay-query-time (Services User Identification) | 475

distinguished-name (Access) | 476

domain-name (Access Profile) | 477

enable-flow-tracing (Services) | 478

enable-session-cache | 479

end-user-profile | 480

fail | 482

file (Services User Identification) | 483

file (System Logging) | 484

filter (Security) | 486

filter (Identity Management Advanced Query) | 487

firewall-authentication | 491

firewall-authentication (Security) | 493

firewall-authentication (Security Policies) | 495

firewall-authentication (User Identification) | 497

firewall-authentication-forced-timeout | 498

firewall-authentication-service | 500

firewall-user | 501

flag (Services) | 502

from-zone (Security Policies) | 503

ftp (Access) | 507

group-profile (Access) | 508

http (Access) | 510

http (Services) | 511

http (Services User Identification) | 512

http (System Services) | 513

https (Services) | 514

https (Services User Identification) | 515

https (System Services) | 517

identity-management | 519

infranet-controller | 523

interface (Services) | 524

interval (Services) | 525

invalid-authentication-entry-timeout (Services User Identification Active Directory and ClearPass) | 526

ip-address (Access Profile) | 528

ip-query (Identity Management Advanced Query) | 529

ip-user-mapping | 531

ldap-options | 533

ldap-server | 534

level (Services) | 535

level (Services User Identification) | 536

lifetime-seconds (Security IKE) | 537

link (Access) | 538

local-authentication-table | 539

log (Services) | 540

login (Access) | 541

nas-port-type | 542

network (Access) | 543

no-remote-trace (Services User Identification) | 544

no-user-query (Services User Identification) | 545

no-tls-certificate-check | 546

pass-through | 547

password (Access) | 549

password (Services) | 550

password (System Services) | 551

permit (Security Policies) | 552

pki-local-certificate (Services) | 554

policies | 555

pool (Access) | 566

port (Access LDAP) | 568

port (Identity Management Advanced Query) | 569

port (Services) | 571

port (System Services) | 572

preferred-ciphers | 573

prefix (Access IPv6) | 574

primary connection (Identity Management Advanced Query) | 575

priority (Security User Identification) | 578

push-to-identity-management | 580

protocol-version | 582

query-api (Services User Identification) | 583

query-api (advanced user query) | 585

radius-options (Access) | 587

radius-server (Access) | 588

range (Access) | 589

rate-limit (Security Log) | 590

redirect-traffic | 591

redirect-url | 592

retry (Access LDAP) | 594

retry (Access RADIUS) | 595

revert-interval (Access LDAP) | 596

revert-interval (Access RADIUS) | 597

root-ca (Services) | 598

routing-instance (Access LDAP) | 599

routing-instance (Access RADIUS) | 600

search | 601

search-filter | 602

secondary connection (Identity Management Advanced Query) | 603

secret (Access Profile) | 605

securid-server | 606

separator | 607

server-certificate (Services) | 608

server-certificate-subject | 609

session-options (Access Profile) | 610

size (Services) | 611

source-address (Access LDAP) | 612

source-address (Access RADIUS) | 613

source-end-user-profile | 614

source-identity-log (Security) | 616

ssl (Services) | 618

ssl-termination-profile | 621

success | 622

system-generated-certificate | 623

telnet (Access) | 624

termination (Services) | 625

test-only-mode | 626

then (Security Policies) | 627

timeout (Access LDAP) | 630

timeout (Access RADIUS) | 631

timeout (Services) | 632

timeout-action | 633

tls-min-version | 634

tls-peer-name | 635

tls-timeout | 636

tls-type | 637

token-api (Services User Identification) | 638

token-api | 639

to-zone (Security Policies) | 641

traceoptions (Access) | 645

traceoptions (Active Directory Access) | 647

traceoptions (Security Firewall Authentication) | 649

traceoptions (Services SSL) | 651

traceoptions (Services UAC) | 654

traceoptions (Services User Identification) | 656

trusted-ca (Services) | 657

uac-policy (Application Services) | 658

uac-service | 659

unified-access-control (Security) | 660

unified-access-control (Services) | 661

user-group-mapping | 663

user-identification (Services) | 665

webapi (System Services) | 670

webapi-clear-text (Security) | 671

webapi-ssl (Security) | 672

web-authentication | 673

web-authentication (Access) | 674

web-authentication (Interfaces) | 676

web-management (System Services) | 677

web-redirect | 681

web-redirect-to-https | 682

web-server (Services) | 683

whitelist (Services) | 684

wins-server (Access) | 685

8

## Operational Commands

clear network-access requests pending | 691

clear network-access requests statistics | 693

clear network-access securid-node-secret-file | 694

clear security firewall-authentication history | 695

clear security firewall-authentication history address | 698

clear security firewall-authentication history identifier | 700

clear security firewall-authentication users | 702

clear security firewall-authentication users address | 705

clear security firewall-authentication users identifier | 707

clear security user-identification local-authentication-table | 709

clear service user-identification identity-management counter | 710

clear services user-identification active-directory-access | 711

clear services user-identification authentication-table | 712

request security user-identification local-authorization-table add | 714

request services user-identification active-directory-access active-directory-authentication-table delete | 716

request services user-identification active-directory-access domain-controller | 718

request services user-identification active-directory-access ip-user-probe | 720

request services user-identification authentication-source aruba-clearpass user-query | 722

request services user-identification authentication-table delete | 724

show network-access requests pending | 733

show network-access requests statistics | 736

show network-access securid-node-secret-file | 738

show security firewall-authentication history | 740

show security firewall-authentication history address | 743

show security firewall-authentication history identifier | 746

show security firewall-authentication jims statistics | 749

show security firewall-authentication users | 751

show security firewall-authentication users address | 755

show security firewall-authentication users identifier | 759

show security user-identification local-authentication-table | 763

show security policies | 766

show services unified-access-control counters | 783

show services unified-access-control policies | 786

show services unified-access-control roles | 789

show services unified-access-control status | 791

show services user-identification active-directory-access domain-controller status | 792

show services user-identification active-directory-access statistics | 795

show services user-identification active-directory-access user-group-mapping | 799

show service user-identification authentication-source aruba-clearpass user-query counters | 802

show service user-identification authentication-source aruba-clearpass user-query status | 804

show services user-identification authentication-table | 805

show service user-identification identity-management | 826

show services user-identification device-information table | 829

# About the Documentation

IN THIS SECTION


- Documentation and Release Notes | **xxi**
- Using the Examples in This Manual | **xxi**
- Documentation Conventions | **xxiii**
- Documentation Feedback | **xxvi**
- Requesting Technical Support | **xxvi**


Use this guide to set and enforce user-based and role-based security policies in Junos OS on SRX Series and NFX Series devices to restrict or permit users individually or in groups, using different authentication methods.

## 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 https://www.juniper.net/documentation/.

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 https://www.juniper.net/books.

## Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

### Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
       scripts {
           commit {
               file ex-script.xsl;
           }
       }
   }
   interfaces {
       fxp0 {
           disable;
           unit 0 {
               family inet {
                   address 10.0.0.1/24;
               }
           }
       }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

### Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the **load** command, see CLI Explorer.

## Documentation Conventions

Table 1 on page xxiv 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. |
| | Tip | Indicates helpful information. |
| | Best practice | Alerts you to a recommended use or implementation. |

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

**Table 2: Text and Syntax Conventions**

| Convention | Description | Examples |
|---|---|---|
| **Bold text like this** | Represents text that you type. | To enter configuration mode, type the **configure** command:<br>user@host> **configure** |
| `Fixed-width text like this` | Represents output that appears on the terminal screen. | `user@host>` **show chassis alarms**<br>`No alarms currently active` |
| *Italic text like this* | • Introduces or emphasizes important new terms.<br>• Identifies guide names.<br>• Identifies RFC and Internet draft titles. | • A policy *term* is a named structure that defines match conditions and actions.<br>• *Junos OS CLI User Guide*<br>• RFC 1997, *BGP Communities Attribute* |

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

| Convention | Description | Examples |
|---|---|---|
| *Italic text like this* | Represents variables (options for which you substitute a value) in commands or configuration statements. | Configure the machine's domain name:<br>[edit]<br>root@# **set system domain-name** *domain-name* |
| **Text like this** | Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components. | • To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level.<br>• The console port is labeled **CONSOLE**. |
| < > (angle brackets) | Encloses optional keywords or variables. | **stub <default-metric *metric*>;** |
| \| (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. | **broadcast \| multicast**<br>**(*string1* \| *string2* \| *string3*)** |
| # (pound sign) | Indicates a comment specified on the same line as the configuration statement to which it applies. | **rsvp { # Required for dynamic MPLS only** |
| [ ] (square brackets) | Encloses a variable for which you can substitute one or more values. | **community name members [ *community-ids* ]** |
| Indention and braces ( { } ) | Identifies a level in the configuration hierarchy. | [edit]<br>routing-options {<br>  static {<br>  route default {<br>  nexthop *address*;<br>  retain;<br>  }<br>  }<br>} |
| ; (semicolon) | Identifies a leaf statement at a configuration hierarchy level. | |
| **GUI Conventions** | | |

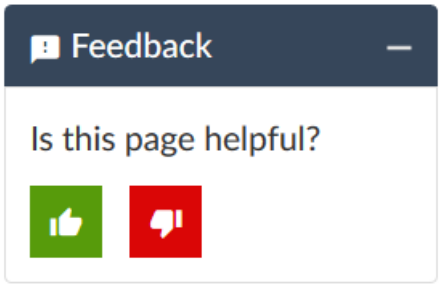
Table 2: Text and Syntax Conventions *(continued)*

| Convention | Description | Examples |
|---|---|---|
| **Bold text like this** | Represents graphical user interface (GUI) items you click or select. | • In the Logical Interfaces box, select **All Interfaces**.<br>• To cancel the configuration, click **Cancel**. |
| > (bold right angle bracket) | Separates levels in a hierarchy of menu selections. | In the configuration editor hierarchy, select **Protocols>Ospf**. |

## Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:




  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software 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 Juniper Care or Partner Support Services 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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

### Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- 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 https://support.juniper.net/support/requesting-support/.

CHAPTER 1

# Overview

Understanding User Authentication for Security Devices | 31

# Understanding User Authentication for Security Devices

Firewall user authentication lets you define firewall users and create policies that require the users to authenticate themselves through one of two authentication schemes: pass-through authentication or web authentication.

User role firewall policies can be integrated with firewall authentication both to authenticate users and to retrieve username and role information. The information is mapped to the IP address of the traffic, stored in the firewall authentication table, and used for user role firewall policy enforcement.

Infranet authentication occurs when an SRX Series device acts as an Infranet Enforcer for an IC Series device. You deploy the Infranet Enforcer in front of the servers and resources that you want to protect. Authentication occurs on the IC Series device and provides policies to the Enforcer to determine whether or not to allow an endpoint access to protected resources.

## RELATED DOCUMENTATION

Configure User Authentication Methods | 59

Configure Unified Access Control in Junos OS | 121

CHAPTER 2

# Firewall User Authentication


Firewall User Authentication Overview | 35

Configure Client Groups | 37

Customize the Firewall Authentication Banner | 41

Configure External Authentication Servers | 44

Configure User Authentication Methods | 59

Encrypt Traffic Using SSL Proxy and TLS | 95

# Firewall User Authentication Overview

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. Junos OS enables administrators to restrict and permit firewall users to access protected resources (different zones) behind a firewall based on their source IP address and other credentials.

Junos OS also supports the administrator and Point-to-Point Protocol (PPP) user types.

> **NOTE:** Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, HTTPS-based authentication is introduced on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways.

After you define firewall users, you can create a policy that requires the users to authenticate themselves through one of three authentication schemes:

- **Pass-through authentication**—A host or a user from one zone tries to access resources on another zone. You must use an FTP client, a Telnet client, an HTTP client, or an HTTPS client to access the IP address of the protected resource and to get authenticated by the firewall. The device uses FTP, Telnet, HTTP, or HTTPS to collect username and password information, and subsequent traffic from the user or host is allowed or denied based on the result of this authentication. When the device is using an HTTPS server, and after the authentication is done, the subsequent traffic from the user is always terminated whether the authentication is successful or not.

  > **NOTE:** Starting with Junos OS Release 12.1X44-D10 and Junos OS Release 17.3R1, support for HTTPS-based authentication is introduced for high-end SRX Series Services Gateways. It is not supported on SRX Series branch devices. For branch devices, you must use HTTP-based authentication.

  > **NOTE:** Starting in Junos OS Release 19.1R1, pass-through firewall user authentication is supported on NFX150 devices.

- **Pass-through with web-redirect authentication**—This authentication method can be used for HTTP or HTTPS client requests. When you configure firewall authentication to use pass-through authentication for HTTP and HTTPs client requests, you can use the web-redirect feature to direct the user's requests to the device's internal webserver. The webserver sends a redirect HTTP or HTTPS response to the

client system directing it to reconnect to the webserver for user authentication. The interface on which the client's request arrives is the interface to which the redirect response is sent.

> **NOTE:** For security reasons, on security policies that you configure for HTTP pass-through authentication, we recommend that you use web-redirect rather than direct pass-through authentication. The web browser may provide security by automatically including credentials for subsequent requests to the target web server.

Using this feature allows for a richer user login experience. For example, instead of a popup prompt asking the user to enter their username and password, users are presented with the login page in a browser. Enabling **web-redirect** has the same effect as if the user typed the web authentication IP address in a client browser. In that sense, **web-redirect** provides a seamless authentication experience; the user does not need to know the IP address of the web authentication source but only the IP address of the resource they are attempting to access. After the user has been authenticated, traffic from user's IP address is allowed to go through the **web-redirect** method.

A message is displayed to inform the user about the successful authentication. After successful authentication, the browser launches the user's original destination URL without their needing to retype the URL.

The following message is displayed:

```
Redirecting to the original url, please wait
```

- Web authentication—Users try to connect, using HTTP or HTTPS, to an IP address on the device that is enabled for Web authentication; in this scenario, you do not use HTTP or HTTPS to get to the IP address of the protected resource. You are prompted for the username and password that are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the result of this authentication.

**Release History Table**

| Release | Description |
|---|---|
| 19.1 | Starting in Junos OS Release 19.1R1, pass-through firewall user authentication is supported on NFX150 devices. |
| 15.1X49-D40 | Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, HTTPS-based authentication is introduced on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways. |
| 12.1X44 | Starting with Junos OS Release 12.1X44-D10 and Junos OS Release 17.3R1, support for HTTPS-based authentication is introduced for high-end SRX Series Services Gateways. |

# Configure Client Groups

**IN THIS SECTION**


- Understanding Client Groups for Firewall Authentication | 37
- Example: Configuring Local Users for Client Groups | 38


To manage multiple firewall users, create user or client groups and store the information.

## Understanding Client Groups for Firewall Authentication

To manage a number of firewall users, you can create user or client groups and store the information either on the local Juniper Networks device or on an external RADIUS or LDAP server.

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response. (For example, LDAP servers do not return such information.)

The RADIUS server sends the client's group information to the Juniper Networks device using Juniper VSA (46). The client-match portion of the policy accepts a string that can be either the username or the groupname to which the client belongs.

The reason to have a single database for different types of clients (except admins) is based on the assumption that a single client can be of multiple types. For example, a firewall user client can also be an L2TP client.

SEE ALSO

Example: Configuring RADIUS and LDAP User Authentication | 46

## Example: Configuring Local Users for Client Groups

IN THIS SECTION


- Requirements | 38
- Overview | 38
- Configuration | 38
- Verification | 40


This example shows how to configure a local user for client groups in a profile.

### Requirements

Before you begin, create an access profile.

### Overview

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response (for example, LDAP servers do not return such information).

This example shows how to configure a local user called Client-1 for client groups G1, G2, and G3 in a profile called Managers. Within this example, client groups are configured for a client. If a client group is not defined for the client, then the client group under the **access profile session-options** hierarchy is used.

### Configuration

#### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

**set access profile Managers client Client-1 client-group G1**

**set access profile Managers client Client-1 client-group G2**

**set access profile Managers client Client-1 client-group G3**

**set access profile Managers client Client-1 firewall-user password pwd**

**set access profile Managers session-options client-group G1**

**set access profile Managers session-options client-group G2**

**set access profile Managers session-options client-group G3**

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a local user for client groups in a profile:

1. Configure the firewall user profile Managers, and assign client groups to it.

```
user@host# edit access profile Managers
[edit access profile Managers]
user@host# set client Client-1 client-group G1
user@host# set client Client-1 client-group G2
user@host# set client Client-1 client-group G3
user@host# set client Client-1 firewall-user password pwd
```

2. Configure client groups in the session options.

```
[edit access profile Managers]
user@host# set session-options client-group G1
user@host# set session-options client-group G2
user@host# set session-options client-group G3
```

**Results**

Confirm your configuration by entering the **show access profile Managers** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

`user@host#` **show access profile Managers**

```
client Client-1 {
    client-group [ G1 G2 G3 ];
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
session-options {
    client-group [ G1 G2 G3 ];
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**


- Troubleshooting with Logs | **40**


To confirm that the configuration is working properly, perform this task:

***Troubleshooting with Logs***

**Purpose**

Use these logs to identify any issues.

**Action**

From operational mode, enter the **show log messages** command and the **show log dcd** command.

# Customize the Firewall Authentication Banner

IN THIS SECTION


- Understanding Firewall Authentication Banner Customization | 41
- Example: Customizing a Firewall Authentication Banner | 42


A banner is a customized message that you can create to indicate a user whether the authentication is successful or failed.

## Understanding Firewall Authentication Banner Customization

A banner is a message that appears on a monitor in different places depending on the type of login.

**Figure 1: Banner Customization**

- At the top of a browser screen after a user has successfully logged into a Web authentication address as shown Figure 1 on page 41.
- Before or after a Telnet, an FTP, an HTTP, or and HTTPS login prompt, success message, and fail message for users

All banners, except for a console login banner, have default messages. You can customize the messages that appear on the banners to better suit the network environment in which you use the device.

# Example: Customizing a Firewall Authentication Banner

IN THIS SECTION


- Requirements | 42
- Overview | 42
- Configuration | 42


This example shows how to customize the banner text that appears in the browser.

## Requirements

Before you begin, create an access profile.

## Overview

A banner is a message that appears on a monitor in different places depending on the type of login. This example shows how to change the banner that appears in the browser to indicate that a user has successfully authenticated after successfully logging in through Web authentication. The new message is “Web authentication is successful.” If the authentication fails, then the new message reads “Authentication failed.”

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

**set access firewall-authentication pass-through default-profile Profile-1**

**set access firewall-authentication pass-through ftp banner fail “ Authentication failed”**

**set access firewall-authentication web-authentication default-profile Profile-1**

**set access firewall-authentication web-authentication banner success “ Web authentication is successful”**

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To customize the banner text that appears in the browser:

1. Specify the banner text for failed pass-through authentication through FTP.

```
[edit]
user@host# set access firewall-authentication pass-through default-profile Profile-1
user@host# set access firewall-authentication pass-through ftp banner fail " Authentication failed"
```

2. Specify the banner text for successful Web authentication.

```
[edit]
user@host# set access web-authentication default-profile Profile-1
user@host# set access web-authentication banner success " Web authentication is successful"
```

**Results**

From configuration mode, confirm your configuration by entering the **show access firewall-authentication** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

`user@host#` **show access firewall-authentication**

```
pass-through {
    default-profile Profile-1;
    ftp {
        banner {
            fail "Authentication failed";
        }
    }
}
web-authentication {
    default-profile Profile-1;
    banner {
        success "Web authentication is successful";
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

# Configure External Authentication Servers

IN THIS SECTION


- Understanding External Authentication Servers | 44
- Example: Configuring RADIUS and LDAP User Authentication | 46
- Enabling LDAP Authentication with TLS/SSL for Secure Connections | 51
- Example: Configuring SecurID User Authentication | 53
- Example: Deleting the SecurID Node Secret File | 57


An external authentication server is used to collect user's credentials from the external servers for authentication.

## Understanding External Authentication Servers

IN THIS SECTION


- Understanding SecurID User Authentication | 45


Authentication, authorization, and accounting (AAA) servers provide an extra level of protection and control for user access in the following ways:

- Authentication determines the firewall user.
- Authorization determines what the firewall user can do.
- Accounting determines what the firewall user did on the network.

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

Once the user's credentials are collected, they are processed using firewall user authentication, which supports the following types of servers:

- Local authentication and authorization
- RADIUS authentication and authorization (compatible with Juniper Steel-Belted Radius server)
- LDAP authentication only (supports LDAP version 3 and is compatible with Windows AD)
- SecurID authentication only (using an RSA SecurID external authentication server)

> **NOTE:** Junos OS also supports administrative authentication using local, RADIUS, and TACACS+ servers.

This topic includes the following sections:

### Understanding SecurID User Authentication

SecurID is an authentication method that allows users to enter either static or dynamic passwords as their credentials. A dynamic password is a combination of a user's PIN and a randomly generated token that is valid for a short period of time, approximately one minute. A static password is set for the user on the SecurID server. For example, the SecurID server administrator might set a temporary static password for a user who lost his or her SecurID token.

When a user attempts to access a resource protected by a policy and SecurID is configured in the profile **authentication-order** parameter as either the only authentication mode or the first one to be used, the device forwards the user's credentials to the SecurID server for authentication. If the user enters valid values, the user is allowed access to the requested resource.

> **NOTE:** The SecurID server includes a feature that presents a user with a challenge if the user provides wrong credentials repeatedly. However, Junos OS does not support the challenge feature. Instead, the SecurID server administrator must resynchronize the RSA token for the user.

For SecurID, you configure information about the Juniper Networks device on the SecurID server, and this information is exported to a file called sdconf.rec.

To install the sdconf.rec file on the device, you must use an out-of-band method such as FTP. Install the file in a directory whose files are not deleted regularly. Do not put it in a temporary directory. For example, you might install it in /var/db/secureid/server1/sdconf.rec.

The sdconf.rec file contains information that provides the Juniper Networks device with the address of the SecurID server. You do not need to configure this information explicitly when you configure the SecurID server to be used as the external authentication server.

# Example: Configuring RADIUS and LDAP User Authentication

IN THIS SECTION


- Requirements | 46
- Overview | 46
- Configuration | 46
- Verification | 50


This example shows how to configure a device for external authentication.

## Requirements

Before you begin, create an authentication user group.

## Overview

You can put several user accounts together to form a user group, which you can store on the local database or on a RADIUS, an LDAP, or a SecurID server. When you reference an authentication user group and an external authentication server in a policy, the traffic matching the policy provokes an authentication check.

This example shows how access profile Profile-1 is configured for external authentication. Two RADIUS servers and one LDAP server are configured in the access profile. However, the order of authentication specifies RADIUS server only, so if the RADIUS server authentication fails, then the firewall user fails to authenticate. The local database is not accessed.

> **NOTE:** If the firewall clients are authenticated by the RADIUS server, then the group-membership VSA returned by the RADIUS server should contain alpha, beta, or gamma client groups in the RADIUS server configuration or in the access profile, Profile-1. Access profiles store usernames and passwords of users or point to external authentication servers where such information is stored.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

**set access profile Profile-1 authentication-order radius**

**set access profile Profile-1 client Client-1 client-group alpha**

**set access profile Profile-1 client Client-1 client-group beta**

**set access profile Profile-1 client Client-1 client-group gamma**

**set access profile Profile-1 client Client-1 firewall-user password pwd**

**set access profile Profile-1 client Client-2 client-group alpha**

**set access profile Profile-1 client Client-2 client-group beta**

**set access profile Profile-1 client Client-2 firewall-user password pwd**

**set access profile Profile-1 client Client-3 firewall-user password pwd**

**set access profile Profile-1 client Client-4 firewall-user password pwd**

**set access profile Profile-1 session-options client-group alpha**

**set access profile Profile-1 session-options client-group beta**

**set access profile Profile-1 session-options client-group gamma**

**set access profile Profile-1 session-options client-idle-timeout 255**

**set access profile Profile-1 session-options client-session-timeout 4**

**set access profile Profile-1 ldap-options base-distinguished-name CN=users,DC=junos,DC=juniper,DC=net**

**set access profile Profile-1 ldap-options search search-filter sAMAccountName=**

**set access profile Profile-1 ldap-options search admin-search distinguished-name cn=administrator,cn=users,dc=junos,dc=juniper,dc=net**

**set access profile Profile-1 ldap-options search admin-search password pwd**

**set access profile Profile-1 ldap-server 203.0.113.39/24**

**set access profile Profile-1 radius-server 203.0.113.62/24 secret example-secret**

**set access profile Profile-1 radius-server 203.0.113.62/24 retry 10**

**set access profile Profile-1 radius-server 203.0.113.27/24 secret juniper**

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a device for external authentication:

1. Specify the RADIUS server for external authentication order.

```
[edit]
user@host# set access profile Profile-1 authentication-order radius
```

2. Configure Client1-4 firewall users and assign the Client-1 firewall user and Client-2 firewall user to client groups.

```
[edit access profile Profile-1]
user@host# set client Client-1 client-group alpha
user@host# set client Client-1 client-group beta
user@host# set client Client-1 client-group gamma
user@host# set client Client-1 firewall-user password pwd
user@host# set client Client-2 client-group alpha
user@host# set client Client-2 client-group beta
user@host# set client Client-2 firewall-user password pwd
user@host# set client Client-3 firewall-user password pwd
user@host# set client Client-4 firewall-user password pwd
```

3. Configure client groups in the session options.

```
[edit access profile Profile-1]
user@host# set session-options client-group alpha
user@host# set session-options client-group beta
user@host# set session-options client-group gamma
user@host# set session-options client-idle-timeout 255
user@host# set session-options client-session-timeout 4
```

4. Configure the IP address for the LDAP server and server options.

```
[edit access profile Profile-1]
user@host# set ldap-options base-distinguished-name CN=users,DC=junos,DC=mycompany,DC=net
user@host# set ldap-options search search-filter sAMAccountName=
user@host# set ldap-options search admin-search password pwd
user@host# set ldap-options search admin-search distinguished-name
   cn=administrator,cn=users,dc=junos,dc=mycompany,dc=net
```

```
user@host# set ldap-server 203.0.113.39/24
```

5. Configure the IP addresses for the two RADIUS servers.

```
[edit access profile Profile-1]
user@host# set radius-server 203.0.113.62/24 secret pwd
user@host# set radius-server 203.0.113.62/24 retry 10
user@host# set radius-server 203.0.113.27/24 secret pwd
```

**Results**

From configuration mode, confirm your configuration by entering the **show access profile Profile-1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

`user@host#` **show access profile Profile-1**

```
authentication-order radius;
client Client-1 {
    client-group [ alpha beta gamma ];
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
client Client-2 {
    client-group [ alpha beta ];
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
client Client-3 {
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
client Client-4 {
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
session-options {
    client-group [ alpha beta gamma ];
    client-idle-timeout 255;
```

```
    client-session-timeout 4;
}
ldap-options {
    base-distinguished-name CN=users,DC=junos,DC=juniper,DC=net;
    search {
        search-filter sAMAccountName=;
        admin-search {
            distinguished-name cn=administrator,cn=users,dc=junos,
       dc=mycompany,dc=net; password "$ABC123"; ## SECRET-DATA
        }
    }
}
ldap-server {
    203.0.113.39/24 ;
}
radius-server {
    203.0.113.62/24  {
        secret "$ABC123"; ## SECRET-DATA
        retry 10;
    }
    203.0.113.27/24  {
        secret "$ABC123"; ## SECRET-DATA
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**


- Troubleshooting with Logs | **50**


To confirm that the configuration is working properly, perform this task:

***Troubleshooting with Logs***

**Purpose**

Use these logs to identify any issues.

**Action**

From operational mode, enter the **show log messages** command and the **show log dcd** command.

## Enabling LDAP Authentication with TLS/SSL for Secure Connections

Beginning with Junos OS Release 15.1X49-D70, SRX Series devices support the Transport Layer Security (TLS) StartTLS extension for LDAP for firewall user authentication and the integrated user firewall authentication for obtaining username and role information through firewall authentication. StartTTLS allows protocol data transfers between the LDAP server and client over the TLS layer after successful negotiation between the peers. StartTLS upgrades an existing insecure LDAP connection to a secure TLS/SSL connection.

> **NOTE:** SRX Series devices support TLSv1.1 and TLS v1.2 to use LDAP authentication with TLS/SSL.

With StartTLS for LDAP, a secure communication can be provided with the following sets of ciphers that provide increasingly strong security:

- High encryption cipher: AES256-SHA,DES-CBC3-SHA
- Medium encryption ciphers: High encryption cipher + RC4-SHA:RC4-MD5:AES128-SHA
- Medium encryption ciphers: Medium encryption ciphers + DES-CBC-SHA:EXP1024-DES-CBC-SHA:EXP1024-RC4-SHA: EXP1024-RC4-MD5:EXP-DES-CBC-SHA:EXP-RC4-MD5

Implementation of StartTLS on LDAP is interoperable with the following standard LDAP servers:

- Windows Active Directory
- Novell e-Directory
- Sun LDAP
- OpenLDAP

By default, LDAP traffic is not transmitted securely. You can set LDAP traffic to be confidential and secure by using Secure Sockets Layer/Transport Layer Security (SSL/TLS) technology.

To configure TLS parameters as a part of LDAP server configuration:

1. Define TLS type as **start-tls** to configure LDAP over StartTLS.

```
[edit]
```

```
user@host# set access profile profile-name ldap-server ip-address tls-type start-tls
```

2. Configure the peer host name to be authenticated.

```
[edit]
user@host# set access profile profile-name ldap-server ip-address tls-peer-name peer-name
```

3. Specify the timeout value on the TLS handshake. You can enter 3 through 90 seconds.

```
[edit]
user@host# set access profile profile-name ldap-server ip-address tls-timeout
```

4. Specify TLS version (v1.1 and v1.2 are supported) as the minimum protocol version enabled in connections. By default, SRX Series device uses TLS v1.2 to negotiate the TLS connection with the LDAP server:

```
[edit]
user@host# set access profile profile-name ldap-server ip-address tls-min-version supported-tls-version
```

**NOTE:** SRX Series devices support an additional check on the LDAP server's certificate during the TLS handshake for LDAP authentication by default. If the validation of the server certificate is not required, you can use the following configuration to ignore the validation of server's certificate and accept the certificate without checking:

```
[edit]
user@host# set access profile profile-name ldap-server ip-address no-tls-certificate-check
```

By default, the **no-tls-certificate-check** remains disabled.

# Example: Configuring SecurID User Authentication

IN THIS SECTION


- Requirements | 53
- Overview | 53
- Configuration | 53
- Verification | 56
- Troubleshooting | 57


This example shows how to configure SecurID as the external authentication server.

## Requirements

Before you begin, create an authentication user group.

## Overview

SecurID is an authentication method that allows users to enter either static or dynamic passwords as their credentials. A dynamic password is a combination of a user's PIN and a randomly generated token that is valid for a short period of time, approximately one minute. A static password is set for the user on the SecurID server. For example, the SecurID server administrator might set a temporary static password for a user who lost his or her SecurID token.

When a user attempts to access a resource protected by a policy and SecurID is configured in the profile **authentication-order** parameter as either the only authentication mode or the first one to be used, the device forwards the user's credentials to the SecurID server for authentication. If the user enters valid values, the user is allowed access to the requested resource.

Specify that Server-1 is to be used as the SecurID server and that its configuration file resides on the device in the **/var/db/securid/Server-1/sdconf.rec** file. From configuration mode, enter this command:

```
user@host# set access securid-server Server-1 configuration-file “/var/db/securid/Server-1/sdconf.rec”
```

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

**set access profile Profile-2 authentication-order securid**

**set access profile Profile-2 client Client-1 client-group alpha**

**set access profile Profile-2 client Client-1 client-group beta**

**set access profile Profile-2 client Client-1 client-group gamma**

**set access profile Profile-2 client Client-1 firewall-user password pwd**

**set access profile Profile-2 client Client-2 client-group alpha**

**set access profile Profile-2 client Client-2 client-group beta**

**set access profile Profile-2 client Client-2 firewall-user password pwd**

**set access profile Profile-2 client Client-3 firewall-user password pwd**

**set access profile Profile-2 client Client-4 firewall-user password pwd**

**set access profile Profile-2 session-options client-group alpha**

**set access profile Profile-2 session-options client-group beta**

**set access profile Profile-2 session-options client-group gamma**

**set access profile Profile-2 session-options client-idle-timeout 255**

**set access profile Profile-2 session-options client-session-timeout 4**

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure SecurID as the external authentication server:

1. For the Profile-2 profile, configure SecurID as the server to be used for external authentication.

   ```
   [edit]
   user@host# set access profile Profile-2 authentication-order securid
   ```

   To share a single SecurID server across multiple profiles, for each profile set the **authentication-order** parameter to include **securid** as the authentication mode.

2. Configure clients 1 through 4 as firewall users, and assign Client-1 and Client-2 to client groups.

```
[edit access profile Profile-2]
user@host# set client Client-1 client-group alpha
user@host# set client Client-1 client-group beta
user@host# set client Client-1 client-group gamma
user@host# set client Client-1 firewall-user password pwd
user@host# set client Client-2 client-group alpha
user@host# set client Client-2 client-group beta
user@host# set client Client-2 firewall-user password pwd
user@host# set client Client-3 firewall-user password pwd
user@host# set client Client-4 firewall-user password pwd
```

3. Configure client groups in the session options.

```
[edit access profile Profile-2]
user@host# set session-options client-group alpha
user@host# set session-options client-group beta
user@host# set session-options client-group gamma
user@host# set session-options client-idle-timeout 255
user@host# set session-options client-session-timeout 4
```

**Results**

From configuration mode, confirm your configuration by entering the **show access profile Profile-2** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
```

```
user@host# show access profile Profile-2
```

```
authentication-order securid;
client Client-1 {
    client-group [ alpha beta gamma ];
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
client Client-2 {
    client-group [ alpha beta ];
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
```

```
    }
}
client Client-3 {
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
client Client-4 {
    firewall-user {
        password "$ABC123"; ## SECRET-DATA
    }
}
session-options {
    client-group [alpha beta gamma];
    client-idle-timeout 255;
    client-session-timeout 4;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**


- Troubleshooting with Logs | **56**


To confirm that the configuration is working properly, perform this task:

***Troubleshooting with Logs***

**Purpose**

Use these logs to identify any issues.

**Action**

From operational mode, enter the **show log messages** command and the **show log dcd** command.

### Troubleshooting

IN THIS SECTION


- Troubleshooting Unsuccessful Authentication In a Dynamic VPN Configuration | 57


#### *Troubleshooting Unsuccessful Authentication In a Dynamic VPN Configuration*

**Problem**

Device fails to locate client address in a dynamic VPN configuration.

**Solution**

1. Verify that the device host name, the domain-search, and the name server are configured properly.

```
[edit system]
user@host# set host-name srxhost.example.net
user@host# set domain-search domain.example.net
user@host# set name-server 203.0.113.11
```

2. Verify that the device host name is getting resolved on the RSA server.

## Example: Deleting the SecurID Node Secret File

IN THIS SECTION


- Requirements | 58
- Overview | 58
- Configuration | 58
- Verification | 58


This example shows how to delete the node secret file.

### Requirements

Before you begin, confirm that it is necessary to delete the SecurID node secret file.

### Overview

When the Juniper Networks device initially communicates successfully with the SecurID server, a node secret file is created for it automatically. The file is created as a result of the handshake between the Juniper Networks device and the SecurID server after the software authenticates the first user successfully. All subsequent communication between the Juniper Networks device and the SecurID server relies on this secret as a representation of trust between the two nodes instead of repeating the handshake with each authentication request.

Under normal circumstances you should not delete the node secret file. In the rare case that you must do so, for example, to debug a serious problem, you can use the **clear** command to remove the file.

**WARNING:** If you delete the file, you must deselect a box on the SecurID server to indicate that the node secret file for the Juniper Networks device and the SecurID server no longer exists. Otherwise, authentication attempts will fail.

### Configuration

#### Step-by-Step Procedure

To delete the node secret file:

1. Use the **clear** command to remove the node secret file. During subsequent user authentication, the device reestablishes a shared secret with the SecurID server and re-creates the node secret file. From operational mode, enter the **clear network-access** command to clear the **securid-node-secret-file** for the Juniper Networks device.

   ```
   user@host> clear network-access securid-node-secret-file
   ```

2. From operational mode, confirm your deletion by entering the **show network-access securid-node-secret-file** command. If the output does not display, repeat the instructions in this example to correct it.

   ```
   user@host> show network-access securid-node-secret-file
   ```

### Verification

Verify the deletion by entering the **show network-access securid-node-secret-file** command.

# Configure User Authentication Methods

IN THIS SECTION


- Understanding Pass-Through Authentication | 59
- Example: Configuring Pass-Through Authentication | 61
- Example: Configuring HTTPS Traffic to Trigger Pass-Through Authentication | 70
- Understanding Web Authentication | 78
- Example: Configuring Web Authentication | 80
- Example: Configuring HTTPS Traffic to Trigger Web Authentication | 89


Pass-through authentication and web authentication are the two authenticating methods to authenticate the users.

## Understanding Pass-Through Authentication

Pass-through user authentication is a form of active authentication; the user is prompted to enter a username and password when pass-through authentication is invoked. If the user's identity is validated, the user is allowed to pass through the firewall and gain access to the requested resources.

When a user attempts to initiate an HTTP, an HTTPS, an FTP, or a Telnet connection request that has a policy requiring authentication, the device intercepts the request and prompts the user to enter a username and password. Depending on the configuration, the device validates the username and password by checking them against those stored in the local database or on an external authentication server.

If an external authentication server is used, after the user's credentials are collected, they are processed through firewall user authentication. The following external authentication servers are supported:

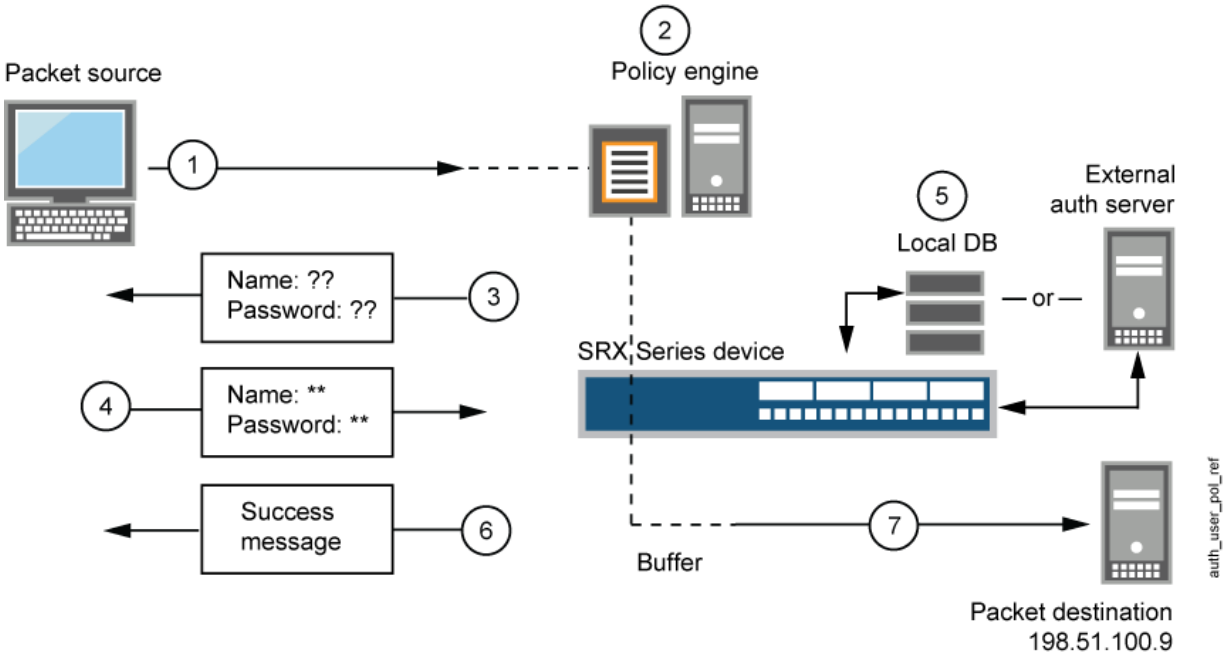
- RADIUS authentication and authorization (compatible with Juniper Steel-Belted Radius servers)

  You can use an external RADIUS server if, in addition to authentication, you want to obtain authorization information about the user's access right (what the user can do on the network).
- LDAP authentication only (supports LDAP version 3, compatible with Windows AD)
- SecurID authentication only (uses an RSA SecurID external authentication server)

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. You can put several user accounts together to form a user group, which you can store on the local database or on a RADIUS, an LDAP, or a SecurID server. When you reference an authentication user group and an external authentication server in a policy, the traffic matching the policy triggers an authentication check.

> **NOTE:** You use family inet to assign an IPv4 address. You use family inet6 to assign an IPv6 address. An interface can be configured with both an IPv4 and an IPv6 address. For the sake of brevity, these examples use IPv4 addresses only.

**Figure 2: Policy Lookup for a User**




The steps in Figure 2 on page 60 are as follows:

1. A client user sends an FTP, an HTTP, an HTTPS, or a Telnet packet to 198.51.100.9.
2. The device intercepts the packet, notes that its policy requires authentication from either the local database or an external authentication server, and buffers the packet.
3. The device prompts the user for login information through FTP, HTTP, HTTPS, or Telnet.
4. The user replies with a username and password.
5. The device either checks for an authentication user account on its local database or sends the login information to the external authentication server as specified in the policy.

6. Finding a valid match (or receiving notice of such a match from the external authentication server), the device informs the user that the login has been successful.
7. For HTTP, HTTPS, or Telnet traffic, the device forwards the packet from its buffer to its destination IP address, 198.51.100.9/24. However, for FTP traffic, after successful authentication, the device closes the session and the user must reconnect to the FTP server at IP address 198.51.100.9/24.

> **NOTE:** For security purposes, we recommend that you use web-redirect rather than direct pass-through authentication on security policies that you configure for HTTP pass-through authentication. The web browser may provide security by automatically including credentials for subsequent requests to the target web server.

After the device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through pass through—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.

The pass-through user authentication method is recommended in situations when security has a higher priority than convenience. This authentication method applies only to the session and child sessions matching the policy that triggered it. You can apply this method on Internet-facing links, if used with caution.

## Example: Configuring Pass-Through Authentication

**IN THIS SECTION**


- Requirements | 62
- Overview | 62
- Configuration | 63
- Verification | 68


This example shows how to configure pass-through authentication to authenticate firewall users. A firewall user is a network user who must provide a username and password when initiating a connection across the firewall.

Pass-through authentication allows SRX Series administrators to restrict users who attempt to access a resource in another zone using FTP, Telnet, HTTP, or HTTPS. If the traffic matches a security policy whose action is pass-through authentication, the user is required to provide login information.

For HTTPS, to ensure security the HTTPS default certificate key size is 2048 bits. If you do not specify a certificate size, the default size is assumed.

### Requirements

Before you begin, define firewall users. See Firewall User Authentication Overview.

This example uses the following hardware and software components:

- SRX Series device
- Firewall user's system
- Packet destination system

### Overview

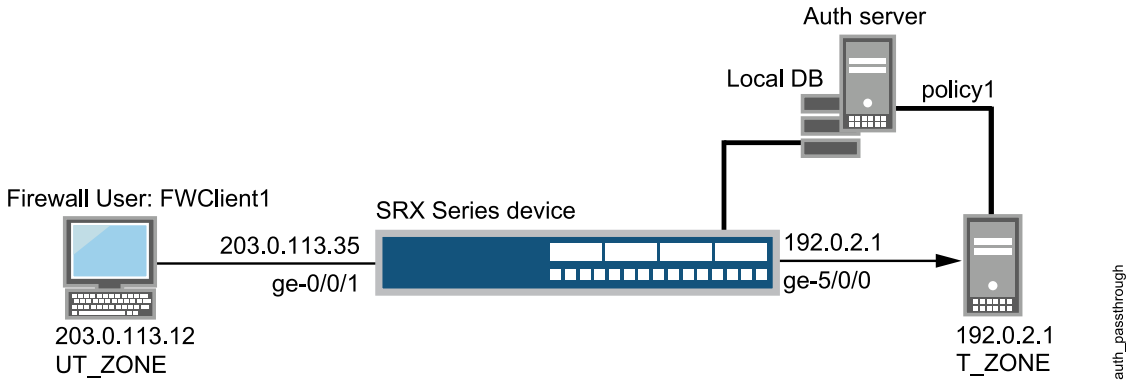
The pass-through authentication process is triggered when a client, referred to as a firewall user, attempts to initiate an FTP, a Telnet, or an HTTP session to access a resource in another zone. The SRX Series firewall acts as a proxy for an FTP, a Telnet, an HTTP, or an HTTPS server so that it can authenticate the firewall user before allowing the user access to the actual FTP, Telnet, or HTTP server behind the firewall.

If traffic generated from a connection request sent by a firewall user matches a security policy rule bidirectionally and that rule specifies pass-through firewall authentication as the action of its **then** clause, the SRX Series device requires the firewall user to authenticate to a Junos OS proxy server.

If the authentication is successful, subsequent traffic from the same source IP address is automatically allowed to pass through the SRX Series device if the traffic matches the security policy tuples.

Figure 3 on page 63 shows the topology used in this example.

Figure 3: Configuring Pass-Through Firewall Authentication




**NOTE:** Although the topology shows use of an external server, it is not covered in the configuration. It is outside the scope of this example.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

**set interfaces ge-0/0/1 unit 0 family inet address 203.0.113.35/24**

**set interfaces ge-5/0/0 unit 0 family inet address 192.0.2.1/24**

**set access profile FWAUTH client FWClient1 firewall-user password *password***

**set access firewall-authentication pass-through default-profile FWAUTH**

**set access firewall-authentication pass-through telnet banner success "WELCOME TO JUNIPER TELNET SESSION"**

**set security zones security-zone UT-ZONE host-inbound-traffic system-services all**

**set security zones security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic protocols all**

**set security zones security-zone T-ZONE host-inbound-traffic system-services all**

**set security zones security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic protocols all**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match source-address any**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match destination-address any**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match application junos-telnet**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 then permit firewall-authentication pass-through client-match FWClient1**

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure pass-through authentication:

1. Configure two interfaces and assign IP addresses to them.

   **NOTE:** For this example, it is optional to assign two addresses to the interfaces.

   ```
   [edit]
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 203.0.113.35/24
   user@host# set interfaces ge-5/0/0 unit 0 family inet address 192.0.2.1/24
   ```

2. Create the FWAUTH access profile for the FWClient1 user, specify the user's password, and define a success banner for Telnet sessions.

   ```
   [edit access]
   user@host# set access profile FWAUTH client FWClient1 firewall-user password pwd
   user@host# set firewall-authentication pass-through default-profile FWAUTH
   user@host# set firewall-authentication pass-through telnet banner success "WELCOME TO JUNIPER TELNET
     SESSION"
   ```

3. Configure security zones.

   **NOTE:** For this example, it is optional to configure a second interface for a security zone.

```
[edit security zones]
user@host# set security-zone UT-ZONE host-inbound-traffic system-services all
user@host# set security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic protocols all
user@host# set security-zone T-ZONE host-inbound-traffic system-services all
user@host# set security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic protocols all
```

4. Assign security policy P1 to the security zones.

```
[edit security policies]
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match source-address any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match destination-address any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match application junos-telnet
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 then permit firewall-authentication
    pass-through client-match FWClient1
```

5. Use Telnet to authenticate the FWClient1 firewall user to host2.

```
user@FWClient1# run telnet 192.0.2.1/24
Trying 192.0.2.1/24...
Connected to 192.0.2.1/24
Escape character is '^]'.
Firewall User Authentication
Username: FWClient1
Password:$ABC123
WELCOME TO JUNIPER TELNET SESSION
Host1 (ttyp0)
login: user
Password: $ABC123
--- JUNOS 10.1R1.1 built 2009-10-12 13:30:18 UTC
%
```

**Results**

From configuration mode, confirm your configuration by entering these commands.

- **show interfaces**
- **show access**
- **show security zones**
- **show security policies**

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, the output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 203.0.113.35;
            }
        }
    }
    ge-5/0/0 {
        unit 0 {
            family inet {
                address 192.0.2.1/24;
            }
        }
    }
...
```

```
user@host# show access
    profile FWAUTH {
        authentication-order password;
        client FWClient1 {
            firewall-user {
                password "$ABC123"; ## SECRET-DATA
            }
        }
    }
    firewall-authentication {
        pass-through {
            default-profile FWAUTH;
            telnet {
                banner {
                    success "WELCOME TO JUNIPER TELNET SESSION";
                }
            }
        }
    }
```

```
user@host# show security zones
   security-zone UT-ZONE {
      host-inbound-traffic {
         system-services {
            all;
         }
      }
         interfaces {
            ge-0/0/1.0 {
               host-inbound-traffic {
                  protocols {
                     all;
                  }
               }
            }
         }
         security-zone T-ZONE {
            host-inbound-traffic {
               system-services {
                  all;
               }
            }
            interfaces {
               ge-5/0/0.0 {
                  host-inbound-traffic {
                     protocols {
                        all;
                     }
                  }
               }
            }
         }
```

```
user@host# show security policies
...
from-zone UT-ZONE to-zone T-ZONE {
   policy P1 {
      match {
         source-address any;
         destination-address any;
         application junos-telnet;
      }
      then {
         permit {
```

```
            firewall-authentication {
               pass-through {
                  client-match FWClient1;
               }
            }
         }
      }
   }
}
```

If you are done configuring the device, enter commit from configuration mode.

## Verification

**IN THIS SECTION**

- Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table | **68**

To confirm that the configuration is working properly, perform this task:

***Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table***

**Purpose**

Display firewall authentication user history and verify the number of firewall users who successfully authenticated and the number of firewall users who failed to log in.

**Action**

From operational mode, enter these **show** commands:

`user@host>` **show security firewall-authentication history**

```
History of firewall authentication data:
Authentications: 2
Id Source Ip Date Time Duration Status User
1 203.0.113.12 2010-10-12 21:24:02 0:00:24 Failed FWClient1
2 203.0.113.12 2010-10-12 21:24:48 0:00:22 Success FWClient1
```

`user@host>` **show security firewall-authentication history identifier 1**

```
Username: FWClient1
Source IP: 203.0.113.12
Authentication state: Success
Authentication method: Pass-through using Telnet
Access start date: 2010-10-12
Access start time: 21:24:02
Duration of user access: 0:00:24
Source zone: UT-ZONE
Destination zone: T-ZONE
Access profile: FWAUTH
Bytes sent by this user: 0
Bytes received by this user: 2660
```

user@host> **show security firewall-authentication users**

```
Firewall authentication data:
Total users in table: 1
Id Source Ip Src zone Dst zone Profile Age Status User
4 203.0.113.12     UT-ZONE  T-ZONE   FWAUTH     1 Success   FWClient1
```

user@host> **show security firewall-authentication users identifier 3**

```
Username: FWClient1
Source IP: 203.0.113.12
Authentication state: Success
Authentication method: Pass-through using Telnet
Age: 3
Access time remaining: 9
Source zone: UT-ZONE
Destination zone: T-ZONE
Access profile: FWAUTH
Interface Name: ge-0/0/1.0
Bytes sent by this user: 0
Bytes received by this user: 1521
```

# Example: Configuring HTTPS Traffic to Trigger Pass-Through Authentication

IN THIS SECTION


- Requirements | 70
- Overview | 71
- Configuration | 72
- Verification | 77


This example shows how to configure HTTPS traffic to trigger pass-through authentication. HTTPS is more secure than HTTP, so it has become more popular and is more widely used.

## Requirements

This example uses the following hardware and software components:

- SRX Series device
- Two PCs running Linux and Open SSL. One PC acts as a client and another as an HTTPS server. The two PCs are used to create key files and to send traffic.
- Junos OS Release 12.1X44-D10 or later for SRX5400, SRX5600, and SRX5800 devices and Junos OS Release 15.1X49-D40 or later for vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways.

> **NOTE:** Starting in Junos OS Release 12.1X44-D10 and Junos OS Release 17.3R1, HTTPS-based authentication is introduced on SRX5400, SRX5600, and SRX5800 devices.
>
> Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, HTTPS-based authentication is introduced on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways.

Before you begin:

An SRX Series device has to decode HTTPS traffic to trigger pass-through authentication. Then, SSL termination proxy creates and installs a private key file and a certification file. The following list describes the steps to create and install a private key file and a certification key file.

> **NOTE:** If you have an official **.crt** file and **.key** file, then you can directly upload and install the files on the SRX Series device. If you do not have a **.crt** file and **.key** file, follow the procedure to create and install the files. Instructions specified in Step 1 and Step 2 must be run on a PC with Linux and OpenSSL installed. Instructions specified in Step 3 and Step 4 must be run in operational mode.

To create and install a private key file and a certification file:

1. On a PC create the **.key** file.

   ```
   openssl genrsa -out /tmp/server.key 1024
   ```

2. On a PC, create the **.crt** file.

   ```
   openssl req -new -x509 -days 365 -key /tmp/server.key -out /tmp/device.crt -subj
     "/C=CN/ST=BJ/L=BJ/O=JNPR/OU=CNRD/CN=203.0.113.11/emailAddress=device@mycompany.com"
   ```

3. Upload the **.key** and **.crt** files to an SRX Series device, and install the files on the device using the following command from operational mode:

   ```
   user@host> request security pki local-certificate load filename /var/tmp/device.crt key /var/tmp/device.key
     certificate-id device
   ```

## Overview

Firewall authentication initiates a secure connection to be established across two devices. A network user must provide a username and password for authentication when initiating a connection across the firewall. Firewall authentication supports HTTPS traffic for pass-through authentication. HTTPS can secure HTTP firewall authentication traffic between users and the SRX Series device.

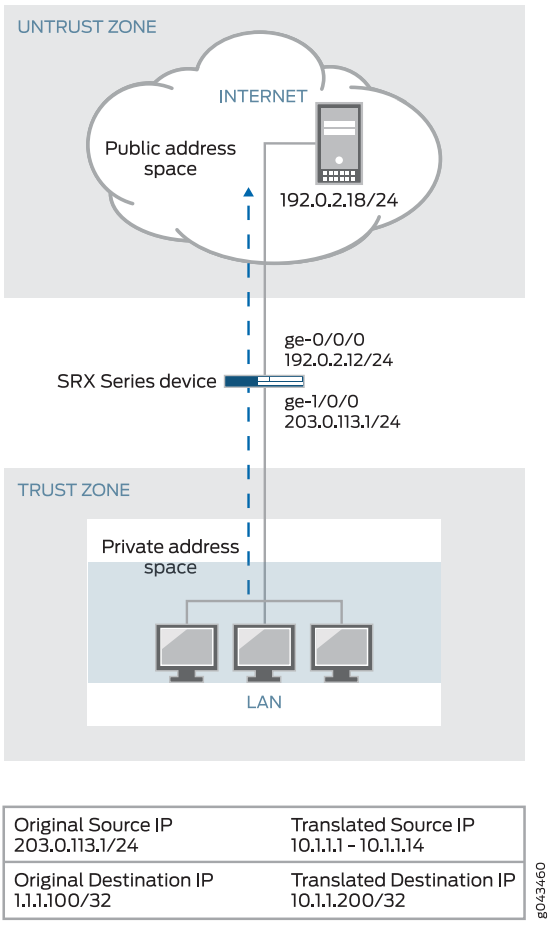
HTTPS is the secure version of HTTP, the protocol over which data is sent between the user and the device that the user is connected to. All communications between the user and the connected devices are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms.

In this example, HTTPS traffic is used to trigger pass-through authentication because HTTPS is more secure than HTTP. For HTTPS traffic to trigger pass-through authentication you must first configure the SSL termination profile.

Figure 4 on page 72 shows an example of pass-through authentication using HTTPS traffic. In this example, a host or a user from an untrust zone tries to access resources on the trust zone. The SRX Series device

uses HTTPS to collect the username and password information. Subsequent traffic from the host or user is allowed or denied based on the result of this authentication.

**Figure 4: Pass-Through Authentication Using HTTPS Traffic**




## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.12/24
set interfaces ge-1/0/0 unit 0 family inet address 203.0.113.1/24
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
```

```
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication pass-through
   access-profile local_pf
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication pass-through
   ssl-termination-profile ssl_pf
set security policies from-zone trust to-zone untrust policy p1 then log session-init
set security policies from-zone trust to-zone untrust policy p1 then log session-close
set security zones security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-1/0/0.0 host-inbound-traffic system-services all
set security zones security-zone untrust interfaces ge-1/0/0.0 host-inbound-traffic protocols all
set access profile local_pf client user1 firewall-user password <password>
set access firewall-authentication pass-through default-profile local_pf
set services ssl termination profile ssl_pf server-certificate device
```

**Step-by-Step Procedure**

To configure HTTPS traffic to trigger pass-through authentication:

1. Configure interfaces and assign IP addresses.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 192.0.2.12/24
user@host# set ge-1/0/0 unit 0 family inet address 203.0.113.1/24
```

2. Configure security policies to permit firewall authenticated traffic from zone trust to zone untrust.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy p1 then permit firewall-authentication pass-through
   access-profile local_pf
user@host# set from-zone trust to-zone untrust policy p1 then permit firewall-authentication pass-through
   ssl-termination-profile ssl_pf
```

3. Specify a policy action to take when a packet matches the criteria.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy p1 match source-address any
user@host# set from-zone trust to-zone untrust policy p1 match destination-address any
user@host# set from-zone trust to-zone untrust policy p1 match application any
user@host# set from-zone trust to-zone untrust policy p1 then log session-init
user@host# set from-zone trust to-zone untrust policy p1 then log session-close
```

4. Configure security zones and assign interfaces.

```
[edit security zones]
user@host#  set security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic protocols all
user@host# set security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic system-services all
```

5. Configure application services for zones.

```
[edit security zones]
user@host# set security-zone trust host-inbound-traffic system-services all protocols all
user@host# set security-zone untrust host-inbound-traffic system-services all protocols all
```

6. Create an access profile and configure the client as a firewall user and set the password.

```
[edit access]
user@host# set profile local_pf client user1 firewall-user password <password>
```

7. Configure the type of firewall and the default profile name where the authentication settings are defined.

```
[edit access]
user@host# set firewall-authentication pass-through default-profile local_pf
```

8. Configure the SSL termination profile and enter a local certificate identifier name.

```
[edit services]
user@host# set ssl termination profile ssl_pf server-certificate device
```

**Results**

From configuration mode, confirm your configuration by entering the **show interfaces**, **show security policies**, **show security zones**, **show access**, and **show services ssl termination** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show interfaces
...
interfaces
    ge-0/0/0 {
        unit 0 {
            family inet {
```

```
            address 192.0.2.12;
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}
```

```
user@host# show security policies
...
policies
    from-zone trust to-zone untrust {
        policy p1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit {
                    firewall-authentication {
                        pass-through {
                            access-profile local_pf;
                            ssl-termination-profile ssl_pf;
                        }
                    }
                }
                log {
                    session-init;
                    session-close;
                }
            }
        }
    }
}
```

```
user@host# show security zones
...
zones {
```

```
    security-zone trust {
        interfaces {
            ge-0/0/0.0 {
                host-inbound-traffic {
                    system-services {
                        all;
                    }
                    protocols {
                        all;
                    }
                }
            }
        }
    }
    security-zone untrust {
        interfaces {
            ge-1/0/0.0 {
                host-inbound-traffic {
                    system-services {
                        all;
                    }
                    protocols {
                        all;
                    }
                }
            }
        }
    }
```

```
user@host# show access
...
access {
    profile local_pf {
        client user1 {
            firewall-user {
                password password;
            }
        }
    }
    firewall-authentication {
        pass-through {
            default-profile local_pf;
        }
    }
```

```
user@host# show services ssl termination
...
services {
    ssl {
        termination {
            profile ssl_pf {
                server-certificate device;
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

### *Verifying the Configuration*

#### Purpose

Verify that the configuration is correct.

#### Action

From operational mode, enter the **show security firewall-authentication users** command for identifier 1.

```
user@host> show security firewall-authentication users identifier 1
    Username: user1
    Source IP: 203.0.113.1/24
    Authentication state: Success
    Authentication method: Pass-through using HTTPS
    Age: 0
    Access time remaining: 10
    Lsys: root-logical-system
    Source zone: trust
    Destination zone: untrust
    Access profile: local_pf
    Interface Name: ge-0/0/0.0
    Bytes sent by this user: 946
    Bytes received by this user: 0
```

#### Meaning

The **show security firewall-authentication users** command displays the firewall authentication user information for the specified identifier. If the output displays Pass-through using HTTPS in the

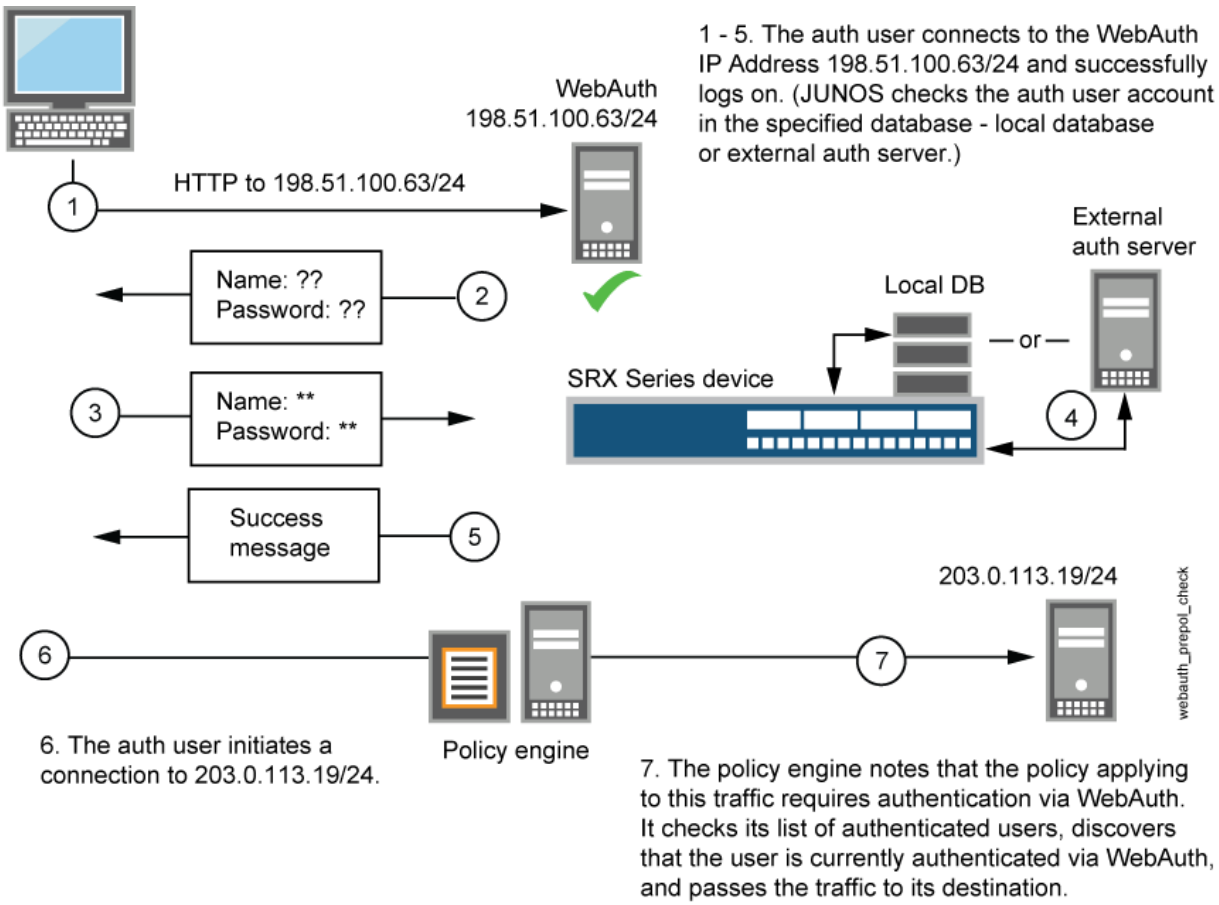
Authentication method field and Success in the Authentication state field, then your configuration is correct.

## Understanding Web Authentication

Web authentication is an alternative to pass-through user authentication. Instead of pointing to the resource that you want to connect to from your client browser, you point the browser to an IP address on the device that is enabled for Web authentication. This initiates an HTTP session to the IP address hosting the Web authentication feature on the device. The device then prompts you for your username and password and caches the result in the device. Later, when traffic encounters a Web authentication policy, you are allowed or denied access based on the prior Web authentication results, as shown in Figure 5 on page 79.

> **NOTE:** You use family inet to assign an IPv4 address. You use family inet6 to assign an IPv6 address. An interface can be configured with both an IPv4 and an IPv6 address. For the sake of brevity, these examples use IPv4 addresses only.

**Figure 5: Web Authentication Example**




Follow these Web authentication guidelines:

- You can leave the default Web authentication server as the local database or you can choose an external authentication server for the role. The default Web authentication profile determines if the user authenticates using the local database or the external authentication server. An access profile stores usernames and passwords of users or points to external authentication servers where such information is stored.
- The Web authentication address must be in the same subnet as the interface that you want to use to host it. For example, if you want authentication users to connect using Web authentication through ethernet3, which has IP address 203.0.113.1/24, then you can assign Web authentication an IP address in the 203.0.113.0/24 subnet.
- You can put a Web authentication address in the same subnet as the IP address of any physical interface or virtual security interface (VSI). (For information about different types of interfaces, see *Security Zones Overview*.)
- You can put Web authentication addresses on multiple interfaces.
- After a device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through Web authentication—from any other user at that

same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.

- With Web authentication enabled, any HTTP traffic to the IP address will get the Web authentication login page instead of the administrator login page. Disabling this option will show the administrator login page (assuming that **[system services web-management HTTP]** is enabled.
- We recommend that you have a separate primary or preferred IP address, if an address is used for Web authentication.

> **NOTE:** The Web authentication method is recommended in situations when the client devices are immediately adjacent to the security gateway and there is high assurance that the client devices are not multiuser hosts. This authentication method is best applied to wireless links and DMZ, or conference room links.

## Example: Configuring Web Authentication

**IN THIS SECTION**


- Requirements | **80**
- Overview | **81**
- Configuration | **82**
- Verification | **87**


This example shows how to enable Web authentication and set up a policy that allows access to a user when traffic encounters a policy that has Web authentication enabled.

### Requirements

Before you begin:

- Define firewall users. See “Firewall User Authentication Overview” on page 35.
- Add the Web authentication HTTP flag under the interface’s address hierarchy to enable Web authentication.

## Overview

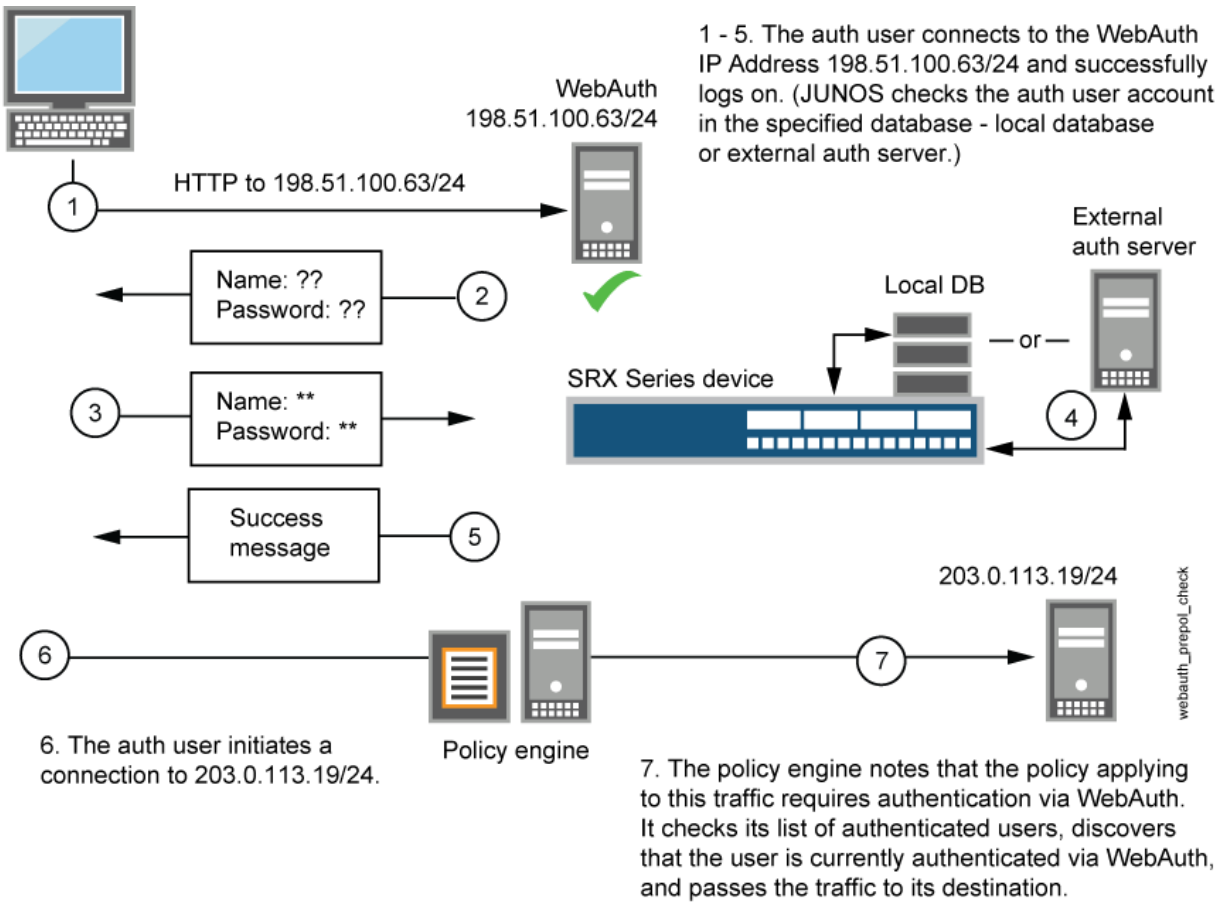
To enable Web authentication, you must specify the IP address of the device hosting the HTTP session. These settings are used if the firewall user accessing a protected resource wants to be authenticated by directly accessing the webserver or by Web authentication. The following instructions show how to set up a policy that allows access to the FWClient1 user when traffic encounters a policy that has Web authentication enabled (Policy-W). (See Figure 6 on page 81.) In this example, FWClient1 has already authenticated through the Web authentication login page.

The FWClient1 firewall user does the following to get authenticated:

a. Points the browser to the Web authentication IP (198.51.100.63/24) to get authenticated first

b. Starts traffic to access resources specified by the policy-W policy

**Figure 6: Web Authentication Example**




When you configure the device as described in these instructions and the user successfully authenticates, the screen illustrated in Figure 7 on page 82 appears.

Figure 7: Web Authentication Success Banner

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

**set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.23/24**

**set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.63/24 web-authentication http**

**set interfaces fe-5/0/0 unit 0 family inet address 203.0.113.15/24**

**set access profile WEBAUTH client FWClient1 firewall-user password pwd**

**set access firewall-authentication web-authentication default-profile WEBAUTH**

**set access firewall-authentication web-authentication banner success "WEB AUTH LOGIN SUCCESS"**

**set security zones security-zone UT-ZONE host-inbound-traffic system-services all**

**set security zones security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic protocols all**

**set security zones security-zone T-ZONE host-inbound-traffic system-services all**

**set security zones security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic protocols all**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match source-address any**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match destination-address any**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match application any**

**set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 then permit firewall-authentication web-authentication client-match FWClient1**

**set system services web-management http interface ge-0/0/1.0**

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure Web authentication:

1. Configure two interfaces and assign IP addresses to them.

   **NOTE:** For this example, it is optional to assign two addresses to the interfaces.

   ```
   [edit]
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.23/24
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.63/24 web-authentication http
   user@host# set interfaces fe-5/0/0 unit 0 family inet address 203.0.113.15/24
   ```

2. Create the WEBAUTH access profile for the FWClient1 user, specify the user's password, and define a success banner.

   ```
   [edit access]
   user@host# set profile WEBAUTH client FWClient1 firewall-user password pwd
   user@host# set firewall-authentication web-authentication default-profile WEBAUTH
   user@host# set firewall-authentication web-authentication banner success "WEB AUTH LOGIN SUCCESS"
   ```

3. Configure security zones.

   **NOTE:** For this example, it is optional to configure a second interface for a security zone.

   ```
   [edit security zones]
   user@host# set security-zone UT-ZONE host-inbound-traffic system-services all
   user@host# set security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic protocols all
   user@host# set security-zone T-ZONE host-inbound-traffic system-services all
   user@host# set security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic protocols all
   ```

4. Assign security policy P1 to the security zones.

   ```
   [edit security policies]
   user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match source-address any
   user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match destination-address any
   ```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match application any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 then permit firewall-authentication
  web-authentication client-match FWClient1
```

5. Activate the HTTP process (daemon) on your device.

```
[edit]
user@host# set system services web-management http interface ge-0/0/1.0
```

**Results**

From configuration mode, confirm your configuration by entering these commands:

- **show interfaces**
- **show access**
- **show security zones**
- **show security policies**
- **show system services**

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

`user@host#` **show interfaces**

```
...
}
ge-0/0/1{
    unit 0 {
        family inet {
            address 198.51.100.23/24 {
            address 198.51.100.63/24 {
                web-authentication http;
            }
        }
    }
}
fe-5/0/0 {
    unit 0 {
```

```
        family inet {
            address 198.51.100.14/24;
        }
    }
}
...
```

user@host# **show access**

```
profile WEBAUTH {
    client FWClient1 {
        firewall-user {
            password "$ABC123"; ## SECRET-DATA
        }
    }
}
firewall-authentication {
    web-authentication {
        default-profile WEBAUTH;
        banner {
            success "WEB AUTH LOGIN SUCCESS";
        }
    }
}
```

user@host# **show security zones**

```
...
}
security-zone UT-ZONE {
    host-inbound-traffic {
       system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                protocols {
                    all;
                }
```

```
            }
        }
    }
}
security-zone T-ZONE {
    host-inbound-traffic {
       system-services {
            all;
        }
    }
    interfaces {
        ge-5/0/0.0 {
            host-inbound-traffic {
                protocols {
                    all;
                }
            }
        }
    }
}
```

user@host# **show security policies**

```
...
from-zone UT-ZONE to-zone T-ZONE {
    policy P1 {
       match {
            source-address any;
            destination-address any;
            application any;
       }
       then {
            permit {
                firewall-authentication {
                    web-authentication {
                        client-match FWClient1;
                    }
                }
            }
       }
    }
}
```

```
user@host# show system services
```

```
...
ftp;
ssh;
telnet;
web-management {
    http {
        interface g-0/0/1.0;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**

- Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table | 87

To confirm that the configuration is working properly, perform this task:

***Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table***

**Purpose**

Display firewall authentication user history and verify the number of firewall users who successfully authenticated and firewall users who failed to log in.

**Action**

From operational mode, enter these **show** commands:

```
user@host> show security firewall-authentication history
```

```
user@host> show security firewall-authentication history identifier 1
```

```
user@host> show security firewall-authentication users
```

user@host> **show security firewall-authentication users identifier 3**

user@host> **show security firewall-authentication history**

```
History of firewall authentication data:
Authentications: 1
Id Source Ip Date Time Duration Status User
5 198.51.100.75      2010-04-24 01:08:57 0:10:30    Success  FWClient1
```

user@host> **show security firewall-authentication history identifier 1**

```
Username: FWClient1
Source IP: 198.51.100.752
Authentication state: Success
Authentication method: Web-authentication
Access start date: 2010-10-12
Access start time: 21:24:02
Duration of user access: 0:00:24
Source zone: N/A
Destination zone: N/A
Access profile: WEBAUTH
Bytes sent by this user: 0
Bytes received by this user: 2660
```

user@host> **show security firewall-authentication users**

```
Firewall authentication data:
Total users in table: 1
Id Source Ip Src zone Dst zone Profile Age Status User
4 198.51.100.75     N/A  N/A   WEBAUTH     1 Success   FWClient1
```

user@host> **show security firewall-authentication users identifier 3**

```
Username: FWClient1
Source IP: 198.51.100.75
Authentication state: Success
Authentication method: Web-authentication
Age: 3
```

```
Access time remaining: 9
Source zone: N/A
Destination zone: N/A
Access profile: WEBAUTH
Interface Name: ge-0/0/1.0
Bytes sent by this user: 0
Bytes received by this user: 1521
```

SEE ALSO

Example: Customizing a Firewall Authentication Banner | 42

*Security Zones Overview*

## Example: Configuring HTTPS Traffic to Trigger Web Authentication

IN THIS SECTION


- Requirements | 89
- Overview | 90
- Configuration | 91
- Verification | 94


This example shows how to configure HTTPS traffic to trigger Web authentication. HTTPS is widely used for Web authentication because it is more secure than HTTP.

### Requirements

Before you begin:

This example uses the following hardware and software components:

- SRX Series device
- Two PCs with Linux and Open SSL installed. One PC acts as a client and another as an HTTPS server. The two PCs are used to create key files and to send traffic.

- Junos OS Release 12.1X44-D10 or later for SRX5400, SRX5600, and SRX5800 devices and Junos OS Release 15.1X49-D40 or later for vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways.

An SRX Series device has to decode the HTTPS traffic to trigger Web authentication. The following list describes the steps to create and install a private key file and a certification key file.

> **NOTE:** If you have an official **.crt** file and **.key** file, then you can directly upload and install the files on the SRX Series device. If you do not have a **.crt** file and **.key** file, then follow the procedure to create and install the files. Instructions specified in Step 1 and Step 2 must be run on a PC which has Linux and OpenSSL installed. Instructions specified in Step 3 and Step 4 must be run in operational mode.

1. From the PC, create the **.key** file.

   ```
   openssl genrsa -out /tmp/server.key 1024
   ```

2. From the PC, create the **.crt** file.

   ```
   openssl req -new -x509 -days 365 -key /tmp/server.key -out /tmp/device.crt -subj
     "/C=CN/ST=BJ/L=BJ/O=JNPR/OU=CNRD/CN=203.0.113.22/emailAddress=device@mycomany.com"
   ```

3. From the SRX Series device, upload the **.key** and **.crt** files and install the files on the device using the following command:

   ```
   user@host> request security pki local-certificate load filename /var/tmp/device.crt key /var/tmp/device.key
     certificate-id device
   ```

## Overview

Firewall authentication initiates a secure connection to be established across two devices. A network user must provide a username and password for authentication when initiating a connection across the firewall. Firewall authentication supports HTTPS traffic for pass-through authentication. HTTPS can secure HTTP firewall authentication traffic between users and the SRX Series device.

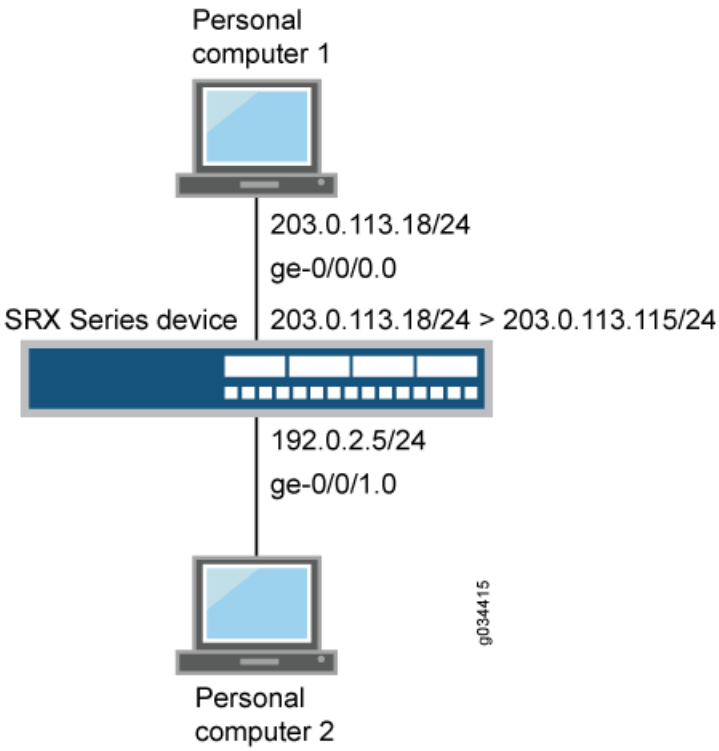
HTTPS is the secure version of HTTP, the protocol over which data is sent between the user and the device that the user is connected to. All communications between the user and the connected devices are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms.

In this example, HTTPS traffic is used to trigger Web authentication because HTTPS is more secure than HTTP.

The user uses HTTPS to access an IP address on the device that is enabled for Web authentication. In this scenario, the user does not use HTTPS to access the IP address of the protected resource. The user is prompted for a username and password, which are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the results of this Web authentication.

Figure 8 on page 91 shows an example of Web authentication using HTTPS traffic.

**Figure 8: Web Authentication Using HTTPS Traffic**




## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set system services web-management https pki-local-certificate device
set interfaces ge-0/0/0 unit 0 family inet address 203.0.113.18/24
set interfaces ge-0/0/0 unit 0 family inet address 203.0.113.115/24 web-authentication https
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.5/24
```

```
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 then permit
set access profile local_pf client user1 firewall-user password user1
set access firewall-authentication web-authentication default-profile local_pf
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication
    web-authentication
```

**Step-by-Step Procedure**

To configure HTTPS traffic to trigger Web authentication:

1. Enable Web-management support to HTTPS traffic.

```
[edit system services]
user@host# set web-management https pki-local-certificate device
```

2. Configure interfaces and assign IP addresses. Enable Web authentication at ge-0/0/0 interface.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 203.0.113.18/24
set ge-0/0/0 unit 0 family inet address 203.0.113.115/24 web-authentication https
user@host# set ge-0/0/1 unit 0 family inet address 192.0.2.5/24
```

3. Configure security policies to permit firewall authenticated traffic from zone trust to zone untrust.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy p1 match source-address any destination-address
    any application any
user@host# set security policies from-zone trust to-zone untrust policy p1 then permit
```

4. Create an access profile, configure the client as a firewall user, and set the password.

```
[edit access]
user@host# set profile local_pf client user1 firewall-user password user1
```

5. Configure the type of firewall authentication settings.

```
[edit access]
```

```
user@host# set firewall-authentication web-authentication default-profile local_pf
```

6. Specify a policy action to take when a packet matches the criteria.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy p1 then permit firewall-authentication
  web-authentication
```

**Results**

From configuration mode, confirm your configuration by entering the **show system services**, **show interfaces**, **show security policies**, and **show access** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show system services
   web-management {
   https {
   pki-local-certificate device;
   }
}
```

```
user@host# show interfaces
   ge-0/0/0 {
   unit 0 {
      family inet {
         address 203.0.113.115/24 {
            web-authentication https;
         }
      }
}
   ge-0/0/1 {
      unit 0 {
         family inet {
            address 192.0.2.5/24;
         }
      }
   }
```

```
user@host# show security policies
from-zone trust to-zone untrust {
   policy p1 {
```

```
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit {
                firewall-authentication {
                    web-authentication;
                }
            }
        }
    }
}
```

```
user@host# show access
    profile local_pf {
        client user1 {
            firewall-user {
                password "user1";
            }
        }
    }
        firewall-authentication {
            web-authentication {
                default-profile local_pf;
            }
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

### *Verifying the Configuration*

#### Purpose

Verify that the configuration is correct.

#### Action

From operational mode, enter the **show security firewall-authentication users identifier *identifier*** command.

## Sample Output

```
user@host> show security firewall-authentication users identifier 1
   Username: user1
   Source IP: 203.1.113.102
   Authentication state: Success
   Authentication method: Web-authentication
   Age: 0
   Access time remaining: 10
   Lsys: root-logical-system
   Source zone: N/A
   Destination zone: N/A
   Access profile: local_pf
   Bytes sent by this user: 0
   Bytes received by this user: 0
```

**Meaning**

The **show security firewall-authentication users identifier *identifier*** command displays the firewall authentication user information using the identifier ID of the user. If the authentication method parameter displays Web authentication and the authentication state parameter displays success in your output then your configuration is correct.

# Encrypt Traffic Using SSL Proxy and TLS

**IN THIS SECTION**


- SSL Proxy Overview | **96**
- Configuring SSL Forward Proxy | **99**
- Enabling Debugging and Tracing for SSL Proxy | **110**
- Transport Layer Security (TLS) Overview | **112**
- Configuring the TLS Syslog Protocol on SRX Series device | **114**


SSL proxy acts as an intermediary, performing SSL encryption and decryption between the client and the server. Better visibility into application usage can be made available when the SSL forward proxy is enabled.

## SSL Proxy Overview

SSL proxy is supported on SRX Series devices only.

Secure Sockets Layer (SSL) is an application-level protocol that provides encryption technology for the Internet. SSL, also called Transport Layer Security (TLS), ensures the secure transmission of data between a client and a server through a combination of privacy, authentication, confidentiality, and data integrity. SSL relies on certificates and private-public key exchange pairs for this level of security.

Server authentication guards against fraudulent transmissions by enabling a Web browser to validate the identity of a webserver. Confidentiality mechanisms ensure that communications are private. SSL enforces confidentiality by encrypting data to prevent unauthorized users from eavesdropping on electronic communications. Finally, message integrity ensures that the contents of a communication have not been tampered with.

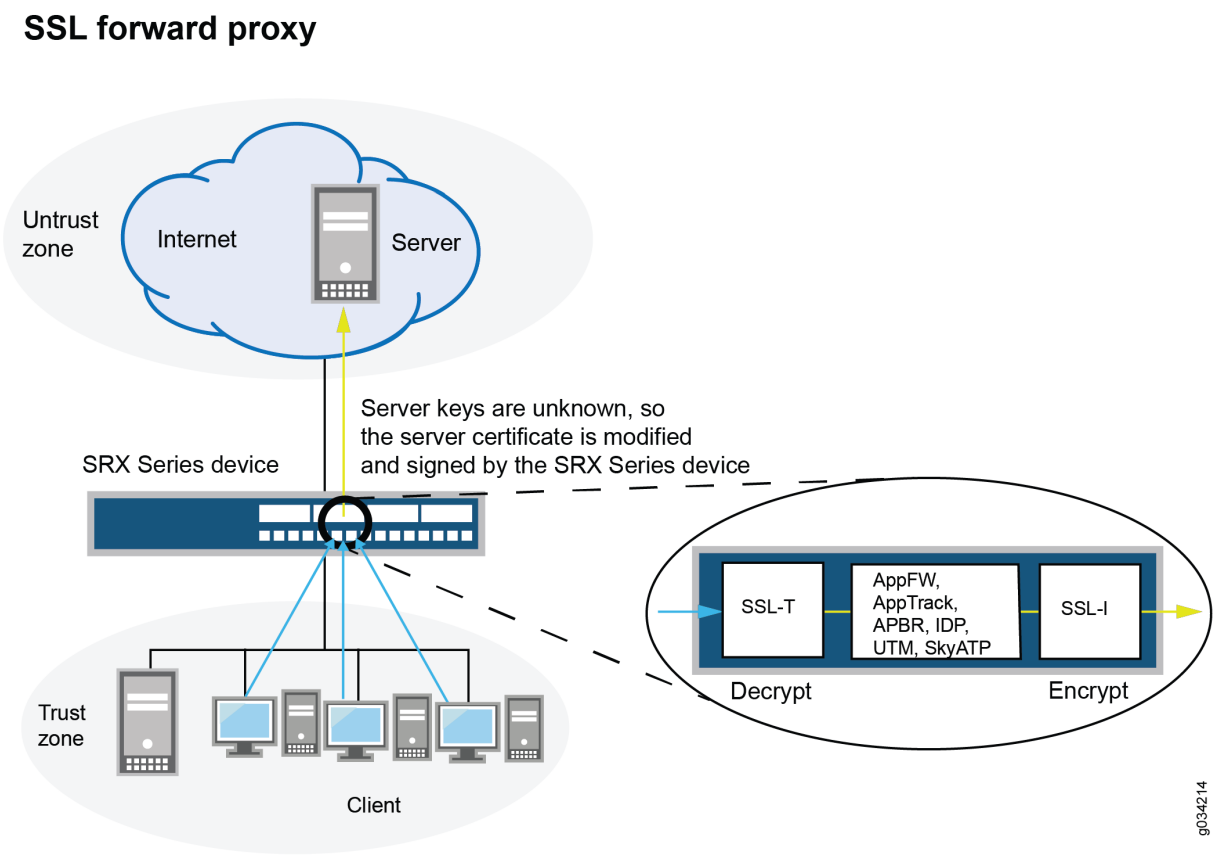
SSL proxy is transparent; that is, it performs SSL encryption and decryption between the client and the server.

Sharing server keys is sometimes not feasible or might not be available in certain circumstances, in which case the SSL traffic cannot be decrypted. SSL proxy addresses this problem by ensuring that it has the keys to encrypt and decrypt the payload:

- For the server, SSL proxy acts as a client—Because SSL proxy generates the shared pre-master key, it determines the keys to encrypt and decrypt.
- For the client, SSL proxy acts as a server—SSL proxy first authenticates the original server and replaces the public key in the original server certificate with a key that is known to it. It then generates a new certificate by replacing the original issuer of the certificate with its own identity and signs this new certificate with its own public key (provided as a part of the proxy profile configuration). When the client accepts such a certificate, it sends a shared pre-master key encrypted with the public key on the certificate. Because SSL proxy replaced the original key with its own key, it is able to receive the shared pre-master key. Decryption and encryption take place in each direction (client and server), and the keys are different for both encryption and decryption.

Figure 9 on page 97 shows how SSL proxy works on an encrypted payload.

Figure 9: SSL Proxy on an Encrypted Payload




When Advanced Security services such as application firewall (AppFW), Intrusion Detection and Prevention (IDP),application tracking (AppTrack), UTM, and SkyATP is configured, the SSL proxy acts as an SSL server by terminating the SSL session from the client and establishing a new SSL session to the server. The SRX Series device decrypts and then reencrypts all SSL proxy traffic. SSL proxy uses the following:

- SSL-T-SSL terminator on the client side.
- SSL-I-SSL initiator on the server side.

IDP, AppFW, AppTracking, advanced policy-based routing (APBR), UTM, SkyATP, and ICAP service redirect can use the decrypted content from SSL proxy. If none of these services are configured, then SSL proxy services are bypassed even if an SSL proxy profile is attached to a firewall policy.

### Benefits of SSL Proxy

- Decrypts SSL traffic to obtain granular application information and enable you to apply advanced security services protection and detect threats.
- Enforces the use of strong protocols and ciphers by the client and the server.

- Provides visibility and protection against threats embedded in SSL encrypted traffic.
- Controls what needs to be decrypted by using Selective SSL Proxy.

### Logical Systems Support

It is possible to enable SSL proxy on firewall policies that are configured using logical systems; however, note the following limitations:

- The “services” category is currently not supported in logical systems configuration. Because SSL proxy is under “services,” you cannot configure SSL proxy profiles on a per-logical-system basis.
- Because proxy profiles configured at a global level (within “services ssl proxy”) are visible across logical system configurations, it is possible to configure proxy profiles at a global level and then attach them to the firewall policies of one or more logical systems.

### Limitations

**NOTE:** On SRX Series devices, for a particular session, the SSL proxy is only enabled if a relevant feature related to SSL traffic is also enabled. Features that are related to SSL traffic are IDP, application identification, application firewall, application tracking, advanced policy-based routing, UTM, SkyATP, and ICAP redirect service. If none of these features are active on a session, the SSL proxy bypasses the session and logs are not generated in this scenario.

**NOTE:** On all SRX Series devices, the current SSL proxy implementation has the following connectivity limitations:

- The SSLv3.0 protocol support is deprecated.
- The SSLv2 protocol is not supported. SSL sessions using SSLv2 are dropped.
- Only X.509v3 certificate is supported.
- Client authentication of SSL handshake is not supported.
- SSL sessions where client certificate authentication is mandatory are dropped.
- SSL sessions where renegotiation is requested are dropped.

SEE ALSO

*Managing Certificates and Keys for SSL Proxy*

*Configuring SSL Proxy*

*Unified Policies for SSL Proxy*

*ICAP Service Redirect*

*SSL Decryption Mirroring*

*SSL Proxy Logs*

# Configuring SSL Forward Proxy

**IN THIS SECTION**


- SSL Proxy Configuration Overview | **99**
- Configuring a Root CA Certificate | **100**
- Generate a Root CA Certificate with CLI | **101**
- Generate a Root CA Certificate with OpenSSL | **102**
- Configuring a CA Profile Group | **103**
- Importing a Root CA Certificate into a Browser | **105**
- Applying an SSL Proxy Profile to a Security Policy | **106**
- Configuring SSL Proxy Logging | **107**
- Configuring Certificate Authority Profiles | **108**
- Exporting Certificates to a Specified Location | **109**
- Ignoring Server Authentication | **110**

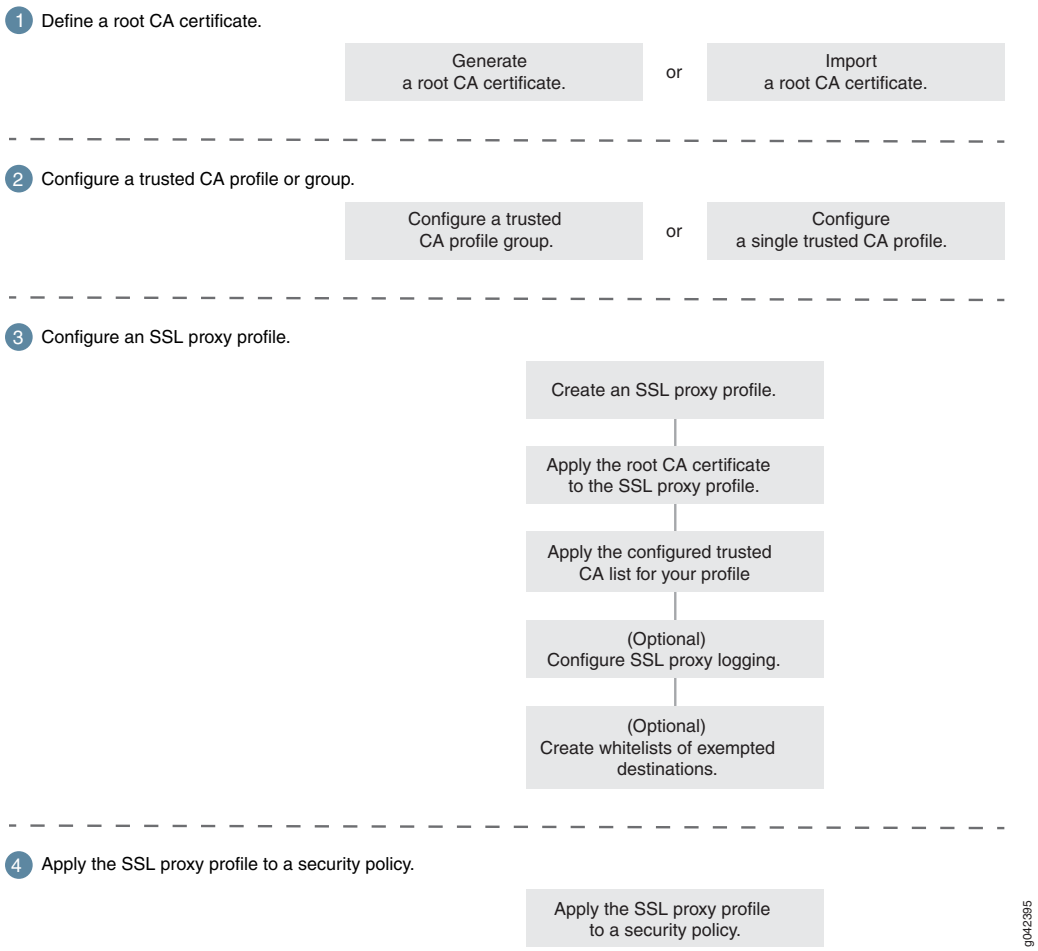

## SSL Proxy Configuration Overview

Figure 10 on page 100 displays an overview of how SSL proxy is configured. Configuring SSL proxy includes:

- Configuring the root CA certificate
- Loading a CA profile group
- Configure SSL proxy profile and associate root CA certificate and CA profile group
- Create a security policy by defining input traffic match criteria

- Applying an SSL proxy profile to a security policy
- Optional steps such as creating whitelists and SSL proxy logging

**Figure 10: SSL Proxy Configuration Overview**




## Configuring a Root CA Certificate

A CA can issue multiple certificates in the form of a tree structure. A root certificate is the topmost certificate of the tree, the private key of which is used to *sign* other certificates. All certificates immediately below the root certificate inherit the signature or trustworthiness of the root certificate. This is somewhat like the *notarizing* of an identity.

You can configure a root CA certificate by first obtaining a root CA certificate (by either generating a self-signed one or importing one) and then applying it to an SSL proxy profile. There are two ways you

can obtain a root CA certificate—by using the Junos OS CLI on an SRX Series device or by using OpenSSL on a UNIX device.

### Generate a Root CA Certificate with CLI

To define a self-signed certificate in CLI, you must provide the following details:

- Certificate identifier (generated in the previous step)
- Fully qualified domain name (FQDN) for the certificate
- e-mail address of the entity owning the certificate
- Common name and the organization involved

Generate a root CA certificate using the Junos OS CLI:

1. From operational mode, generate a PKI public/private key pair for a local digital certificate.

   ```
   user@host>request security pki generate-key-pair certificate-id certificate-id size size type type
   ```

   Example:

   ```
   user@host>request security pki generate-key-pair certificate-id SECURITY-cert size 2048 type ecdsa
   ```

2. Define a self-signed certificate.

   ```
   user@host>request security pki local-certificate generate-self-signed certificate-id certificate-id domain-name
     domain-name subject subject email email-id add-ca-constraint
   ```

   Example:

   ```
   user@host> request security pki local-certificate generate-self-signed certificate-id SECURITY-cert
     domain-name labs.abc.net subject DC=mydomain.net,L=Sunnyvale,O=Mydomain,OU=LAB,CN=SECURITY
     email lab@labs.abc.net add-ca-constraint
   ```

   By configuring the **add-ca-constraint** option, you make sure that the certificate can be used for signing other certificates.

3. From configuration mode, apply the loaded certificate as root-ca in the SSL proxy profile.

   ```
   [edit]
   user@host# set services ssl proxy profile profile-name root-ca certificate-id
   ```

   Example:

```
[edit]
user@host# set services ssl proxy profile SECURITY-SSL-PROXY root-ca SECURITY-cert
```

4. Import the root CA as a trusted CA into client browsers. This is required for the client browsers to trust the certificates signed by the SRX Series device. See "Importing a Root CA Certificate into a Browser" on page 105.

### Generate a Root CA Certificate with OpenSSL

To generate a root CA certificate using OpenSSL:

1. Create folders **keys** and **certs**.

```
mkdir /etc/pki/tls/keys
mkdir /etc/pki/tls/certs
```

2. Change to the **openssl** directory.

```
cd /etc/pki/tls
```

3. Create a CA certificate key.

```
% openssl genrsa -des3 -out keys/ssl-proxy-ca.key 2048
```

This step creates an RSA key using the 3DES encryption named **ca.key** that is 2048 in length. You also need to enter a password that is used to encrypt the private key. This is critical to security if the key is lost because it will still be encrypted.

4. Create a CA certificate based on the CA private key (created in the previous step).

```
% openssl req -new -x509 -days 1095 –key keys/ssl-proxy-ca.key -out certs/ssl-inspect-ca.cer
```

The expiration date for this certificate is 3 years or 1095 days. However, you can set it to a different value. When creating the certificate, you need to enter the password and the certificate information that includes distinguished name (DN), country name, and so forth.

5. Import the CA private and public keys into the SRX Series device. Copy the **ca.key** and **ca.cer** keys to the **/var/tmp** directory on the SRX Series device. You can copy using SCP, or open the files and copy them into "vi" on the SRX Series device to create new files.

```
user@host> request security pki local-certificate load certificate-id ssl-inspect-ca key /var/tmp/ssl-proxy-ca.key
  filename /var/tmp/ssl-inspect-ca.cer passphrase password
```

6. From configuration mode, apply the loaded certificate as root-ca in the SSL proxy profile.

```
[edit]
user@host# set services ssl proxy profile ssl-inspect-profile root-ca ssl-inspect-ca
```

7. Import the root CA as a trusted CA into client browsers. This is required for the client browsers to trust the certificates signed by the SRX Series device. See "Importing a Root CA Certificate into a Browser" on page 105.

### Configuring a CA Profile Group

The CA profile defines the certificate information for authentication. It includes the public key that SSL proxy uses when generating a new certificate. Junos OS allows you to create a group of CA profiles and load multiple certificates in one action, view information about all certificates in a group, and delete unwanted CA groups.

You can load a group of CA profiles by obtaining a list of trusted CA certificates, defining a CA group, and attaching the CA group to the SSL proxy profile.

1. Obtain a list of trusted CA certificates by using one of the following methods:

   - Junos OS provides a default list of trusted CA certificates that you can load on your system. The Junos OS package contains the default CA certificates as a PEM file (for example, **trusted_CA.pem**). After you download the Junos OS package, the default certificates are available on your system.

     From operational mode, load the default trusted CA certificates (the group name identifies the CA profile group):

     ```
     user@host> request security pki ca-certificate ca-profile-group load ca-group-name group-name filename
       default
     ```

     Example:

     ```
     user@host> request security pki ca-certificate ca-profile-group load ca-group-name SECURITY-CA-GROUP
       filename default
     ```

   - Alternatively, you can define your own list of trusted CA certificates and import them on your system. You get the list of trusted CAs in a single PEM file (for example **IE-all.pem**) and save the PEM file in a specific location (for example, **/var/tmp**). See Knowledge Base Article KB23144.

From operational mode, load the trusted list to the device (the group name identifies the CA profile group):

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name group-name filename
  /var/tmp/IE-all.pem
```

Example:

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name SECURITY-CA-GROUP
  filename /var/tmp/custom-file.pem
```

2. Attach the trusted CA or trusted CA group to the SSL proxy profile. You can attach all trusted CA or one trusted CA at a time:

   - Attach all CA profile groups:

     ```
     [edit]
     user@host# set services ssl proxy profile profile-name trusted-ca all
     ```

     Example

     ```
     [edit]
     user@host# set services ssl proxy profile SECURITY-SSL-PROXY trusted-ca all
     ```

   - Attach one CA profile group (the group name identifies the CA profile group).

     ```
     [edit]
     user@host# set services ssl proxy profile profile-name trusted-ca ca-name
     ```

     Example

     ```
     [edit]
     user@host# set services ssl proxy profile SECURITY-SSL-PROXY trusted-ca orgA-ca-profile
     ```

You can easily display information about all certificates in a CA profile group:

```
user@host> show security pki ca-certificates ca-profile-group group-name
```

You can delete a CA profile group. Remember that deleting a CA profile group deletes all certificates that belong to that group:

```
user@host> clear security pki ca-certificates ca-profile-group group-name
```

### Importing a Root CA Certificate into a Browser

In order to have your browser or system automatically trust all certificates signed by the root CA configured in the SSL proxy profile, you must instruct your platform or browser to trust the CA root certificate.

To import a root CA certificate:

1. Generate a PEM format file for the configured root CA.

   **request security pki local-certificate export certificate-id *root-ca* type pem filename *path/file-name.pem***

2. Import a root CA certificate into a browser.

   From Internet Explorer (version 8.0):

   a. From the Tools menu, select **Internet Options**.

   b. On the Content tab, click **Certificates**.

   c. Select the **Trusted Root Certification Authorities** tab and click **Import**.

   d. In the Certificate Import Wizard, navigate to the required root CA certificate and select it.

   From Firefox (version 39.0):

   a. From the Tools menu, select **Options**.

   b. From the Advanced menu, select the **Certificates** tab and click **View Certificate**.

   c. In the Certificate Manager window, select the **Authorities** tab and click **Import**.

   d. Navigate to the required root CA certificate and select it.

   From Google Chrome (45.0):

   a. From the Settings menu, select **Show Advanced Settings**.

   b. From the Advanced menu, select the **Certificates** tab and click **View Certificate**.

   c. Under HTTPS/SSL, click **Manage Certificates**.

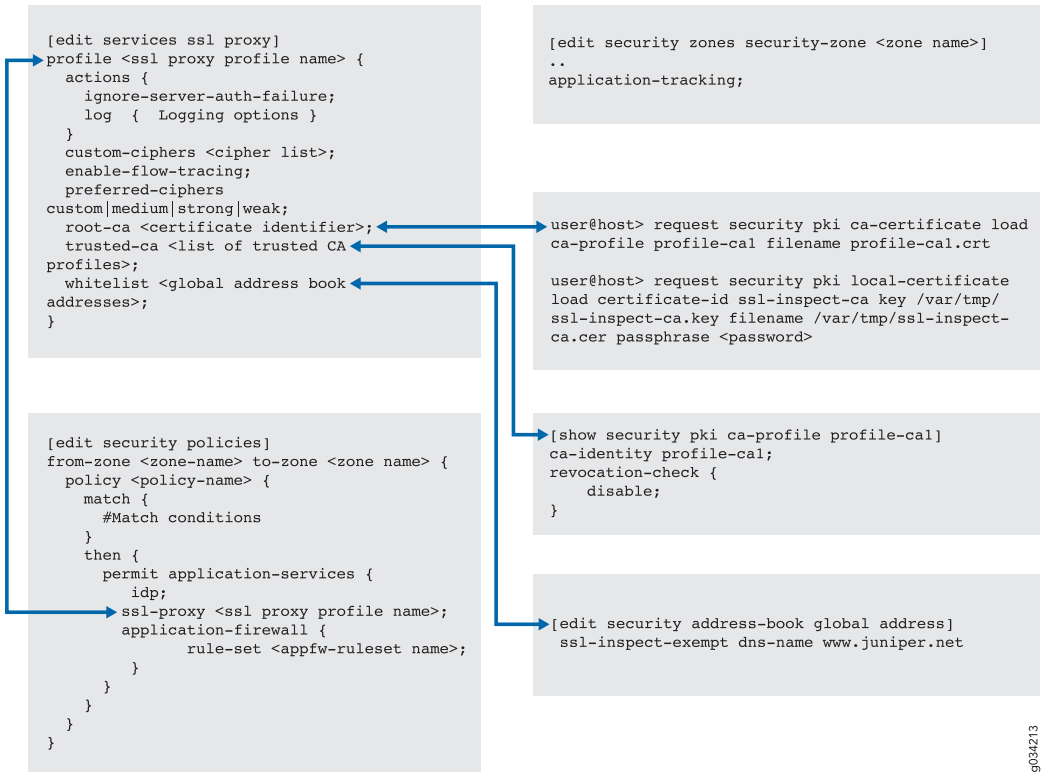
d. In the Certificate window, select **Trusted Root Certification Authorities** and click **Import**.

e. In the Certificate Import Wizard, navigate to the required root CA certificate and select it.

## Applying an SSL Proxy Profile to a Security Policy

SSL proxy is enabled as an application service within a security policy. In a security policy, you specify the traffic that you want the SSL proxy enabled on as match criteria and then specify the SSL proxy CA profile to be applied to the traffic. Figure 11 on page 106 displays a graphical view of SSL proxy profile and security policy configuration.

**Figure 11: Applying an SSL Proxy Profile to a Security Policy**




To enable SSL proxy in a security policy:

This example assumes that you have already creates security zones trust and untrust and creating a security policy for the traffic from trust zone to untrust zone.

1. Create a security policy and specify the match criteria for the policy. As match criteria, specify the traffic for which you want to enable SSL proxy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy policy-name match source-address
    source-address
user@host# set security policies from-zone trust to-zone untrust policy policy-name match destination-address
    destination-address
user@host# set security policies from-zone trust to-zone untrust policy policy-name match application
    application
```

Example:

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy SECURITY_POLICY match
    source-address any
user@host# set security policies from-zone trust to-zone untrust policy policy SECURITY_POLICY match
    destination-address any
user@host# set security policies from-zone trust to-zone untrust policy policy SECURITY_POLICY match
    application any
```

2. Apply the SSL proxy profile to the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy policy SECURITY_POLICY then
    permit application-services ssl-proxy profile-name SECURITY-SSL-PROXY
```

### Configuring SSL Proxy Logging

When configuring SSL proxy, you can choose to set the option to receive some or all of the logs. SSL proxy logs contain the logical system name, SSL proxy whitelists, policy information, SSL proxy information, and other information that helps you troubleshoot when there is an error.

You can configure logging of *all* or specific events, such as error, warning, and information events. You can also configure logging of sessions that are whitelisted, dropped, ignored, or allowed after an error occurs.

```
[edit]
user@host# set services ssl proxy profile profile-name actions log all
user@host# set services ssl proxy profile profile-name actions log sessions-whitelisted
user@host# set services ssl proxy profile profile-name actions log sessions-allowed
user@host# set services ssl proxy profile profile-name actions log errors
```

You can use **enable-flow-tracing** option to enable debug tracing.

### Configuring Certificate Authority Profiles

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on an SRX Series device. For example, you might have one profile for orgA and one for orgB. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one then create a new CA profile (for example, Microsoft-2008). You can group multiple CA profiles in one trusted CA group for a given topology.

In this example, you create a CA profile called ca-profile-security with CA identity microsoft-2008. You then create proxy profile to the CA profile.

1. From configuration mode, configure the CA profile used for loading the certificate.

   ```
   [edit]
   user@host# set security pki ca-profile profile-name ca-identity ca-identity
   ```

   Example:

   ```
   user@host# set security pki ca-profile ca-profile-security ca-identity example.com
   ```

2. Commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

3. From operational mode, load the certificate using PKI commands.

   ```
   user@host> request security pki ca-certificate load ca-profile profile-name filename filename
   ```

   Example:

   ```
   user@host> request security pki ca-certificate load ca-profile ca-profile-security filename srx-123.crt
   ```

4. From configuration mode, disable the revocation check (if required).

   ```
   [edit]
   user@host# set security pki ca-profile profile-name ca-identity ca-identity revocation-check disable
   ```

   Example:

```
[edit]
user@host# set security pki ca-profile ca-profile-security ca-identity example.com revocation-check disable
```

5. From configuration mode, configure the loaded certificate as a trusted CA in the SSL proxy profile.

```
[edit]
user@host# set services ssl proxy profile ssl-proxy-profile-name trusted-ca ca-profile-name
```

Example:

```
[edit]
user@host# set services ssl proxy profile ssl-proxy-sample trusted-ca ca-profile-security
```

> **NOTE:** More than one trusted CA can be configured for a profile.

6. (Optional) If you have multiple trusted CA certificates, you do not have to specify each trusted CA separately. You can load *all* the trusted CA certificates using the following command from configuration mode.

```
[edit]
user@host# set services ssl proxy profile ssl-proxy-profile-name root-ca ssl-inspect-ca
user@host# set services ssl proxy profile ssl-proxy-profile-name trusted-ca all
```

> **NOTE:** Alternatively, you can import a set of trusted CAs from your browser into the SRX Series device. See Knowledge Base article KB23144.

### Exporting Certificates to a Specified Location

When a self-signed certificate is generated using a PKI command, the newly generated certificate is stored in a predefined location (**var/db/certs/common/local**).

Use the following command to export the certificate to a specific location (within the device). You can specify the certificate ID, the filename, and the type of file format (DER/PEM):

```
user@host> request security pki local-certificate export certificate-id certificate-id filename filename  type der
```

### Ignoring Server Authentication

Junos OS allows you to configure an option to ignore server authentication completely. If you configure your system to ignore authentication, then any errors encountered during server certificate verification at the time of the SSL handshake are ignored. Commonly ignored errors include the inability to verify CA signature, incorrect certificate expiration dates, and so forth. If this option is not set, all the sessions where the server sends self-signed certificates are dropped when errors are encountered.

We do not recommend using this option for authentication because configuring it results in websites not being authenticated at all. However, you can use this option to effectively identify the root cause of dropped SSL sessions.

From configuration mode, specify to ignore server authentication:

```
[edit]
user@host# set services ssl proxy profile profile-name actions ignore-server-auth-failure
```

SEE ALSO

- *Understanding Certificates and PKI*
- *Understanding Self-Signed Certificates*
- *show services ssl proxy statistics*
- *clear services ssl proxy statistics*

## Enabling Debugging and Tracing for SSL Proxy

Debug tracing on both Routing Engine and the Packet Forwarding Engine can be enabled for SSL proxy by setting the following configuration:

```
user@host# set services ssl traceoptions file file-name
```

SSL proxy is supported on SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, SRX5800 devices and vSRX instances. Table 3 on page 111 shows the supported levels for trace options.

**Table 3: Trace Levels**

| Cause Type | Description |
|---|---|
| Brief | Only error traces on both the Routing Engine and the Packet Forwarding Engine. |
| Detail | Packet Forwarding Engine–Only event details up to the handshake should be traced.<br><br>Routing Engine–Traces related to commit. No periodic traces on the Routing Engine will be available |
| Extensive | Packet Forwarding Engine–Data transfer summary available.<br><br>Routing Engine–Traces related to commit (more extensive). No periodic traces on the Routing Engine will be available. |
| Verbose | All traces are available. |

Table 4 on page 111 shows the flags that are supported.

**Table 4: Supported Flags in Trace**

| Cause Type | Description |
|---|---|
| cli-configuration | Configuration-related traces only. |
| initiation | Enable tracing on the SSL-I plug-in. |
| proxy | Enable tracing on the SSL-Proxy-Policy plug-in. |
| termination | Enable tracing on the SSL-T plug-in. |
| selected-profile | Enable tracing only for profiles that have **enable-flow-tracing** set. |

You can enable logs in the SSL proxy profile to get to the root cause for the drop. The following errors are some of the most common:

- Server certification validation error. Check the trusted CA configuration to verify your configuration.
- System failures such as memory allocation failures.
- Ciphers do not match.
- SSL versions do not match.

- SSL options are not supported.
- Root CA has expired. You need to load a new root CA.

You can enable the **ignore-server-auth-failure** option in the SSL proxy profile to ensure that certificate validation, root CA expiration dates, and other such issues are ignored. If sessions are inspected after the **ignore-server-auth-failure** option is enabled, the problem is localized.

SEE ALSO

traceoptions (Services SSL) | **651**

## Transport Layer Security (TLS) Overview

**IN THIS SECTION**


- Benefits of TLS | **113**
- TLS Versions | **113**
- Three Essential Services of TLS | **113**
- TLS Handshake | **114**
- Encrypting Syslog Traffic with TLS | **114**


Transport Layer Security (TLS) is an application-level protocol that provides encryption technology for the Internet. TLS relies on certificates and private-public key exchange pairs for this level of security. It is the most widely used security protocol for the applications that require data to be securely exchanged over a network, such as file transfers, VPN connections, instant messaging, and voice over IP (VoIP).

TLS protocol is used for certificate exchange, mutual authentication, and negotiating ciphers to secure the stream from potential tampering and eavesdropping. TLS is sometimes called as Secure Sockets Layer (SSL). TLS and SSL are not interoperable, though TLS currently provides some backward compatibility.

SRX Series devices provides TLS inspection that use the TLS protocol suite consisting of different TLS versions, ciphers, and key exchange methods. TLS inspection feature enables SRX Series devices to inspect HTTP traffic encrypted in TLS on any port.

### Benefits of TLS

- TLS ensures the secure transmission of data between a client and a server through a combination of privacy, authentication, confidentiality, and data integrity.

### TLS Versions

Following are the versions of TLS:

- TLS version 1.0—Provides secure communication over networks by providing privacy and data integrity between communicating applications
- TLS version 1.1—This enhanced version of TLS provides protection against cipher-block chaining (CBC) attacks.
- TLS version 1.2 — This enhanced version of TLS provides improved flexibility for negotiation of cryptographic algorithms.

  Starting with Junos OS Release 12.3X48-D30, SRX Series devices support TLS version 1.2. SRX Series devices running earlier release of 12.3X48-D30, supports TLS version 1.0.

### Three Essential Services of TLS

The TLS protocol is designed to provide three essential services to the applications running above it: encryption, authentication, and data integrity.

- **Encryption**—In order to establish a cryptographically secure data channel, the server and the client must agree on which cipher suites are used and the keys used to encrypt the data. The TLS protocol specifies a well-defined handshake sequence to perform this exchange. TLS uses public key cryptography, which allows the client and server to negotiate a shared secret key without having to establish any prior knowledge of each other, and to do so over an unencrypted channel.
- **Authentication**—As part of the TLS handshake, the protocol allows both server and the client to authenticate their identity. Implicit trust between the client and the server (because the client accepts the certificate generated by the server) is an important aspect of TLS. It is extremely important that server authentication is not compromised; however, in reality, self- signed certificates and certificates with anomalies are in abundance. Anomalies can include expired certificates, instances of common name not matching a domain name, and so forth.
- **Integrity**—With encryption and authentication in place, the TLS protocol does message framing mechanism and signs each message with a Message Authentication Code (MAC). The MAC algorithm does the effective checksum, and the keys are negotiated between the client and the server.

### TLS Handshake

Each TLS session begins with a handshake during which the client and server agree on the specific security key and the encryption algorithms to use for that session. At this time, the client also authenticates the server. Optionally, the server can authenticate the client. Once the handshake is complete, transfer of encrypted data can begin.

### Encrypting Syslog Traffic with TLS

TLS protocol ensures the syslog messages are securely sent and received over the network. TLS uses certificates to authenticate and encrypt the communication. The client authenticates the server by requesting its certificate and public key. Optionally, the server can also request a certificate from the client, thus mutual authentication is also possible.

A certificate on the server that identifies the server and the certificate of certificate authority (CA) issued by the server must be available with the client for TLS to encrypt syslog traffic.

For mutual authentication of client and the server, a certificate with the client that identifies the client and the certificate of CA issued by client must be available on the server. Mutual authentication ensures that the syslog server accepts log messages only from authorized clients.

SEE ALSO

ssl (Services) | **618**

*initiation (Services)*

## Configuring the TLS Syslog Protocol on SRX Series device

**IN THIS SECTION**

- Requirements | **115**
- Overview | **115**
- Configuration | **115**
- Verification | **118**

This example shows how to configure the Transport Layer Security (TLS) syslog protocol on SRX Series devices to receive encrypted syslog events from network devices that support TLS syslog event forwarding.

## Requirements

Before you begin, enable server certificate verification and encryption or decryption capabilities.

## Overview

TLS syslog protocol enables log sources to receive encrypted syslog events from network devices that supports TLS syslog event forwarding. The log source creates a listen port for incoming TLS syslog events and generates a certificate file for the network devices.

In this example, you will configure a syslog collector associated with one SSL-I profile. Each SSL-I profile will enable the user to specify things like preferred ciphers suite and trusted CA certificates. Multiple SSL-I profiles can be configured and associated to different collector servers.

## Configuration

### CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security log mode stream
set security log format sd-syslog
set security log source-interface ge-0/0/1.0
set security log transport protocol tls
set security log transport tls-profile ssl-i-tls
set security log stream server1 format sd-syslog
set security log stream server1 category all
set security log stream server1 host 192.0.2.100
set services ssl initiation profile ssl-i-tls protocol-version all
set services ssl initiation profile ssl-i-tls trusted-ca all
set services ssl initiation profile ssl-i-tls actions ignore-server-auth-failure
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure TLS syslog protocol:

1. Set the log mode to stream.

```
[edit security]
user@host# set log mode stream
```

2. Set the format for remote security message logging to sd-syslog (structured system log).

```
[edit security]
user@host# set log format sd-syslog
```

3. Set the host source interface number.

```
[edit security]
user@host# set log source-interface ge-0/0/1.0
```

4. Set security log transport protocol tls to be used to log the data.

```
[edit security]
user@host# set log transport protocol tls
```

5. Specify the TLS profile name.

```
[edit security]
user@host# set log transport tls-profile ssl-i-tls
```

6. Set the log stream to use the structured syslog format for sending logs to server 1.

```
[edit security]
user@host# set log stream server1 format sd-syslog
```

7. Set the category of server 1 logging to all .

```
[edit security]
user@host# set log stream server1 category all
```

8. Set server host parameters by entering the server name or IP address.

```
[edit security]
```

```
user@host# set log stream server1 host 192.0.2.100
```

9. Define the protocol version all for SSL initiation access profile.

```
[edit services]
user@host# set ssl initiation profile ssl-i-tls protocol-version all
```

10. Attach all CA profile groups to the SSL initiation profile to use when requesting a certificate from the peer.

```
[edit services]
user@host# set ssl initiation profile ssl-i-tls trusted-ca all
```

11. Define the SSL initiation access profile to ignore the server authentication failure.

```
[edit services]
user@host# set ssl initiation profile ssl-i-tls actions ignore-server-auth-failure
```

**Results**

From configuration mode, confirm your configuration by entering the **show security log** command. If the output does not display the intended configuration, then repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security log
    mode stream;
        format sd-syslog;
        source-interface ge-0/0/1.0;
        transport {
            protocol tls;
            tls-profile ssl-i-tls;
        }
        stream server1 {
            format sd-syslog;
            category all;
            host {
            192.0.2.100;
            }
        }
```

```
    }
```

```
[edit]
user@host# run show configuration services ssl initiation
    profile ssl-i-tls {
        protocol-version all;
        trusted-ca all;
        actions {
            ignore-server-auth-failure;
        }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To verify that the configuration is working properly, enter the **show log** command on the syslog server.

SEE ALSO

- tls-type | 637
- tls-timeout | 636
- tls-min-version | 634

3

CHAPTER

# Unified Access Control with IC Series UAC Appliance


Configure Unified Access Control in Junos OS | **121**

Set Up Communication between Junos OS Enforcer and IC Series UAC Appliance | **124**

Enforce Policies and Configure Endpoint Security with Junos OS Enforcer | **138**

Configure Captive Portal on Junos OS Enforcer | **143**

Classify Traffic Based on User Roles | **151**

# Configure Unified Access Control in Junos OS

IN THIS SECTION


- Understanding UAC in a Junos OS Environment | 121
- Enabling UAC in a Junos OS Environment (CLI Procedure) | 123


A Unified Access Control (UAC) uses IC Series UAC Appliances, Infranet Enforcers, and Infranet agents to protect your network by ensuring only valid users can access the resources.

## Understanding UAC in a Junos OS Environment

**NOTE:** Beginning on August 1, 2015, all Junos Pulse software and hardware products will be sold and supported by Pulse Secure. To make the transition as seamless as possible, and to provide support for Juniper customers and partners, please visit https://www.juniper.net/us/en/pulsesecure/.

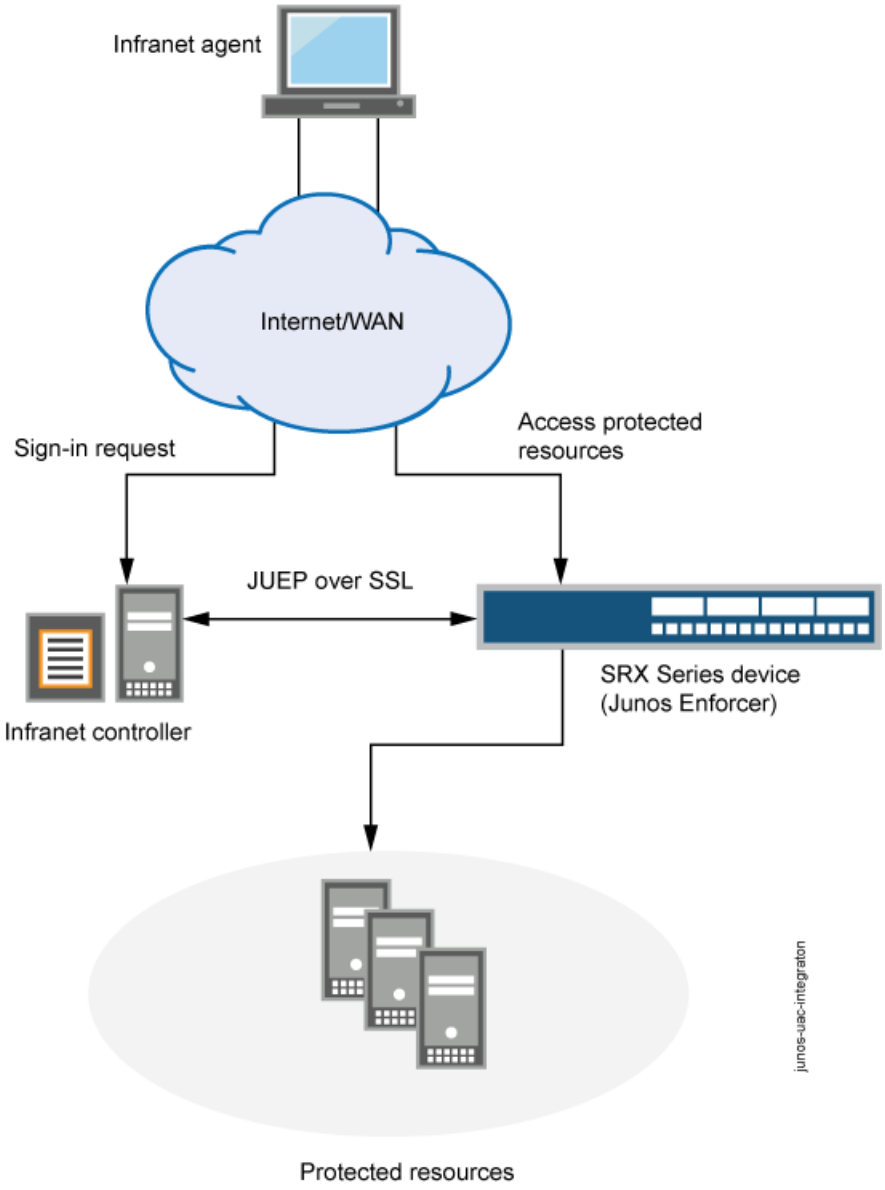
A Unified Access Control (UAC) deployment uses the following components to secure a network and ensure that only qualified end users can access protected resources:

- IC Series UAC Appliances—An IC Series appliance is a policy decision point in the network. It uses authentication information and policy rules to determine whether or not to provide access to individual resources on the network. You can deploy one or more IC Series appliances in your network.
- Infranet Enforcers—An Infranet Enforcer is a policy enforcement point in the network. It receives policies from the IC Series appliance and uses the rules defined in those policies to determine whether or not to allow an endpoint access to a resource. You deploy the Infranet Enforcers in front of the servers and resources that you want to protect.
- Infranet agents—An Infranet agent is a client-side component that runs directly on network endpoints (such as users' computers). The agent checks that the endpoint complies to the security criteria specified in Host Checker policies and relays that compliance information to the Infranet Enforcer. The Infranet Enforcer then allows or denies the endpoint access based on the compliance results.

An SRX Series device can act as an Infranet Enforcer in a UAC network. Specifically, it acts as a Layer 3 enforcement point, controlling access by using IP-based policies pushed down from the IC Series appliance.

When deployed in a UAC network, an SRX Series device is called a Junos OS Enforcer. See Figure 12 on page 122.

**Figure 12: Integrating a Junos OS Security Device into a Unified Access Control Network**



**NOTE:** You can use the Junos OS Enforcer with the IC Series appliance and Secure Access devices in an IF-MAP Federation network. In a federated network, multiple IC Series appliances and Secure Access devices that are not directly connected to the Junos OS Enforcer can access resources protected by the security device. There are no configuration tasks for IF-MAP Federation on the Junos OS Enforcer. You configure policies on IC Series appliances that can dynamically create authentication table entries on the Junos OS Enforcer.

## Enabling UAC in a Junos OS Environment (CLI Procedure)

Junos OS security policies enforce rules for transit traffic, defining what traffic can pass through the Juniper Networks device. The policies control traffic that enters from one zone (from-zone) and exits another (to-zone). To enable an SRX Series device as a Junos OS Enforcer in a UAC deployment, you must:

- Identify the source and destination zones through which UAC traffic will travel. It also needs the list of interfaces, including which zones they are in. The IC Series UAC Appliance uses the destination zone to match its own IPsec routing policies configured on IC Series appliance.
- Identify Junos OS security policies that encompass those zones, and enable UAC for those policies.

Before you begin:

1. Set up the interfaces through which UAC traffic should enter the SRX Series device.
2. Group interfaces with identical security requirements into zones. See *Example: Creating Security Zones*.
3. Create security policies to control the traffic that passes through the security zones. See *Example: Configuring a Security Policy to Permit or Deny All Traffic*.

To configure UAC through a Junos OS security policy, enter the following configuration statement:

```
user@host# set security policies from-zone zone-name to-zone zone-name policy match then permit
    application-services uac-policy
```

# Set Up Communication between Junos OS Enforcer and IC Series UAC Appliance

IN THIS SECTION


- Understanding Communications Between the Junos OS Enforcer and the IC Series UAC Appliance | 124
- Understanding Communications Between Junos OS Enforcer and a Cluster of IC Series UAC Appliances | 125
- Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure) | 126
- Understanding Junos OS Enforcer Implementations Using IPsec | 128
- Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI) | 129


In a Unified Access Control (UAC) network, an SRX Series device is called as Junos OS Enforcer when it is deployed in the UAC environment. The SRX Series device verifies the certificate which IC Series appliance submits. The SRX Series device and IC Series appliance perform mutual authentication. After authentication, the IC Series appliance sends user and resource access policy information to the SRX Series device to act as the Junos OS Enforcer.

## Understanding Communications Between the Junos OS Enforcer and the IC Series UAC Appliance

When you configure an SRX Series device to connect to an IC Series UAC Appliance, the SRX Series device and the IC Series appliance establish secure communications as follows:

1. If more than one IC Series device are configured as Infranet Controllers on the SRX Series device, a round-robin algorithm determines which of the configured IC Series devices is the active Infranet Controller. The others are failover devices. If the active Infranet Controller becomes inoperative, the algorithm is reapplied to the remaining IC Series devices that are configured to establish the new active Infranet Controller.
2. The active IC Series appliance presents its server certificate to the SRX Series device. If configured to do so, the SRX Series device verifies the certificate. (Server certificate verification is not required; however, as an extra security measure you can verify the certificate to implement an additional layer of trust.)

3. The SRX Series device and the IC Series appliance perform mutual authentication using the proprietary challenge-response authentication. For security reasons, the password is not included in the message sent to the IC Series appliance.
4. After successfully authenticating with the SRX Series device, the IC Series appliance sends its user authentication and resource access policy information. The SRX Series device uses this information to act as the Junos OS Enforcer in the UAC network.
5. Thereafter, the IC Series appliance and the Junos OS Enforcer can communicate freely with one another over the SSL connection. The communications are controlled by a proprietary protocol called *Junos UAC Enforcer Protocol (JUEP)*.

## Understanding Communications Between Junos OS Enforcer and a Cluster of IC Series UAC Appliances

You can configure a Junos OS Enforcer to work with more than one IC Series UAC Appliance in a high availability configuration known as an IC Series appliance cluster. The Junos OS Enforcer communicates with only one IC Series appliance at a time; the other IC Series appliances are used for failover. If the Junos OS Enforcer cannot connect to the first IC Series appliance you added to a cluster, it tries to connect to the failed IC Series appliance again. Then it fails over to the other IC Series appliances in the cluster. It continues trying to connect to IC Series appliances in the cluster until a connection occurs.

When the Junos OS Enforcer cannot establish a connection to an Infranet Enforcer, it preserves all its existing authentication table entries and Unified Access Control (UAC) policies and takes the timeout action that you specify. Timeout actions include:

- **close**—Close existing sessions and block any further traffic. This is the default option.
- **no-change**—Preserve existing sessions and require authentication for new sessions.
- **open**—Preserve existing sessions and allow new sessions access.

Once the Junos OS Enforcer can reestablish a connection to an IC Series appliance, the IC Series appliance compares the authentication table entries and UAC policies stored on the Junos OS Enforcer with the authentication table entries and policies stored on the IC Series appliance and reconciles the two as required.

> **NOTE:** The IC Series appliances configured on a Junos OS Enforcer should all be members of the same IC Series appliance cluster.

## Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure)

To configure an SRX Series device to act as a Junos OS Enforcer in a UAC deployment, and therefore to enforce IC Series UAC Appliance policies, you must specify an IC Series appliance to which the SRX Series device should connect.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See "Enabling UAC in a Junos OS Environment (CLI Procedure)" on page 123.
2. (Optional) Create a profile for the certificate authority (CA) that signed the IC Series appliance's server certificate, and import the CA certificate onto the SRX Series device. See *Example: Loading CA and Local Certificates Manually*.
3. Configure user authentication and authorization by setting up user roles, authentication and authorization servers, and authentication realms on the IC Series appliance.
4. Configure resource access policies on the IC Series appliance to specify which endpoints are allowed or denied access to protected resources.

To configure an SRX Series device to act as a Junos OS Enforcer:

1. Specify the IC Series appliance(s) to which the SRX Series device should connect.
   - To specify the IC Series appliance hostname:

     ```
     user@host# set services unified-access-control infranet-controller hostname
     ```

   - To specify the IC Series appliance IP address:

     ```
     user@host# set services unified-access-control infranet-controller hostname address ip-address
     ```

**NOTE:** When configuring access to multiple IC Series appliances, you must define each separately. For example:

```
user@host# set services unified-access-control infranet-controller IC1
user@host# set services unified-access-control infranet-controller IC2
user@host# set services unified-access-control infranet-controller IC3
```

```
user@host# set services unified-access-control infranet-controller IC1 address 10.10.10.1
user@host# set services unified-access-control infranet-controller IC2 address 10.10.10.2
user@host# set services unified-access-control infranet-controller IC3 address 10.10.10.3
```

Make sure that all of the IC Series appliances are members of the same cluster.

**NOTE:** By default, the IC Series appliance should select port 11123.

2. Specify the Junos OS interface to which the IC Series appliance should connect:

```
user@host# set services unified-access-control infranet-controller hostname interface interface-name
```

3. Specify the password that the SRX Series device should use to initiate secure communications with the IC Series appliance:

**NOTE:** Any change in the Unified Access Control's (UAC) contact interval and timeout values in the SRX Series device will be effective only after the next reconnection of the SRX Series device with the IC Series appliance.

```
user@host# set services unified-access-control infranet-controller hostname password password
```

4. (Optional) Specify information about the IC Series appliance's server certificate that the SRX Series device needs to verify the certificate.

   - To specify the server certificate subject that the SRX Series device checks:

```
user@host# set services unified-access-control infranet-controller hostname server-certificate-subject
    certificate-name
```

- To specify the CA profile associated with the certificate:

```
user@host# set services unified-access-control infranet-controller hostname ca-profile ca-profile
```

**NOTE:** An IC Series appliance server certificate can be issued by an intermediate CA. There are two types of CAs—root CAs and intermediate CAs. An intermediate CA is secondary to a root CA and issues certificates to other CAs in the public key infrastructure (PKI) hierarchy. Therefore, if a certificate is issued by an intermediate CA, you need to specify the complete list of CA profiles in the certification chain.

## Understanding Junos OS Enforcer Implementations Using IPsec

To configure an SRX Series device to act as a Junos OS Enforcer using IPsec, you must:

- Include the identity configured under the security IKE gateway. The identity is a string such as "gateway1.mycompany.com", where gateway1.mycompany.com distinguishes between IKE gateways. (The identities specify which tunnel traffic is intended.)
- Include the preshared seed. This generates the preshared key from the full identity of the remote user for Phase 1 credentials.
- Include the RADIUS shared secret. This allows the IC Series UAC Appliance to accept RADIUS packets for extended authentication (XAuth) from the Junos OS Infranet Enforcer.

When configuring IPsec between the IC Series appliance, the Odyssey Access Client, and the SRX Series device, you should note that the following are IKE (or Phase 1) proposal methods or protocol configurations that are supported from the IC Series appliance to the Odyssey Access Client:

- IKE proposal: **authentication-method pre-shared-keys** (you must specify **pre-shared-keys**)
- IKE policy:
  - **mode aggressive** (you must use aggressive mode)
  - **pre-shared-key ascii-text *key*** (only ASCII text preshared-keys are supported)
- IKE gateway: dynamic
  - **hostname *identity*** (you must specify a unique identity among gateways)
  - **ike-user-type group-ike-id** (you must specify **group-ike-id**)
  - **xauth access-profile *profile*** (you must specify **xauth**)

The following are IPsec (or Phase 2) proposal methods or protocol configurations that are supported from the IC Series appliance to the Odyssey Access Client.

- IPsec proposal: **protocol esp** (you must specify **esp**)
- IPsec VPN: **establish-tunnels immediately** (you must specify **establish-tunnels immediately**)

**NOTE:**

- Only one IPsec VPN tunnel is supported per from-zone to to-zone security policy. This is a limitation on the IC Series appliance.
- Junos OS security policies enable you to define multiple policies differentiated by different source addresses, destination addresses, or both. The IC Series appliance, however, cannot differentiate such configurations. If you enable multiple policies in this manner, the IC Series appliance could potentially identify the incorrect IKE gateway.

## Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI)

To configure an SRX Series device to act as a Junos OS Enforcer using IPsec:

1. Set system and syslog information using the following configuration statements:

```
system {
host-name test_host;
    domain-name test.mycompany.com;
    host-name test_host;
    root-authentication {
    encrypted-password "$ABC123";
    }
    services {
    ftp;
        ssh;
        telnet;
        web-management {
        http {
            interface ge-0/0/0.0;
        }
    }
}
syslog {
user * {
```

```
    any emergency;
      }
      file messages {
      any critical;
      authorization info;
      }
      file interactive-commands {
      interactive-commands error;
    }
    }
    max-configurations-on-flash 5;
    max-configuration-rollbacks 5;
    license {
    autoupdate {
      url https://ae1.mycompany.com/junos/key_retrieval;
    }
    }
    ntp {
    boot-server 1.2.3.4;
    server 1.2.3.4;
}
}
```

**NOTE:** On SRX Series devices, the factory default for the maximum number of backup configurations allowed is five. Therefore, you can have one active configuration and a maximum of five rollback configurations. Increasing this backup configuration number will result in increased memory usage on disk and increased commit time.

To modify the factory defaults, use the following commands:

```
root@host#  set system max-configurations-on-flash number
root@host#  set system max-configuration-rollbacks number
```

where **max-configurations-on-flash** indicates backup configurations to be stored in the configuration partition and **max-configuration-rollbacks** indicates the maximum number of backup configurations.

2. Configure the interfaces using the following configuration statements:

```
interfaces {
ge-0/0/0 {
```

```
    unit 0 {
      family inet {
        address 10.64.75.135/16;
      }
    }
}
ge-0/0/1 {
    unit 0 {
      family inet {
        address 10.100.54.1/16;
      }
    }
}
ge-0/0/2 {
    unit 0 {
      family inet {
        10.101.54.1/16;
      }
    }
}
```

3. Configure routing options using the following configuration statements:

```
routing-options {
static {
    route 0.0.0.0/0 next-hop 10.64.0.1;
      route 10.11.0.0/16 next-hop 10.64.0.1;
      route 172.0.0.0/8 next-hop 10.64.0.1;
    route 10.64.0.0/16 next-hop 10.64.0.1;
}
}
```

4. Configure security options using the following configuration statements:

```
security {
ike {
    traceoptions {
      file ike;
      flag all;
    }
    proposal prop1 {
    authentication-method pre-shared-keys;
```

```
        dh-group group2;
        authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
    }
    policy pol1 {
    mode aggressive;
        proposals prop1;
    pre-shared-key ascii-text "$ABC123";
    }
    gateway gateway1 {
    ike-policy pol1;
        dynamic {
        hostname gateway1.mycompany.com;
            connections-limit 1000;
        ike-user-type group-ike-id;
        }
        external-interface ge-0/0/0;
    xauth access-profile infranet;
    }
    gateway gateway2 {
    ike-policy pol1;
        dynamic {
        hostname gateway2.mycompany.com;
            connections-limit 1000;
        ike-user-type group-ike-id;
        }
        external-interface ge-0/0/0;
    xauth access-profile infranet;
}
}
```

5. Configure IPsec parameters using the following configuration statements:

```
ipsec {
proposal prop1 {
    protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm 3des-cbc;
    lifetime-seconds 86400;
    }
    policy pol1 {
    proposals prop1;
    }
```

```
    vpn vpn1 {
    ike {
        gateway gateway1;
        ipsec-policy pol1;
    }
    }
    vpn vpn2 {
    ike {
        gateway gateway2;
        ipsec-policy pol1;
    }
}
}
```

6. Configure screen options using the following configuration statements:

```
screen {
ids-option untrust-screen {
    icmp {
        ping-death;
        }
        ip {
        source-route-option;
        tear-drop;
        }
        tcp {
        syn-flood {
            alarm-threshold 1024;
                attack-threshold 200;
                source-threshold 1024;
                destination-threshold 2048;
                queue-size 2000;
            timeout 20;
            }
        land;
    }
}
}
```

7. Configure zones using the following configuration statements:

```
zones {
```

```
security-zone trust {
    tcp-rst;
        host-inbound-traffic {
        system-services {
            all;
            }
            protocols {
            all;
        }
        }
        interfaces {
        ge-0/0/0.0;
    }
    }
    security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
            }
            protocols {
            all;
        }
        }
        interfaces {
        ge-0/0/1.0;
    }
    }
    security-zone zone101 {
    host-inbound-traffic {
        system-services {
            all;
            }
            protocols {
            all;
        }
        }
        interfaces {
        ge-0/0/2.0;
    }
}
}
```

8. Configure policies for UAC using the following configuration statements:

```
policies {
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
                destination-address any;
            application any;
            }
            then {
            permit;
        }
    }
    }
    from-zone trust to-zone untrust {
    policy default-permit {
        match {
            source-address any;
                destination-address any;
            application any;
            }
            then {
            permit;
        }
        }
        policy default-deny {
        match {
            source-address any;
                destination-address any;
            application any;
            }
            then {
            permit;
        }
        }
        policy pol1 {
        match {
            source-address any;
                destination-address any;
            application any;
            }
            then {
            permit {
                tunnel {
                    ipsec-vpn vpn1;
```

```
            }
            application-services {
            uac-policy;
          }
          }
          log {
          session-init;
          session-close;
        }
      }
    }
    }
    from-zone untrust to-zone trust {
    policy pol1 {
      match {
        source-address any;
          destination-address any;
        application any;
        }
        then {
        permit;
          log {
          session-init;
          session-close;
        }
      }
    }
    }
    from-zone trust to-zone zone101 {
    policy pol1 {
      match {
        source-address any;
          destination-address any;
        application any;
        }
        then {
        permit {
          tunnel {
            ipsec-vpn vpn2;
            }
            application-services {
            uac-policy;
          }
          }
```

```
            log {
            session-init;
            session-close;
          }
        }
        }
        policy test {
        match {
          source-address any;
            destination-address any;
          application any;
          }
          then {
          permit;
        }
      }
      }
      default-policy {
      deny-all;
}
}
}
```

9. Configure RADIUS server authentication access using the following configuration statements:

```
access {
profile infranet {
    authentication-order radius;
      radius-server {
      10.64.160.120 secret "$ABC123";
    }
}
}
```

10. Configure services for UAC using the following configuration statements:

```
services {
unified-access-control {
    infranet-controller IC27 {
      address 3.23.1.2;
        interface ge-0/0/0.0;
      password "$ABC123";
```

```
        }
        infranet-controller prabaIC {
        address 10.64.160.120;
            interface ge-0/0/0.0;
        password "$ABC123";
        }
        certificate-verification optional;
        traceoptions {
        flag all;
    }
}
}
```

# Enforce Policies and Configure Endpoint Security with Junos OS Enforcer

**IN THIS SECTION**


- Understanding Junos OS Enforcer Policy Enforcement | **139**
- Configuring Junos OS Enforcer Failover Options (CLI Procedure) | **140**
- Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) | **141**
- Verifying Junos OS Enforcer Policy Enforcement | **141**
- Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer | **142**
- Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer | **143**


In a Unified Access Control (UAC) environment, after an SRX Series device becomes Junos OS Enforcer, the SRX Series device allows or denies traffic based on Junos OS security policy. Infranet agent runs on the endpoints to secure traffic by checking UAC Host Checker policies. Based on the Host Checker compliance results, Junos OS Enforcer allows or denies the endpoint access.

## Understanding Junos OS Enforcer Policy Enforcement

Once the SRX Series device has successfully established itself as the Junos OS Enforcer, it secures traffic as follows:

1. First, the Junos OS Enforcer uses the appropriate Junos OS security policy to process the traffic. A *security policy* uses criteria such as the traffic's source IP address or the time of day that the traffic was received to determine whether or not the traffic should be allowed to pass.
2. Once it determines that the traffic may pass based on the Junos OS security policy, the Junos OS Enforcer maps the traffic flow to an authentication table entry. The Junos OS Enforcer uses the source IP address of the first packet in the flow to create the mapping.

   An *authentication table entry* contains the source IP address and user role(s) of a user who has already successfully established a UAC session. A *user role* identifies a group of users based on criteria such as type (for instance, "Engineering" or "Marketing") or status (for instance, "Antivirus Running"). The Junos OS Enforcer determines whether to allow or deny the traffic to pass based on the authentication results stored in the appropriate authentication table entry.

   The IC Series UAC Appliance pushes authentication table entries to the Junos OS Enforcer when the devices first connect to one another and, as necessary, throughout the session. For example, the IC Series appliance might push updated authentication table entries to the Junos OS Enforcer when the user's computer becomes noncompliant with endpoint security policies, when you change the configuration of a user's role, or when you disable all user accounts on the IC Series appliance in response to a security problem such as a virus on the network.

   If the Junos OS Enforcer drops a packet because of a missing authentication table entry, the device sends a message to the IC Series appliance, which in turn may provision a new authentication table entry and send it to the Junos OS Enforcer. This process is called dynamic authentication table provisioning.
3. Once it determines that the traffic may pass based on the authentication table entries, the Junos OS Enforcer maps the flow to a resource. The Junos OS Enforcer uses the destination IP address specified in the flow to create the mapping. Then the device uses that resource as well as the user role specified in the authentication table entry to map the flow to a resource access policy.

   A *resource access policy* specifies a particular resource to which you want to control access based on user role. For instance, you might create a resource access policy that allows only users who are members of the Engineering and Antivirus Running user roles access to the Engineering-Only server. Or you might create a resource access policy that allows members of the No Antivirus Running user role access to the Remediation server on which antivirus software is available for download.

   The IC Series appliance pushes resource access policies to the Junos OS Enforcer when the devices first connect to one another and when you modify your resource access policy configurations on the IC Series appliance.

If the Junos OS Enforcer drops the packet because of a “deny” policy, the Junos OS Enforcer sends a message to the IC Series appliance, which in turn sends a message to the endpoint’s Odyssey Access Client (if available). (The IC Series appliance does not send “deny” messages to the agentless client.)

4. Once it determines that the traffic may pass based on the resource access policies, the Junos OS Enforcer processes the traffic using the remaining application services defined in the Junos OS policy. The Junos OS Enforcer runs the remaining services in the following order: Intrusion Detection and Prevention (IDP), URL filtering, and Application Layer Gateways (ALGs).

## Configuring Junos OS Enforcer Failover Options (CLI Procedure)

To configure IC Series UAC Appliance failover processing, you must configure the Junos OS Enforcer to connect to a cluster of IC Series appliances. The Junos OS Enforcer communicates with one of these IC Series appliances at a time and uses the others for failover processing.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies.
2. Configure the SRX Series device as a Junos OS Enforcer. During the configuration, define a cluster of IC Series appliances to which the Junos OS Enforcer should connect. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 123.

To configure failover processing:

1. Specify how often (in seconds) the Junos OS Enforcer should expect a heartbeat signal from the IC Series appliance indicating an active connection:

```
user@host# set services unified-access-control interval seconds
```

2. Specify the interval (in seconds) at which the Junos OS Enforcer should consider the current connection timed out:

> **NOTE:** Any change in the Unified Access Control’s (UAC) contact interval and timeout values in the SRX Series device will be effective only after the next reconnection of the SRX Series device with the IC Series appliance.

```
user@host# set services unified-access-control timeout seconds
```

3. Specify how the Junos OS Enforcer should handle all current and subsequent traffic sessions when its connection to an IC Series appliance cluster times out:

```
user@host# set services unified-access-control timeout-action (close | no-change | open)
```

## Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure)

When configured in test-only mode, the SRX Series device enables all UAC traffic to go through regardless of the UAC policy settings. The device logs the UAC policy's access decisions without enforcing them so you can test the implementation without impeding traffic.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See "Enabling UAC in a Junos OS Environment (CLI Procedure)" on page 123
2. Configure the SRX Series devices as a Junos OS Enforcer. See "Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure)" on page 126.
3. If you are connecting to a cluster of IC Series UAC Appliances, enable failover options. See "Configuring Junos OS Enforcer Failover Options (CLI Procedure)" on page 140.

To activate or deactivate test-only mode, enter the following configuration statement:

```
user@host#  set services unified-access-control test-only-mode (true | false)
```

## Verifying Junos OS Enforcer Policy Enforcement

**IN THIS SECTION**


- Displaying IC Series UAC Appliance Authentication Table Entries from the Junos OS Enforcer | **142**
- Displaying IC Series UAC Appliance Resource Access Policies from the Junos OS Enforcer | **142**

### Displaying IC Series UAC Appliance Authentication Table Entries from the Junos OS Enforcer

**Purpose**

Display a summary of the authentication table entries configured from the IC Series UAC Appliance.

**Action**

Enter the **show services unified-access-control authentication-table** CLI command.

### Displaying IC Series UAC Appliance Resource Access Policies from the Junos OS Enforcer

**Purpose**

Display a summary of UAC resource access policies configured from the IC Series UAC Appliance.

**Action**

Enter the **show services unified-access-control policies** CLI command.

## Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer

An Infranet agent helps you secure traffic on your network starting with the endpoints that initiate communications as follows:

1. The Infranet agent, which runs directly on the endpoint, checks that the endpoint is compliant with your Unified Access Control (UAC) Host Checker policies. You can use a wide variety of criteria within a UAC Host Checker policy to determine compliance. For example, you can configure the Host Checker policy to confirm that the endpoint is running antivirus software or a firewall or that the endpoint is not running specific types of malware or processes.
2. The Infranet agent transmits the compliance information to the Junos OS Enforcer.
3. The Junos OS Enforcer allows or denies the endpoint access to the resources on your network based on the Host Checker compliance results.

Because the Infranet agent runs directly on the endpoint, you can use the Infranet agent to check the endpoint for security compliance at any time. For instance, when a user tries to sign into the IC Series UAC Appliance, you can require the Infranet agent to send compliance results immediately—the user will not even see the sign-in page until the Infranet agent returns positive compliance results to the IC Series appliance. You can also configure the Infranet agent to check for compliance after the user signs in or periodically during the user session.

If the endpoints running the Infranet agent have appropriate access, they will automatically send their compliance results to the IC Series appliance, and the IC Series appliance will update the authentication

table entries accordingly and push them to the Junos OS Enforcer. The Junos OS Enforcer supports connections with the Odyssey Access Client and “agentless” Infranet agents.

### Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer

To integrate the Infranet agent into a Junos OS-UAC deployment, no special configuration is required on the Junos OS Enforcer. You simply need to create security policies enabling access to the appropriate endpoints as you would for any other Junos OS-UAC deployment.

RELATED DOCUMENTATION

*Security Policies User Guide for Security Devices*

# Configure Captive Portal on Junos OS Enforcer

**IN THIS SECTION**


- Understanding the Captive Portal on the Junos OS Enforcer | **144**
- Understanding Captive Portal Configuration on the Junos OS Enforcer | **146**
- Understanding the Captive Portal Redirect URL Options | **146**
- Example: Creating a Captive Portal Policy on the Junos OS Enforcer | **148**


By enabling the captive portal on the Junos OS enforcer, you can redirect a user to authenticate through IC Series UAC Appliance without their knowledge. After successful authentication, the IC Series appliance redirects the user to the protected resource that they want to access.

## Understanding the Captive Portal on the Junos OS Enforcer

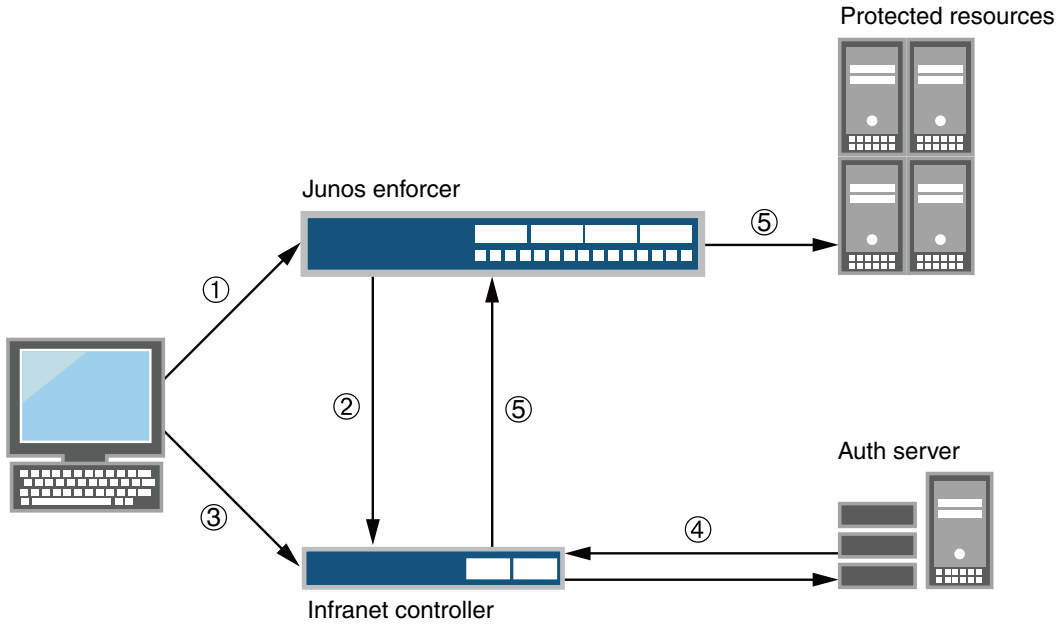
In a Unified Access Control (UAC) deployment, users might not be aware that they must first sign in to the IC Series UAC Appliance for authentication and endpoint security checking before they are allowed to access a protected resource behind the Junos OS Enforcers. To help users sign in to the IC Series appliance, you can configure the captive portal feature. The captive portal feature allows you to configure a policy in the Junos OS Enforcer that automatically redirects HTTP traffic destined for protected resources to the IC Series appliance or to a URL configured in the Junos OS Enforcer.

You can configure a captive portal for deployments that use either source IP enforcement or IPsec enforcement, or a combination of both enforcement methods.

Figure 13 on page 145 shows the captive portal feature enabled on a Junos OS Enforcer. Users accessing protected resources are automatically redirected to the IC Series appliance:

1. Users point to a protected resource using the browser.
2. The Junos OS Enforcer determines that the user is not authenticated and redirects the request to the IC Series appliance or another server.
3. Users enter their Infranet username and password to log in.
4. The IC Series appliance passes the user credentials to an authentication server.
5. After authentication, the IC Series appliance redirects the users to the protected resource they wanted to access.

**Figure 13: Enabling the Captive Portal Feature on a Junos OS Enforcer**




By default, the Junos OS Enforcer encodes and forwards to the IC Series appliance the protected resource URL that the user entered. The IC Series appliance uses the protected resource URL to help users navigate to the protected resource. The manner in which the IC Series appliance uses the protected resource URL depends on whether or not the user's endpoint is running the Odyssey Access Client or Junos Pulse. If the user's endpoint is not running the Odyssey Access Client or Junos Pulse (that is, it is in an agentless or Java agent configuration), the IC Series appliance automatically opens a new browser window and uses HTTP to access the protected resource after the user signs in. If the endpoint is using the Odyssey Access Client, the IC Series appliance inserts a hypertext link in the webpage that automatically opens after the user signs in. The user must then click that hypertext link to access the protected resource by means of HTTP in the same browser window.

The Junos OS Enforcer supports the captive portal feature only for HTTP traffic. If you attempt to access a protected resource by using HTTPS or a non-browser application (such as an e-mail application), the Junos OS Enforcer does not redirect the traffic. When using HTTPS or a non-browser application, you must manually sign in to the IC Series appliance first before attempting to access protected resources.

## Understanding Captive Portal Configuration on the Junos OS Enforcer

To configure the captive portal feature, you create a security policy on the Junos OS Enforcer and then specify a redirection option for the captive portal security policy. You can choose to redirect traffic to an external server or to the IC Series UAC Appliance. You can also choose to redirect all traffic or unauthenticated traffic only.

- Redirecting traffic to an external webserver—You can configure the Junos OS Enforcer to redirect HTTP traffic to an external webserver instead of the IC Series appliance. For example, you can redirect HTTP traffic to a webpage that explains to users the requirement to sign in to the IC Series appliance before they can access the protected resource. You could also include a link to the IC Series appliance on that webpage to help users sign in.
- Redirecting unauthenticated traffic—Select this option if your deployment uses source IP only or a combination of source IP and IPsec. The Junos OS Enforcer redirects clear-text traffic from unauthenticated users to the currently connected IC Series appliance or to an IP address or domain name that you specify in a redirect URL. After a user signs in to the IC Series appliance and the user's endpoint system meets the requirements of the IC Series appliance security policies, the Junos OS Enforcer allows the user's clear-text traffic to pass through in source IP deployments. For IPsec deployments, the Odyssey Access Client creates a VPN tunnel between the user and the Junos OS Enforcer. The Junos OS Enforcer then applies the VPN policy, allowing the encrypted traffic to pass through.
- Redirecting all traffic—Specify this option if you want to redirect all traffic to the URL that you specify in a redirect URL.
- Redirecting traffic with multiple IC Series appliances—You can configure multiple IC Series appliances on your Junos OS Enforcer, but it is connected to only one IC Series appliance at any given time. If the connection to the IC Series appliance fails, the Junos OS Enforcer tries to connect to next configured IC Series appliance. As a result, you cannot be sure which IC Series appliance is connected to the Junos OS Enforcer at any given time. To ensure that the Junos OS Enforcer redirects traffic to the connected IC Series appliance, configure the default redirect URL or the **%ic-ip%** option in the URL.

## Understanding the Captive Portal Redirect URL Options

By default, after you configure a captive portal policy, the Junos OS Enforcer redirects HTTP traffic to the currently connected IC Series UAC Appliance by using HTTPS. To perform the redirection, the Junos OS Enforcer uses the IP address or domain name that you specified when you configured the IC Series appliance instance on the Junos OS Enforcer. The format of the URL that the Junos OS Enforcer uses for default redirection is:

```
https://%ic-ip%/?target = %dest-url%  &enforcer = %enforcer-id%  &policy = %policy-id%
  &dest-ip = %dest-ip%
```

If you configured your Junos OS Enforcer to work with multiple IC Series appliances in a cluster, and the current IC Series appliance becomes disconnected, the Junos OS Enforcer automatically redirects HTTP traffic to the next active IC Series appliance in its configuration list. The Junos OS Enforcer redirects traffic to only one IC Series appliance at a time.

Otherwise, the browser displays a certificate warning to users when they sign in. You do not need to override the default redirection destination except in these situations:

- You are using a VIP for a cluster of IC Series appliances, and the Junos OS Enforcer is configured to connect to the IC Series appliance physical IP addresses.
- You want to redirect traffic to a webserver instead of the IC Series appliance.
- If, because of split DNS or IP routing restrictions at your site, the Junos OS Enforcer uses a different address for the IC Series appliance than endpoints, you must specify the domain name or IP address that endpoints must use to access the IC Series appliance.

> **NOTE:** If a captive portal policy is configured with the IC Series UAC Appliance URL as the target, then use only HTTPS to redirect traffic.

Table 5 on page 147 lists different options that you can configure in the redirect URL string.

**Table 5: Redirect URL String Options**

| URL String | Description |
|---|---|
| **%dest-url%** | Specifies the protected resource which the user is trying to access. |
| **%enforcer-id%** | Specifies the ID assigned to the Junos OS Enforcer by the IC Series appliance. |
| **%policy-id%** | Specifies the encrypted policy ID for the captive portal security policy that redirected the traffic. |
| **%dest-ip%** | Specifies the IP address or hostname of the protected resource which the user is trying to access. |
| **%ic-ip%** | Specifies the IP address or hostname of the IC Series appliance to which the Junos OS Enforcer is currently connected. |

# Example: Creating a Captive Portal Policy on the Junos OS Enforcer

IN THIS SECTION


- Requirements | 148
- Overview | 148
- Configuration | 149
- Verification | 151


This example shows how to create a captive portal policy on the Junos OS Enforcer. In this example, you deploy a Junos OS Enforcer in front of the data center resources you want to protect and configure the captive portal feature on the Junos OS Enforcer. The Junos OS Enforcer then automatically redirects HTTP traffic destined for the protected resource to the IC Series UAC Appliance for authentication.

## Requirements

Before you begin:

- Deploy the IC Series appliance in the network so that users can access the device. Use the internal port on the IC Series appliance to connect users, the Junos OS Enforcer, and authentication servers. See "Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure)" on page 126.
- Set up security zones and interfaces on the Junos OS Enforcer. Make sure that end users are in a different security zone than protected resources. For example, protected resources in the data center are configured in the trusted zone and users in an untrusted zone. See *Example: Creating Security Zones*.
- Add individual users to either an external authentication server or the local authentication server. Set up roles and realms for individual users. You can provision access to protected resources based on your network security needs.

## Overview

In this example, you want to protect the trusted zone from users on the LAN by making sure that only compliant and authenticated users are granted access. New users join your network every month. You want to configure the captive portal feature on your system so that unauthenticated users are redirected to the IC Series appliance automatically without requiring new users to remember to log in to the IC Series appliance.

The configuration instructions in this topic describe how to create a security policy called **my-policy**, specify a match condition for this policy, specify the captive portal policy as a part of the UAC policy, and set criteria for redirecting traffic to the IC Series appliance. In this example, the policy **my-policy**:

- Specifies the match condition to include any traffic from a previously configured zone called **trust** to another previously configured zone called **untrust**.
- Specifies the captive portal policy called **my-captive-portal-policy** as part of the UAC policy.
- Specifies the redirect-traffic criteria as **unauthenticated**.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set security policies from-zone untrust to-zone trust policy my-policy match destination-address any
    source-address any application junos-http
set security policies from-zone untrust to-zone trust policy my-policy then permit application-services uac-policy
    captive-portal my-captive-portal-policy
set services unified-access-control captive-portal my-captive-portal-policy redirect-traffic unauthenticated
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To create a captive portal policy on the Junos OS Enforcer:

1. Specify the match condition for the policy.

   ```
   [edit security policies from-zone untrust to-zone trust policy my-policy]
   user@host# set match destination-address any source-address any application junos-http
   ```

2. Specify the captive portal policy as part of the UAC policy to be applied on the traffic that matches the conditions specified in the security policy.

   ```
   [edit security policies from-zone untrust to-zone trust policy my-policy]
   user@host# set then permit application-services uac-policy captive-portal my-captive-portal-policy
   ```

3. Redirect all unauthenticated traffic to the IC Series appliance.

```
[edit services unified-access-control]
user@host# set captive-portal my-captive-portal-policy redirect-traffic unauthenticated
```

**Results**

Confirm your configuration by entering the **show services** and **show security policies** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
   user@host# show services
   unified-access-control {
   captive-portal my-captive-portal-policy {
      redirect-traffic unauthenticated;
   }
}
```

```
[edit]
   user@host# show security policies
   ...
   from-zone untrust to-zone trust {
   policy my-policy {
      match {
         source-address any;
         destination-address any;
         application junos-http;
      }
      then {
         permit {
            application-services {
               uac-policy {
                  captive-portal my-captive-portal-policy;
               }
            }
         }
      }
   }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

IN THIS SECTION


- Verifying the Captive Portal Policy | 151


To confirm that the configuration is working properly, perform this task:

***Verifying the Captive Portal Policy***

**Purpose**

Verify that the captive portal policy was created.

**Action**

From operational mode, enter the **show security policies detail** command.

# Classify Traffic Based on User Roles

IN THIS SECTION


- Understanding Unified Access Control | 151
- Acquiring User Role Information from an Active Directory Authentication Server | 152
- Obtaining Username and Role Information Through Firewall Authentication | 172


A user is allowed or denied access based on the security policies. User role firewall security policies let you classify traffic based on the roles to which a user is assigned. The user role information can be collected from Junos Pulse server or third-party authentication server.

## Understanding Unified Access Control

In Junos OS Release 12.1 and later, user role firewall security policies let you classify traffic based on the roles to which a user is assigned. Based on match criteria, which includes the user's role, you create policies

to apply services that allow or block access to resources. The user role firewall is similar to the identity-based network access control (NAC) solution available with UAC on the SRX Series device. A user role firewall, however, does not require the Junos Pulse/Odyssey installation, and it supports agentless transparent authentication.

User role information can be collected in several ways: locally on the SRX Series device, from a Junos Pulse Access Control Service device, or by relaying authentication data from a third-party authentication server through a Junos Pulse Access Control Service device to the SRX Series device.

## Acquiring User Role Information from an Active Directory Authentication Server

**IN THIS SECTION**


- Requirements | **153**
- Overview | **153**
- Configuration | **156**


Networks have used the IP address as a way of identifying users and servers. The strategy is based on the assumption that users or groups of users connect to the network from fixed locations and use one device at a time.

Wireless networking and mobile devices require a different strategy. Individuals can connect to the network using multiple devices simultaneously. The way in which devices connect to the network changes rapidly. It is no longer possible to identify a user with a group of statically allocated IP addresses.

In Junos OS Release 12.1 and later, user role firewall security policies let you classify traffic based on the roles to which a user is assigned. Based on match criteria, which includes the user's role, you create policies to apply services that allow or block access to resources. The user role firewall is similar to the identity-based network access control (NAC) solution available with UAC on the SRX Series device. A user role firewall, however, does not require the Junos Pulse/Odyssey installation, and it supports agentless transparent authentication.

User role information can be collected in several ways: locally on the SRX Series device, from a Junos Pulse Access Control Service device, or by relaying authentication data from a third-party authentication server through a Junos Pulse Access Control Service device to the SRX Series device.

Incorporating a third-party authentication server into a user role firewall configuration can also provide single sign-on (SSO) support. This allows a browser-based user to authenticate once and have that authentication communicated to other trusted servers in the domain as needed.

### Requirements

This solution uses the following hardware and software components:

- One MAG Series Junos Pulse Gateway device with software release 4.2 or later
- The MAGx600-UAC-SRX license installed on the MAG Series device
- One SRX Series device with Junos OS Release 12.1 or later
- One Microsoft Active Directory server using version 2008

> **NOTE:** Microsoft Windows 2003 is also compatible with this functionality, but terminology, pathways, and settings might differ from what is presented in this document.

Before you begin:

- Ensure that the MAG Series device is configured as an Access Control Service and is accessible to the network. See the *MAG Series Junos Pulse Gateway Hardware Guide* for configuration details.
- Ensure that the MAGx600-UAC-SRX license is installed on the MAG Series device.
- Ensure that the SRX Series device is configured and initialized with Junos OS version 12.1 or later.
- Ensure that the Active Directory authentication server is configured for standard Junos Pulse Access Control Service authentication. See your third-party documentation.
- Ensure that the administrator has the appropriate capabilities for configuring the roles, users, and device interactions.

### Overview

In this solution an SRX Series device obtains user role information dynamically from a Microsoft Active Directory authentication server. Authentication verification and user role information from the Active Directory server is relayed by the Access Control Service on the MAG Series device to the SRX Series device.

Users within the same domain are connected to a LAN segment. They are associated with user role groups, such as developer or manager, depending on their work in the organization. When a user authenticates to the AD authentication server, the user should be able to access protected resources without having to authenticate a second time.

The SRX Series device is configured as an enforcer for the MAG Series device. It receives user role information from the MAG Series device and applies user role firewall policies accordingly to incoming and outgoing traffic.

When the SRX Series device has no user role information for a user, the user's browser is redirected to the MAG Series device. Transparently to the user, the MAG Series device requests verification from the browser. The browser retrieves a token from the Active Directory server confirming authentication and passes it to the MAG Series device. With the information provided by the token, the MAG Series device retrieves user role information for the user from the Active Directory server and creates an authentication table entry consisting of the current IP address and the user role data. The MAG Series device pushes the updated table to the SRX Series device and redirects the browser back to the SRX to request access again. This time, the table does contain user role information which is then retrieved and used as part of the match criteria for applying user role firewall services.

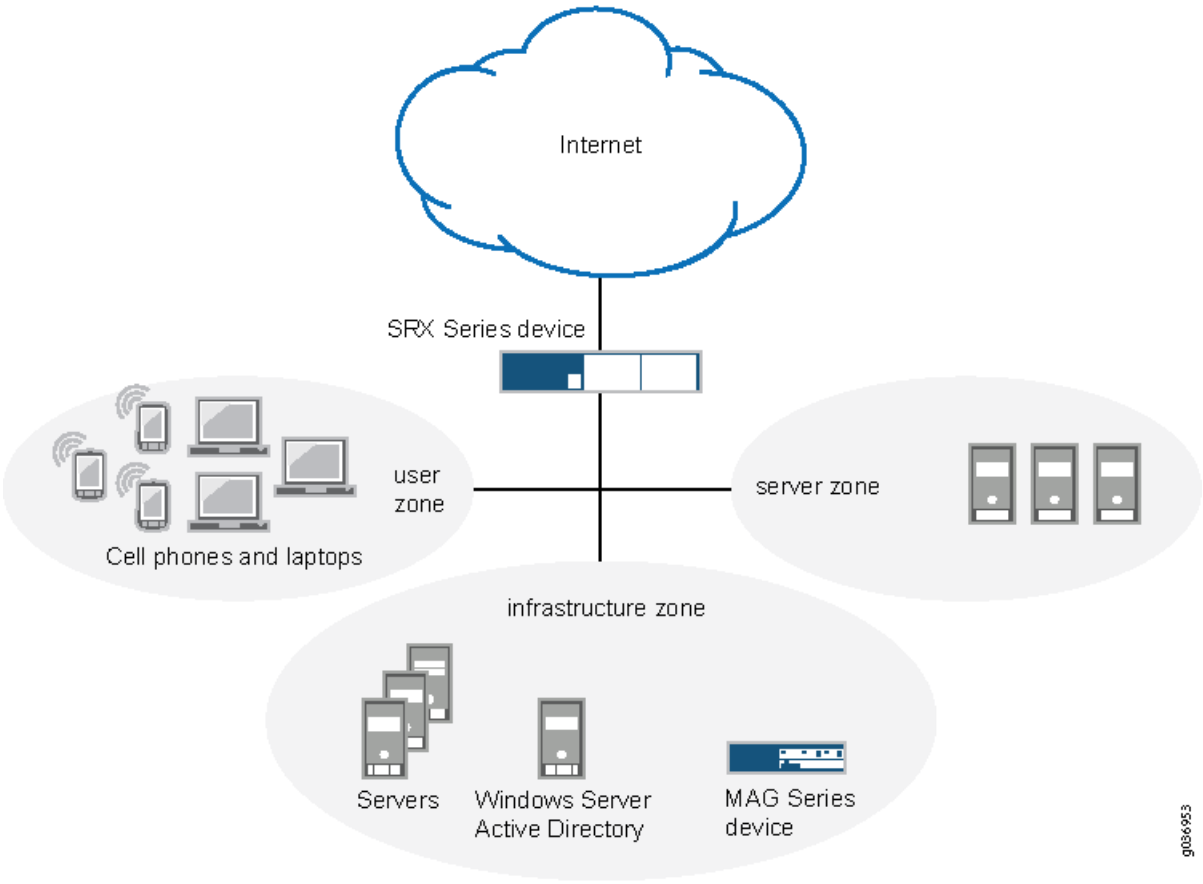
The user is not aware of the process unless the Active Directory (AD) server has no current authentication for the user. When that is the case, the server prompts the user for name and password. Once authentication occurs, the server returns a token to the browser.

The procedure documented here initially configures the MAG Series device as the authenticator. The configuration is later modified to retrieve authentication information from the AD server. This solution uses SPNEGO negotiation and Kerberos authentication to secure communications among the SRX Series device, the MAG Series device, the browser, and the authentication server.

***Topology***

Figure 14 on page 155 shows the topology for this deployment in which the MAG Series device is used initially as the authenticationn source. Later, the AD server is used transparently unless the user is not authenticated, in which case he is prompted for a user name and password.

**Figure 14: Single Sign-On Support Topology**




A user's request to access another resource is controlled by roles and groups associated with the user. For example, a user belonging to a group of developers named Dev might have access to a particular test server. The same user might also be the manager and belong to the Mgr group that can access certain HR resources. A contractor working for this manager might require access to the test server as well but not to the HR resources. In this case, the user would be added to the Dev group and perhaps a Contractor group, but not the Mgr group.

User role firewall policies defined on the SRX Series device control the groups and user roles that can access various resources. In this configuration, if user role data does not exist for a user requesting access, a policy redirects the user's browser to the MAG Series device to authenticate the user and retrieve any associated user role data.

A token exchange among the Access Control Service, the browser, and the Active Directory server remains transparent to the user while it verifies the user's authentication. The exchange uses SPNEGO negotiation and Kerberos authentication for encrypting and decrypting messages among the devices.

With information obtained from the response token, the MAG Series device retrieves the user's roles and groups directly from the Active Directory server. It then creates an authentication table entry and passes it to the SRX Series device.

## Configuration

Configure the devices for this solution by performing the following tasks.

- Connect the SRX Series device and the MAG Series device in an enforcer configuration.
- Configure the Access Control Service on the MAG Series device for local user authentication and verify that authentication information is transferred between the devices.
- Configure a captive portal policy on the SRX Series device to redirect any unauthenticated user to the Access Control Service and verify that redirection is functioning properly.
- Configure the Microsoft Active Directory authentication server to interact with the Access Control Service and the endpoints.
- Reconfigure the Access Control Service for remote authentication by the Active Directory server and redefine Active Directory groups for the SRX Series device.
- Configure endpoint browsers for the SPNEGO protocol

> **NOTE:** Configuring the Access Control Service using local authentication is not necessary for this solution. However, by configuring local authentication first you can verify the captive portal interaction between the MAG Series device and the SRX Series device.

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

***Connecting the SRX Series Device to the Access Control Service***

**Step-by-Step Procedure**

In an enforcer configuration, the Access Control Service on the MAG Series device and the SRX Series device communicate over a secure channel. When the SRX Series device first connects with the Access Control Service, the devices exchange information to ensure secure communication. Optionally, you can use digital security certificates as an enhanced mechanism for establishing trust.

See the *Unified Access Control Administration Guide* for details about configuring certificate trust between the SRX Series device and the Access Control Service.

To connect the SRX Series device and the Access Control Service on the MAG Series device:

1. Configure the SRX Series device.
   a. Configure the zones and interfaces of the devices.

```
user@host# set security zones security-zone user interfaces ge-0/0/0
user@host# set security zones security-zone infrastructure interfaces ge-0/0/1
user@host# set security zones security-zone untrust interfaces ge-0/0/2
```

b. Configure the IP addresses of the interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.12.12.1/8
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.0.0.22/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 203.0.113.19/24
```

c. Identify the Access Control Service as a new Infranet Controller, and configure the interface for the connection to it.

```
[edit]
user@host# set services unified-access-control infranet-controller mag123 address 10.0.0.22
user@host# set services unified-access-control infranet-controller mag123 interface fxp0.0
```

d. Specify the password for securing interactions between the Access Control Service and the SRX Series device.

```
[edit]
user@host# set services unified-access-control infranet-controller mag123 password pwd
```

**NOTE:** The same password must be configured on both devices.

e. (Optional) Specify the full name of the Access Control Service certificate that the SRX Series device must match during connection.

```
user@host# set services unified-access-control infranet-controller mag123 ca-profile ca-mag123-enforcer
```

f. If you are done configuring the SRX Series device, enter commit from configuration mode.

2. Configure the Access Control Service from the administrator console on the MAG Series device.

a. Navigate to the Infranet Enforcer page, and click **New Enforcer**.

b. Select Junos, enter the password set previously on the SRX Series device (InSub321), and enter the serial number of the SRX Series device.

c. Click **Save Changes**.

**Results**

When both devices are configured, the SRX Series device connects automatically to the Access Control Service.

- From the Access Control Service, select **System>Status>Overview** to view the status of the connection to the SRX Series device. The diode in the display is green if the connection is functioning. To display additional information, click the device name.
- From operational mode on the SRX Series device, confirm your connection by entering the **show services unified-access-control status** command. If the output does not display the intended configuration, repeat the instructions in this section to correct the configuration.

```
user@host> show services unified-access-control status
```

```
Host           Address         Port   Interface     State
       mag123        10.0.0.22        11123  fxp0.0        connected
```

***Configuring the Access Control Service for Local User Authentication***

**Step-by-Step Procedure**

When a user is authenticated, the Access Control Service on the MAG Series device updates its authentication table with the IP address and associated roles of the user, and pushes the updated table to the SRX Series device. If this user data is deleted or modified, the Access Control Service updates the authentication table with the new information and again pushes it to the SRX Series device.

To test the successful transfer and content of the authentication table, this task configures the Access Control Service on the MAG Series device for local authentication. Within this configuration you can test the user role firewall from the SRX Series device without affecting other network operations. A later task modifies this configuration to provide user role retrieval from the remote Active Directory server.

> **NOTE:** It is not a requirement to configure the Access Control Service for local user authentication. It is provided so that you can test each task in the configuration.

To configure the Access Control Service for local authentication:

1. Define roles on the Access Control Service.
   a. From the administrator console of the Access Control Service, select **Users>User Roles>New User Role**.
   b. Enter **dev** as the role name.

In this solution, use the default values for other role settings.

c. Click **Save Changes**.

> **NOTE:** This solution assumes that the MAGx600-UAC-SRX license is installed on the Access Control Service. If the full-feature license is installed, you will need to disable OAC Install and enable Agentless Access.

2. Configure the default authentication server.
   a. Select **Authentication>Auth. Servers**.
   b. Select **System Local**. This establishes the MAG Series device as the default authentication server.
3. Create users.
   a. Select the **Users** tab, and click **New**.
   b. Create **user-a** by entering the following details.
      - Username
      - User's full name
      - Password
      - Password confirmation
   c. Repeat the previous step to create **user-b**.
   d. Click **Save Changes**.
4. Create a realm.
   a. Select **Users>User Realms>New User Realm**.
   b. Enter **REALM6** as the realm name.

c. Select **System Local** in the Authentication box.

d. Click **Save Changes**.

5. From the same page, create role mapping rules.

a. Select the **Role Mapping** tab, and click **New Rule**.

b. Define two rules with the following details.

- Enter username user-a, and assign it to role dev.
- Enter username user-b, and assign it to role dev.

c. Click **Save Changes**.

6. Set up the default sign-in page.

a. Select **Authentication>Signing In>Sign-in Policies**.

b. Click the default **Sign-in policy (*/)**.

c. In the **Sign-in URL** box, enter the IP address of this device.

d. In **Authentication realm, Available realms**, select REALM6.

e. Click **Save Changes**.

**Results**

Verify the results of the configuration. If the output does not display the intended configuration, repeat the instructions in this section to correct the configuration.

1. Verify that local authentication on the Access Control Service is functioning properly.

- Open a browser window from an endpoint in the network.
- Enter the fully qualified domain name for the Access Control Service.

  The default sign-in page should display.
- Sign in as user-a, and provide the defined password.

2. From operational mode on the SRX Series device:

a. Confirm that the authentication table on the SRX Series device was updated with **user-a**.

```
user@host> show services unified-access-control authentication-table
```

```
Id    Source IP       Username     Age      Role identifier
1     203.0.113.102    user-a       0        0000000001.000005.0
Total: 1
```

b. Confirm that the correct role has been associated with the role identifier.

```
user@host> show services unified-access-control roles
```

```
Name                Identifier

dev                 0000000001.000005.0
```

c. List all roles associated with user-a.

```
user@host> show services unified-access-control authentication-table detail
```

```
Identifier: 1
  Source IP: 203.0.113.102
  Username: user-a
  Age: 0
  Role identifier       Role name
    0000000001.000005.0 dev
```

***Configuring Redirection from the SRX Series Device to the Access Control Service***

**Step-by-Step Procedure**

Local authentication, as configured in the previous task, requires users to log on to the Access Control Service directly to gain access to network resources. The SRX Series device can be configured to automatically redirect the browser of an unauthenticated user to the Access Control Service if a user requests access to a protected resource directly. You can define a user role firewall policy to redirect an unauthenticated user to a captive portal on the Access Control Service for sign-in.

> **NOTE:** Other services, such as IDP, UTM, AppFW, and AppQoS, can be configured as well as the UAC captive portal implementation. The solution focuses on captive portal for authentication for user role implementation only.

To configure redirection from the SRX Series device to the Access Control Service:

1. From configuration mode on the SRX Series device, configure the profile for the captive portal acs-device.

```
[edit]
user@host# set services unified-access-control captive-portal acs-device redirect-traffic unauthenticated
```

2. Add either the redirection URL for the Access Control Service or a default URL.

```
[edit]
user@host# set services unified-access-control captive-portal acs-device
redirect-url “https://%ic-url%/?target=%dest-url%&enforcer=%enforcer-id%”
```

   This command specifies the default target and enforcer variables so that the browser is returned to the SRX Series device after authentication.

3. Allow traffic to the Acitve Directory (AD) server, the Access Control Service, and the other infrastructure servers.

```
[edit]
user@host# set security policies from-zone user to-zone infrastructure policy Allow-AD-UAC match
   source-address any
user@host# set security policies from-zone user to-zone infrastructure policy Allow-AD-UAC match
   destination-address any
user@host# set security policies from-zone user to-zone infrastructure policy Allow-AD-UAC application
   any
user@host# set security policies from-zone user to-zone infrastructure policy Allow-AD-UAC then permit
```

4. Configure a security policy that redirects HTTP traffic from zone user to zone untrust if the source-identity is unauthenticated-user.

```
[edit]
user@host# set security policies from-zone user to-zone untrust policy user-role-fw1 match source-address
    any
user@host# set security policies from-zone user to-zone untrust policy user-role-fw1 match
    destination-address any
user@host# set security policies from-zone user to-zone untrust policy user-role-fw1 match application http
user@host# set security policies from-zone user to-zone untrust policy user-role-fw1 match source-identity
    unauthenticated-user
```

5. Configure the action to be taken when traffic matches the criteria for user-role-fw1.

   In this case, traffic meeting the specified criteria is allowed access to the UAC captive portal defined by the acs-device profile.

```
user@host# set security policies from-zone user to-zone untrust policy user-role-fw1 then permit
    application-services uac-policy captive-portal acs-device
```

6. Configure a security policy allowing access to any HTTP traffic from zone user to zone untrust.

```
[edit]
user@host# set security policies from-zone user to-zone untrust policy user-role-fw2 match source-address
    any
user@host# set security policies from-zone user to-zone untrust policy user-role-fw2 match
    destination-address any
user@host# set security policies from-zone user to-zone untrust policy user-role-fw2 match application http
user@host# set security policies from-zone user to-zone untrust policy user-role-fw2 match source-identity
    any
user@host# set security policies from-zone user to-zone untrust policy user-role-fw2 then permit
```

> **NOTE:** It is important to position the redirection policy for unauthenticated users before a policy for “any” user so that UAC authentication is not shadowed by a policy intended for authenticated users.

7. If you are done configuring the policies, commit the changes.

```
[edit]
user@host# commit
```

**Results**

Confirm your configuration with the following procedures. If the output does not display the intended configuration, repeat the instructions in this section to correct the configuration.

1. From configuration mode, confirm your captive portal profile configuration by entering the **show services** command.

```
[edit]
user@host# show services
```

```
...
unified-access-control {
 captive-portal acs-device {
  redirect-traffic unauthenticated;
  redirect-url “https://%ic-url%/?target=%dest-url%&enforcer=%enforcer-id%”
...
```

2. From configuration mode, confirm your policy configuration by entering the **show security policies** command.

```
user@host# show security policies
```

```
...
from-zone user to-zone infrastructure {
 policy Allow-AD-UAC  {
  match {
   source-address any;
   destination-address any;
   application any;
  }
  then {
   permit
  }
 }
}
from-zone user to-zone untrust {
 policy user-role-fw1 {
  match {
   source-address any;
   destination-address any;
   application http;
   source-identity unauthenticated-user
```

```
  }
  then {
   permit {
    application-services {
     uac-policy {
      captive-portal acs-device;
     }
    }
   }
  }
 }
}
from-zone user to-zone untrust {
 policy user-role-fw2 {
  match {
   source-address any;
   destination-address any;
   application http;
   source-identity any
  }
  then {
   permit
  }
 }
}
...
```

3. Verify that the redirection policy is functioning correctly.

   a. Open a browser window from a second endpoint in the network.

   b. Enter a third-party URL, such as www.google.com.

      The default sign-in page from the Access Control Service prompts for a user and password.

   c. Enter the username **user-b** and its password.

      The browser should display the requested URL.

> **NOTE:** If a pop-up blocker is set on the endpoint, it could interfere with this functionality.

d. From operational mode on the SRX Series device, verify that the authentication data and roles from the Access Control Service were pushed to the SRX Series device successfully.

```
user@host> show services unified-access-control authentication-table
```

```
Id    Source IP       Username     Age      Role identifier
1     203.0.113.112    user-a       0        0000000001.000005.0
2     203.0.113.15    user-b       0        0000000001.000005.0
Total: 2
```

***Configuring Active Directory Settings***

**Step-by-Step Procedure**

SPNEGO negotiation and Kerberos authentication are transparent to the user and network administrator, but certain configuration options enable the use of these protocols. This section identifies configuration requirements when using Active Directory as the authentication server. To interact in SPNEGO negotiation, the Access Control Service requires a keytab file created by Active Directory. Refer to your third-party documentation for more information about enabling SPNEGO and Kerberos usage.

This section is not intended to be a tutorial for Active Directory. However, there are specific configuration details required for this solution. See your third-party documentation to set up Active Directory as a domain controller.

To configure the Active Directory authentication server:

1. Add a DNS entry as the UAC service account in the **Forward Lookup Zones**. In this way clients can refer to the MAG Series device by name or by IP address.

   This UAC service account name will be used in the next section when reconfiguring the UAC service on the MAG Series device.

2. Single sign-on authentication requires that the UAC service account password never expires. To modify user settings:

   a. From the Active Directory Users and Computers application in DNS, select **Users>New>User** and select the UAC service account created in step 1.

   b. Select the **Account** tab.

   c. In user settings, click **Password Never Expires**.

3. On the Domain Controller, open a command line, and enter the **ktpass** command to create the SPNEGO keytab file.

   The keytab file created on the Active Directory server contains the full service principal name (SPN) and other encryption information from the server. The keytab file is then uploaded to the Access Control Service on the MAG Series device. This shared information identifies one device to the other whenever encrypted messages and responses are sent.

   Use the following syntax.

   ```
   ktpass -out output-file-name -mapuser uac-service-acccount-name -prin service://fqdn@REALM
   ```

   **ktpass**—Third-party Kerberos utility that maps an SPN to a user, in this case, to the UAC service account. The executable is available for download. Refer to your third-party documentation for the source for this utility.

   **-out *output-file-name***—The name for the SPNEGO keytab file you are creating.

   **-mapuser *uac-service-acccount-name***—The name of the UAC service account created in step 1.

   **-prin *service://fqdn@REALM***—The service principal name. The Kerberos authentication uses the SPN in its communication. It does not use an IP address.

   ***service***—The HTTP service.

   ***fqdn***—The hostname of the Junos Pulse Access Control Service. The *service://FQDN* portion of the name is provided by the Access Control Service when registering with the Active Directory server.

   ***REALM***—The realm of the Active Directory authentication server. It is the same as the domain name. The Kerberos realm name is always in uppercase letters following the recommendation in RFC 1510. This affects interoperability with other Kerberos-based environments.

   The following command creates an SPNEGO keytab file named ic.ktpass.

   ```
   ktpass -out ic.ktpass -mapuser icuser@UCDC.COM -princ HTTP/mag123.ucdc.com@UCDC.COM -pass Doj73096
   ```

   This file is copied to the Access Control Service on the MAG Series device in the next section when SPNEGO is configured for remote authentication.

***Reconfiguring Remote Authentication on the Access Control Service***

**Step-by-Step Procedure**

This section reconfigures the Access Control Service on the MAG Series device to query the remote Active Directory server instead of the local authentication table when authenticating a user. The following steps add services and authentication options to the Access Control Service on the MAG Series device. The configuration of the SRX Series device remains unchanged.

When you reconfigure the realm's authentication server, the Access Control Service displays all roles or groups from the configured domain controller and its trusted domains. Establishing role mapping rules equates the authentication server's roles or groups to those defined on the Access Control Service.

To reconfigure remote authentication on the Access Control Service:

1. From the administrator console of the Access Control Service on the MAG Series device, select **Authentication>Auth. Servers**.

2. Choose the **Active Directory/Windows NT** server type, and click **Add New Server**.

3. Enter the profile of the new authentication server.

   a. Name the Active Directory server.

   b. Enter its NetBIOS domain name in the domain box.

      **NOTE:** You might receive the following message: "Either the server is not a domain controller of the domain, or the NetBIOS name of the domain is different from the Active Directory (LDAP) name." This message is informational and does not affect the processing of the authentication.

   c. Enter the Kerberos Realm name.

      The Kerberos realm name is the FQDN of the Active Directory domain. For example, if "mycompany" is the domain or NetBIOS name, mycompany.com is the Kerberos realm name.

   d. In the Domain Join Configuration section, enter the username and password of the UAC services account which has permission to join computers to the Active Directory domain.

      Select the Save credentials box.

   e. Enter the Container name.

      This is the name of the container in Active Directory where you created the UAC services account for the Access Control Service.

   f. Enter the Computer Name.

Specify the machine ID that the Access Control Service uses to join the specified Active Directory domain as a computer. This name is derived from the licence hardware ID of the Access Control Service in the following format: 0161MT2L00K2C0.

g. Verify that the join operation has succeeded.

   The Join Status indicator provides a color-coded status for the domain join operation as follows:

   - Gray: Not started
   - Yellow: In progress
   - Red: Failed to join
   - Green: Joined the domain

h. Select **Kerberos** and **NTLM v2** as the authentication protocols.

i. In the Trusts section, select the Allow trusted domains box.

j. Select **Enable SPNEGO**.

k. Use the Browse button to upload the keytab file that you created in the previous section.

l. Click **Save Changes** and **Test Configuration**.

4. Ensure that SSO is enabled.
   a. Select **Users>User Realms** and the realm name.

   b. Select the Active Directory server name from the **Auth Server** list.

   c. Select the **Authentication Policy** tab.

   d. Verify that the **SSO** option is selected.

   e. Click **Save Changes**.

5. Create role-mapping policies for groups acquired from the authentication server.

Groups from the Active Directory authentication server need to be mapped to roles on the Access Control Service. You first need to create roles, and then map one or more groups to the appropriate role.

a. Select the Role Mapping tab.

b. Click **New Rule**, enter a role name, and click **Save Changes**.

   You do not need to add users to the role. Create as many roles as needed to map the groups from the Active Directory authentication server.

c. Click **Groups**, and select **Search** to list the groups defined in the domain controller.

d. Select the group names that you want to map to the new role.

e. Repeat steps b through d to create and map other groups.

f. Click **Save Changes**.

***Configuring Endpoint Browsers for the SPNEGO***

**Step-by-Step Procedure**

Ensure that endpoint browsers have SPNEGO enabled. For further information, see your third-party documentation.

- Internet Explorer

  From **Security>Local Intranet>Sites>Advanced** add the trusted URL.

  IE performs SPNEGO without any further endpoint configuration but the user is prompted for a username and password. The username and password can be cached.

  To provide single sign-on support, an Internet Explorer configuration can be pushed by configuring a group policy on the Active Directory server. See your third-party documentation for further information.

  Integrated Windows Authentication must be enabled. Use the **Tools>Internet Options>Advanced>Security>Enable Integrated Windows Authentication** path to verify that IWA is enabled.

- Firefox (Windows and MacOS)

  The configuration is in a hidden location. For the URL, type **about:config** and search for the word **trusted**. The required key is the comma separated parameter named **network.negotiate-auth.trusted-uris**.

  > **NOTE:** You need to specify the URL of the resource (in this solution, the FQDN or domain controller value UCDC.com).

- Chrome

  Use the Internet Explorer setting. From **Security>Local Intranet>Sites>Advanced** add the trusted URL.

  An internet Explorer configuration can also be pushed by configuring a group policy on the Active Directory server. This configuration is honored by Chrome.

After successful authentication, the standard agentless page is shown along with a second window with the protected resource (unless a pop-up blocker prevents this).

SEE ALSO

*Authentication and Integrated User Firewalls Feature Guide*

## Obtaining Username and Role Information Through Firewall Authentication

User role firewall policies can be integrated with firewall authentication both to authenticate users and to retrieve username and role information. The information is mapped to the IP address of the traffic, stored in the firewall authentication table, and used for user role firewall policy enforcement.

The following CLI statements configure firewall authentication for user role firewall enforcement.

1. If not already established, define the access profile to be used for firewall authentication. You can skip this step if an existing access profile provides the client data needed for your implementation.

   The access profile is configured in the **[edit access profile]** hierarchy as with other firewall authentication types. It defines clients as firewall users and the passwords that provide them access. Use the following command to define a profile and add client names and passwords for firewall authentication.

   ```
   set access profile profile-name client client-name firewall-user password pwd
   ```

2. If HTTPS traffic is expected, define the access profile to be used for SSL termination services. You can skip this step if an existing SSL termination profile provides the services needed for your implementation.

   The SSL termination profile is configured in the **[edit services ssl]** hierarchy.

   ```
   set services ssl termination profile ssl-profile-name server-certificate certificate-type
   ```

3. Enable the firewall authentication table as an authentication source.

   ```
   set security user-identification authentication-source firewall-authentication priority priority
   ```

   The priority value determines the sequence in which authentication sources are checked. The default value is 150 for the firewall authentication table. (It is 100 for the local authentication table and 200 for the Unified Access Control (UAC) authentication table.) By default, the local authentication table is checked first, the firewall authentication table is next, and the UAC authentication table is third if it is enabled. You can change this sequence by changing the priority value of one or more of the tables.

4. Configure policies that permit traffic for user firewall authentication.

   ```
   edit security policies from-zone zone to-zone zone policy policy-name
   set match source-identity unauthenticated-user
   set then permit firewall-authentication user-firewall access-profile profile-name ssl-termination-profile
     profile-name
   ```

When unauthenticated traffic is permitted for firewall authentication, the user is authenticated based on the access profile configured in this statement. The **ssl-termination-profile** option is needed only for HTTPS traffic.

By specifying the authentication type **user-firewall**, the firewall authentication table is propagated with the IP address, the username, and any group names associated with the authenticated user. (Group names from firewall authentication are interpreted as roles by the user role firewall.) Any further traffic from this IP address will match the IP address in the firewall authentication table, and not require authentication. The associated username and roles are retrieved from the table for use as potential match criteria in subsequent security policies.

## RELATED DOCUMENTATION

Understanding User Authentication for Security Devices | 31

Firewall User Authentication Overview | 35

4

CHAPTER

# Integrated User Firewall


Integrated User Firewall Overview | **177**

Configure Integrated User Firewall | **196**

Configure Captive Portal for Unauthenticated Browsers | **223**

Manage Event logs to Generate IP Address-to-User Mapping | **227**

Logging User Identity Information Based on Zones | **235**

Control Network Access Using Device Identity Authentication | **243**

# Integrated User Firewall Overview

IN THIS SECTION


- Overview of Integrated User Firewall | 177
- Understanding Active Directory Authentication Tables | 182
- Understanding the Invalid Authentication Table Entry Timeout Setting | 190
- LDAP Functionality in Integrated User Firewall | 194


## Overview of Integrated User Firewall

IN THIS SECTION


- Integrated User Firewall and Authentication Sources | 177
- Benefits of Integrated User Firewall | 178
- How the Integrated User Firewall Works | 178
- Deployment Scenario for User Firewall Integration with Windows Active Directory | 179
- Limitations | 181


This topic includes the following sections:

### Integrated User Firewall and Authentication Sources

The SRX Series device already supports Unified Access Control (UAC) integration with Network Access Control (NAC) and a user firewall that can derive its authentication source from Windows Active Directory via the UAC MAG Series Junos Pulse Gateway. Many customers want simple user firewall functionality without full NAC, and do not want the additional cost or complexity of user role firewall (which has Active Directory dependencies such as Kerberos, SPNEGO on Browsers, Active Directory DNS/Certs, and UAC configuration).

The integrated user firewall fulfills the requirement for simplicity. It retrieves user-to-IP address mappings from the Windows Active Directory for the firewall policies usage as match criteria. This feature consists of the SRX Series or NFX Series device polling the event log of the Active Directory controller to determine,

by username and source IP address, who has logged in to the device. Then the username and group information are queried from the LDAP service in the Active Directory controller. Once the device has the IP address, username, and group relationship information, it generates authentication entries. With the authentication entries, the device user firewall module enforces user-based and group-based policy control over traffic.

Integrated UserFW is optimal for deployments with 1 or 2 SRX Series devices in an environment, supporting up to 2 domains, and up to 20 domain controllers. For deployments with 3 or more SRX Series devices, more than 2 domains, more than 20 domain controllers, or where additional features are required, JIMS solution is a better choice see "Configure Juniper Identity Management Service to Obtain User Identity Information" on page 275 for more information.

### Benefits of Integrated User Firewall

The integrated user firewall feature introduces an authentication source via integration with Microsoft Active Directory technology.

- Provides visibility into who is accessing the SRX Series or NFX Series and best-effort security for access to the device.
- A single-box solution, requiring only SRX Series or NFX Series.
- Requires fewer configuration steps than the UAC integration with NAC, which uses the UAC MAG Series for SRX Series devices.
- Does not require the configuration of a captive portal, although that option is available to enforce on users who do not authenticate.
- Ideal for small-to-medium businesses and low-scale deployments.
- Supports high availability (HA).

### How the Integrated User Firewall Works

At a high level, this feature involves the UserID process in Routing Engine, which reads the Windows event log from the Active Directory controller and abstracts IP address-to-user mapping information. The process correlates users to the groups to which they belong, via the LDAP protocol with the LDAP service in the Active Directory controller. Thus, the process has gathered enough information to generate authentication entries. The network administrator then references the authentication entries in user firewall security policies to control traffic.

Starting in Junos OS Release 17.4R1, you can assign IPv6 addresses to Active Directory domain controllers and the LDAP server. Prior to Junos OS Release 17.4R1, only IPv4 was supported.

A more detailed explanation of how this feature works is as follows:

1. The SRX Series or NFX Series device reads the Active Directory event log to get source IP address-to-username mapping information. To do so, a process in the SRX Series Routing Engine implements a Windows Management Instrumentation (WMI) client with Microsoft Distributed COM/Microsoft RPC stacks and an authentication mechanism to communicate with a Windows Active Directory controller in an Active Directory domain. Using event log information retrieved from the Active Directory controller, the process knows the IP addresses of active Active Directory users and abstracts IP-to-Active Directory username mapping information. The process monitors Active Directory event log changes via the same WMI DCOM interface to adjust local mapping information to reflect any change in the Active Directory server. Starting in Junos OS Release 17.4R1, the SRX Series WMI client can read the Active Directory event log to obtain IPv6 addresses, in addition to IPv4 addresses. Prior to Junos OS Release 17.4R1, the WMI client could read only IPv4 addresses.

2. The process uses LDAP to query the LDAP service interface of the Active Directory to identify the groups to which users belong. Having the IP address, the Active Directory user, and the groups, the process can generate authentication entries accordingly.

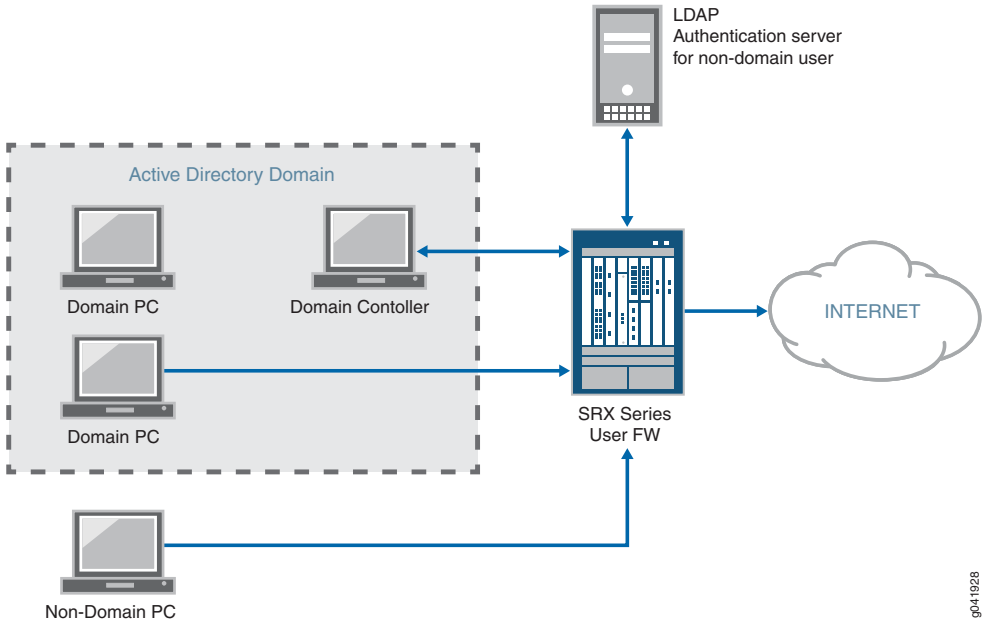
3. The process pushes the authentication entries to the Packet Forwarding Engine authentication table. The Packet Forwarding Engine uses the entries and user policy to apply user firewall access control to traffic.

For SRX Series devices, this feature supports two domains and up to 10 Active Directory controllers in a domain. For NFX Series devices, this feature supports two domains and up to 5 Active Directory controllers in a domain.

### Deployment Scenario for User Firewall Integration with Windows Active Directory

Figure 15 on page 180 illustrates a typical scenario where the integrated user firewall feature is deployed. Users in the Active Directory domain and users outside the Active Directory domain want access to the Internet through an SRX Series device. The domain controller might also act as the LDAP server.

**Figure 15: Scenario for Integrated User Firewall**




The SRX Series device reads and analyzes the event log of the domain controller and generates an authentication table as an Active Directory authentication source for this feature. The user firewall is aware of any domain user on an Active Directory domain device via the Active Directory authentication source. The SRX Series device administrator configures a user firewall policy that enforces the desired user-based or group-based access control.

For any non-domain user or domain user on a non-domain machine, the administrator specifies a captive portal to force the user to do firewall authentication (if the SRX Series supports captive portal for the traffic type). After the user enters a name and password and passes firewall authentication, the SRX Series gets firewall authentication user/group information and can enforce user firewall policy to control the user accordingly.

In addition to captive portal, if the IP address or user information is not available from the event log, the user can again log in to the Windows PC to generate an event log entry. Then the system generates the user's authentication entry accordingly.

Starting with Junos OS 17.4R1, the SRX Series devices and NFX Series devices can search the Active Directory authentication table, the local authentication table, and the firewall authentication table for information based on IPv6 addresses. Prior to Junos OS Release 17.4R1, only IPv4 was supported.

For example, prior to Junos OS Release 17.4R1, if the specification for the source-address field of a security policy was set to "any", implying also IPv6, integrated user firewall ignored the traffic rather than searching for a matching user entry in the authentication tables.

Consider the following scenario and security policy configuration in light of support for IPv6 addresses. When traffic arrives at the SRX Series device from a user whose IP address (source-address) is 2001:db8::1:1, given a source-identity match—that is, as illustrated in this example, the user belongs to the role2 group—the SRX Series UserFW module is able to authenticate the user, and it sets up a session for the user's traffic flow.

```
user@host set security policies from-zone trust to-zone untrust policy p1 match source-address any
user@host set security policies from-zone trust to-zone untrust policy p1 match destination-address any
user@host set security policies from-zone trust to-zone untrust policy p1 match application any
user@host set security policies from-zone trust to-zone untrust policy p1 match source-identity role2
user@host set security policies from-zone trust to-zone untrust policy p1 then permit
```

Prior to Junos OS Release 17.4R1, when **any-ipv6** was specified for the source-address field in a user firewall security policy, a commit warning message was issued indicating that only IPv4 addresses were supported. That message is no longer issued.

### Limitations

- Windows Active Directory controllers earlier than Windows 2003 are not supported.
- Tracking the status of non-Windows Active Directory users is not supported.
- Logical systems are not supported.
- The WMIC does not support multiple users logged on to the same PC.
- Domain controllers and domain PCs must be running Windows OS. The minimum support for a Windows client is Windows XP. The minimum support for a server is Windows Server 2003.
- You cannot use the Primary Group, whether by its default name of Domain Users, or any other name (if you happened to have changed it), in integrated user firewall configurations.

  When a new user is created in Active Directory, the user is added to the global security group Primary Group which is by default called Domain Users. The Primary Group is less specific than other groups created in Active Directory because all users belong to it. Consequently, it can become very large.

SEE ALSO

Manage Event logs to Generate IP Address-to-User Mapping | 227

show services user-identification authentication-table | 805

user-identification (Services) | 665

# Understanding Active Directory Authentication Tables

IN THIS SECTION


- Active Directory Authentication as an Authentication Source | 182
- Active Directory Authentication Tables | 182
- State Information for Active Directory Authentication Table Entries | 185
- Active Directory Authentication Table Management | 186
- Timeout Interval for Table Entries | 188


This topic includes the following sections:

## Active Directory Authentication as an Authentication Source

On an SRX Series device or NFX Series device, user information tables serve as the authentication source for information required by firewall security policies. The device supports various user information tables including local, user firewall, and Unified Access Control (UAC) types. The integrated user firewall feature introduces another type of authentication source—Active Directory authentication.

The integrated user firewall feature gathers user and group information for Active Directory authentication by reading domain controller event logs, probing domain PCs, and querying Lightweight Directory Access Protocol (LDAP) services within the configured Windows domain. Up to two Windows domains are supported.

From the user and group information, the integrated user firewall feature generates an Active Directory authentication table on the Routing Engine of the device, which then pushes the authentication table to the Packet Forwarding Engine. Security policies use the information in the table to authenticate users and to provide access control for traffic through the firewall.

## Active Directory Authentication Tables

The Active Directory authentication table contains the IP address, username, and group mapping information that serves as the authentication source for the device integrated user firewall feature. Information in the table is obtained by reading Windows Active Directory domain controller event logs, probing domain PCs, and querying LDAP services within a specified Windows domain.

Reading domain controller event logs generates a list of IP address-to-user mapping information that is used to create entries in the Active Directory authentication table. Once entries have been added in the table, a query is sent to the LDAP server for user-to-group mapping information.

Starting with Junos OS Release 17.4R1, the SRX Series and NFX Series device supports IPv6 addresses for user firewall (UserFW) authentication. IPv6 addresses can be used in Active Directory authentication table entries, local authentication table entries, and firewall authentication table entries. They can also be used for device identity addresses with Active Directory as the authentication source. An IPv6 address can also be configured for the Windows domain controller. Previously only IPv4 addresses were supported.

In addition to IPv4, IPv6 traffic can match any security policy configured for source identity. Previously, if a security policy was configured for source identity and “any” was specified for its IP address, the SRX Series user firewall ignored the IPv6 traffic.

When user traffic arrives at the device, the Active Directory authentication table is searched for an entry corresponding to the source IP address of the traffic to authenticate the user. The device can also search for an entry in the local authentication table and the firewall authentication table, if an entry is not found in the Active Directory authentication table.

The device supports use of IPv6 and IPv4 addresses associated with source identities in security policies. If an entry exists, policies matching that entry are applied to the traffic and access is allowed or denied.

The LDAP server returns all group information; this includes not only information about the groups you directly belong to, but also all the parent (and parent of the parent and so on) groups that you belong to. Group information returned from the LDAP server is compared with the source identity in security policies. If there is a match, Active Directory authentication table entries are updated to include only the group information provided in the security policy. In this way, only relevant group information is listed in the authentication table. Whenever source identity is updated, the authentication table is also updated to reflect the up-to-date relevant group information for all listed users.

**NOTE:** The integrated user firewall feature for both Active Directory authentication and ClearPass authentication will manage up to 2048 sessions for each user for whom there is a user identity and authentication entry in the authentication table. There might be additional sessions associated with a user beyond the 2048 supported sessions, but they are not managed by integrated user firewall. When an authentication entry in an authentication table is deleted, integrated user firewall only closes sessions that are associated with that entry. It will not close sessions that it does not manage. That is, sessions that are not associated with the authentication entry are not closed.

Only IPv4 addresses are supported for ClearPass.

Table 6 on page 184 lists Active Directory authentication table support by SRX Series devices and NFX Series devices. Platform support depends on the Junos OS release in your installation.

**Table 6: Active Directory Authentication Table Support**

| Devices | Active Directory Authentication Table Entries | Domains | Active Directory Controllers |
|---|---|---|---|
| SRX100, SRX110, SRX210, SRX220 | 500 | 1 | 5 |
| SRX240 | 1000 | 1 | 5 |
| SRX300 | 500 | 1 | 5 |
| SRX320 | 500 | 1 | 5 |
| SRX340, 345 | 1000 | 1 | 5 |
| SRX550M | 5000 | 2 | 10 |
| SRX650 | 5000 | 2 | 10 |
| SRX1400 | 20,000 | 2 | 10 |
| SRX1500 | 20,000 | 2 | 10 |
| SRX3000 line | 50,000 | 2 | 10 |
| SRX4000 line | 50,000 | 2 | 10 |
| SRX5000 line | 256,000 | 2 | 10 |
| vSRX (2 vCPUs and 4 GB vRAM, 5 vCPUs and 8 GB vRAM ) | 5000 | 2 | 10 |
| vSRX (9 vCPUs and 16 GB vRAM, 17 vCPUs and 32 GB vRAM ) | 10,000 | 2 | 10 |
| NFX150 | 500 | 1 | 5 |

Once the maximum number of authentication table entries is reached, no additional entries are created.

To be compliant with the Active Directory authentication table, entries must adhere to the following parameters:

- Usernames are limited to 64 characters.
- Group names are limited to 64 characters.
- Each entry can be associated with up to 200 relevant groups (configured in the source identity field). For example, if you belong to 1000 groups in LDAP and out of these, no more than 200 groups are configured in the source identity field, you are compliant with the Active Directory authentication table.

The Active Directory authentication table must be enabled as the authentication source for integrated user firewall information retrieval in the Windows Active Directory environment. Use the following statement for that purpose:

```
user@host# set security user-identification authentication source active-directory-authentication-table priority
    priority
```

**NOTE:** The **priority** option specifies the sequence in which user information tables are checked. Using the lowest setting for the Active Directory authentication source specifies the highest priority, meaning that the Active Directory authentication source is searched first.

### State Information for Active Directory Authentication Table Entries

Active Directory authentication table entries can be in one of four states:

**Initial**—Specifies that IP address-to-user mapping information was obtained by reading domain controller event logs and an entry was added to the authentication table. Entries in this state are changed to valid when the table is pushed from the Routing Engine to the Packet Forwarding Engine.

**Valid**—Specifies that a valid entry was obtained by reading domain controller event logs or that a valid response was received from a domain PC probe and the user is a valid domain user.

**Invalid**—Specifies that an invalid response was received from a domain PC probe and the user is an invalid domain user.

**Pending**—Specifies that a probe event generated an entry in the authentication table, but no probe response has been received from the domain PC. If a probe response is not received within 90 seconds, the entry is deleted from the table.

For a list of probe responses, see “Understanding Integrated User Firewall Domain PC Probing” on page 231.

To display Active Directory authentication entries, along with their state information, use the following command:

```
user@host>show services user-identification active-directory-access active-directory-authentication-table all
```

```
Domain: www.example1.net
Total count: 3
Source IP       Username      Groups          State
 2001:db8::1:1     u2            r1, r3, r4      initial
 192.168.10.3     u3           r5, r6, r4      pending
 2001:db8::2:1    u4           r3, r4          initial

Domain: www.example2.net
Total count: 2
Source IP       Username      Groups          State
 10.1.1.2         u4           r1, r3, r4      valid
 10.1.1.3         u5           r5, r6, r4      invalid
```

Command options allow you to display information by **user** or **group**, and to define additional output levels—**brief**, **domain**, **extensive**, **node**.

### Active Directory Authentication Table Management

Windows domain environments are constantly changing as users log in and out of the network and as network administrators modify user group information. The integrated user firewall feature manages changes in the Windows domain by periodically reading domain controller event logs and querying the LDAP server for user-to-group mapping information. That information is used in updating the Active Directory authentication table as appropriate.

Additionally, a probe function is provided to address changes that occur between reading event logs, or to address the case where event log information is lost. An on-demand probe is triggered when client traffic arrives at the firewall but a source IP address for that client cannot be found in the table. And at any point, manual probing is available to probe a specific IP address

Changes to the Active Directory Authentication table also occur due to source identity changes in the security policy configuration.

Table 7 on page 187 describes events that trigger an Active Directory authentication table update.

**Table 7: Events Triggering Active Directory Authentication Table Updates**

| Event | Active Directory Authentication Table Update |
|---|---|
| A domain controller event log is read at configured intervals. | New IP address-to-user entries are added in the authentication table in initial state. Group information is retrieved from the LDAP server.<br><br>When the authentication entry is pushed to Packet Forwarding Engine, the state is changed to valid. |
| An on-demand or manual probe is sent to a domain PC. | An entry is added in the authentication table in pending state. If a probe response is not returned within 90 seconds, the state of the entry is deleted. |
| An on-demand or manual probe response is received from a domain PC. | Based on the response, entries in pending state are changed to valid or invalid. For valid responses, the group information is retrieved from the LDAP server. For invalid responses, the entry is marked as invalid. |
| An LDAP server query identifies new user-to-group mapping information. | Entries are updated with the group information. |
| An LDAP server query identifies deleted user information. | Entries associated with that user are deleted from the table. |
| An LDAP server query identifies deleted group information. | The affected group information is updated.<br><br>For example, user2 belongs to group2, and group2 belongs to group1. And, group1 is listed as a source-identity for group2. For any authentication entry of user2, group1 is listed in its relevant groups. However, if group2 is removed from the LDAP server, user2 loses the connection with group1, and as a result, group1 is removed from the user2 authentication table. |
| An LDAP server query identifies added group information. | If the group is referenced in a security policy, entries associated with this group are updated to add the group information. |
| The source identity information is removed from a security policy configuration. | Entries associated with the source identity are deleted from Active Directory authentication table. |

**NOTE:** If an entry is deleted from the table, any sessions attached to that entry are also deleted. If an entry in the table is updated to add or remove group information, there is no impact to existing sessions for that entry.

**NOTE:** When you use the CLI to delete an Active Directory authentication entry, the system closes the related session and writes a session-close message to the log file. However, the session-close message does not contain the source identity information for the user, that is, the user and user group information.

To manually delete an entry from the table, use the **request services user-identification active-directory-access active-directory-authentication-table** command. Options exist for deleting a specific IP address, domain, group, or user.

To clear the contents of the Active Directory authentication table, use the **clear services user-identification active-directory access active-directory-authentication-table** command.

### Timeout Interval for Table Entries

When a user is no longer active, a timer is started for that user's entry in the Active Directory authentication table. When time is up, the user's entry is removed from the table. Entries in the table remain active as long as there are sessions associated with the entry.

To set the timeout value, use the following statement:

```
user@host# set services user-identification active-directory-access authentication-entry-timeout minutes
```

The default **authentication-entry-timeout** interval is 30 minutes. To disable timeouts, set the interval to 0.

**NOTE:** We recommend that you disable timeouts when disabling on-demand probing in order to prevent someone from accessing the Internet without logging in again.

To view timeout information for Active Directory authentication table entries, use the following command:

```
user@host>show services user-identification active-directory-access active-directory-authentication-table all
    extensive
```

```
Domain: www.example1.net
Total entries: 2
Source IP: 192.168.1.2
Username: u2
Groups: r1, r3, r4
State: initial
Access start date: 2014-03-22
Access start time: 10:56:58
Age time: 20 min

Source IP: 192.168.1.3
Username: u3
Groups: r5, r6, r4
State: pending
Access start date: 2014-03-22
Access start time: 10:46:58
Age time: 10 min
```

This example shows that the timer has started for two entries—the entry for user u2 will time out in 20 minutes, while the entry for user u3 will time out in 10 minutes. When session traffic is associated with an entry, the age time value changes to “infinite.”

SEE ALSO

Understanding Integrated User Firewall Domain PC Probing | 231

active-directory-authentication-table | 402

user-identification (Services) | 665

# Understanding the Invalid Authentication Table Entry Timeout Setting

IN THIS SECTION


- Timeout Setting for Invalid Authentication Entries | 190
- How the Invalid Authentication Entry Timeout Works for Windows Active Directory | 191
- How the Invalid Authentication Entry Timeout Works for SRX Series and NFX Series Aruba ClearPass | 192


## Timeout Setting for Invalid Authentication Entries

Starting in Junos OS Release 15.1X49-D100, for SRX Series devices and vSRX, you can protect invalid user authentication entries in an authentication table from expiring before the user can be validated by configuring a timeout setting that is specific to invalid entries. The invalid authentication entry timeout setting is separate from the common authentication entry timeout setting that is applied to valid entries.

Authentication entries in both the Windows Active Directory authentication table and the ClearPass authentication table contain a timeout value after which the entry expires. Prior to introduction of this feature, a single, common timeout setting was applied to valid and invalid authentication entries. That is, if an invalid authentication entry was created in either of these tables, the current setting of the common timeout for the table—which applied to all of the table's entries—was applied to it.

For both the Active Directory authentication table and the ClearPass authentication table, the invalid entry could expire before the user's identity could be validated. Here is what could cause that event to occur in each case:

- Windows Active Directory uses a mechanism to probe an unauthenticated user's device for user identity authentication information based on the IP address of the device. It is not uncommon for Windows to trigger a WMI probe that fails because it occurs before the user logs in. After an unsuccessful probe, the system generates an entry in the authentication table with an INVALID state for the IP address of the device. If you configured a value for the invalid timeout setting, that timeout is applied to the entry. If you did not configure a value for the invalid entry timeout setting, then its default timeout of 30 minutes is applied.

  The invalid authentication entry timeout setting is separate from the common authentication entry timeout setting that is applied to valid entries.

  Starting in Junos OS Release 17.4R1, the integrated user firewall supports IPv6 device addresses in the Windows Active Directory authentication table. Prior to Junos OS Release 17.4R1, only IPv4 addresses were supported.

- For the ClearPass feature, if an unauthenticated user attempts to join the network and the IP address of the user's device is not found—that is, it is not in the Packet Forwarding Engine—the device queries Aruba ClearPass for the user's information. If the query is unsuccessful, the system generates an INVALID authentication entry for the user. If you configured a value for the invalid timeout setting, that timeout is applied to the entry. If you did not configure the invalid entry timeout, then its default timeout of 30 minutes is applied to the new entry.

  **NOTE:** The invalid entry timeout is also applied to entries whose state is changed from valid or pending to INVALID.

You configure the timeout setting to be applied to invalid authentication entries in the Windows Active Directory authentication table and the ClearPass authentication table separately. If you do not configure a timeout setting, the invalid authentication entry timeout default value of 30 minutes is applied. The application and effect of the timeout value is determined differently for these authentication sources.

### How the Invalid Authentication Entry Timeout Works for Windows Active Directory

Use the following command to configure the invalid authentication entry timeout setting for entries in the Windows Active Directory authentication table. In this example, the invalid authentication entry timeout value is set to 40 minutes. That timeout value is applied to new invalid entries.

```
user@host# set services user-identification active-directory-access invalid-authentication-entry-timeout 40
```

The new timeout value is also applied to existing invalid entries but within the context of the current timeout value assigned to them and the timeout state. Suppose that the authentication table contains existing invalid entries to which an invalid authentication entry timeout setting or the default was previously applied. In this case, the new invalid entry timeout setting has effect on the timeout for these entries, but in a different way. For these entries, the original timeout setting—the time that has expired since the original timeout value was applied–and the new timeout setting collude to produce the resulting timeout value that is applied to the existing entry.

As Table 8 on page 192 shows, in some cases the resulting timeout is extended, in some cases it is shortened, and in some cases it causes the original timeout to expire and the invalid authentication entry to which is applies to be deleted.

**Table 8: How New Invalid Authentication Entry Timeout Settings Affect Timeout Settings for Existing Invalid Entries in the Active Directory Authentication Table**

| Original Invalid Entry Timeout Setting for Existing Entry | Elapse Time | New Invalid Entry Timeout Configuration Setting | Resulting Timeout Setting for Existing Invalid Entry |
|---|---|---|---|
| 20 minutes | 5 minutes | 50 minutes | 45 minutes |
| 50 minutes | 10 minutes | 20 minutes | 10 minutes |
| 50 minutes | 40 minutes | 20 minutes | Timeout expired and entry is removed from the authentication table |
| 40 minutes | 20 minutes | 0 | 0 |

**NOTE:** Just as the new invalid timeout entry is imposed on that of old invalid entries, producing various and unique results, a new invalid entry is subject to the same rules and effects when the invalid entry timeout value is changed.

### How the Invalid Authentication Entry Timeout Works for SRX Series and NFX Series Aruba ClearPass

Use the following command to configure the invalid authentication entry timeout for entries in the ClearPass authentication table. In this example, invalid authentication entries in the ClearPass authentication table expires 22 minutes after they are created.

```
user@host# set services user-identification authentication-source aruba-clearpass
    invalid-authentication-entry-timeout 22
```

- When you initially configure the invalid authentication entry timeout value for ClearPass, it is applied to any invalid authentication entries that are generated *after* it was configured. However, all existing invalid authentication entries retain the default timeout of 30 minutes.
- If you do not configure the invalid authentication entry timeout setting, the default timeout of 30 minutes is applied to all invalid authentication entries.

  If you configure the invalid authentication entry timeout setting and delete it later, the default value is applied to new invalid authentication entries generated after the deletion. However, any existing invalid authentication entries to which a configured value had been applied previously retain that value.

- If you change the setting for the invalid authentication entry timeout value, the new value is applied to all invalid authentication entries that were created *after* the value was changed. However, all existing invalid authentication entries retain the former invalid authentication entry timeout setting applied to them. Those entries to which the default value of 30 minutes had been applied previously retain that setting.
- When the pending or valid state of an entry is changed to invalid, the invalid authentication entry timeout setting is applied to it.

  When the state of an invalid authentication entry is changed to pending or valid, the invalid authentication entry timeout setting is no longer applicable to it. The timeout value set for the common authentication entry timeout is applied to it

Table 9 on page 193 shows how a new invalid entry timeout value affects new and existing invalid entries.

**Table 9: How New Invalid Authentication Entry Timeout Settings Affect Timeout Settings for Invalid Entries in the ClearPass Authentication Table**

| Invalid Entry Timeout Setting | Intial Invalid Entry Timeout Setting | Elapse Time | New Invalid Entry Timeout Configuration Setting | Final Timeout Setting for Existing Invalid Entry |
|---|---|---|---|---|
| New invalid authentication entry | | | 50 | 50 |
| Existing invalid entry timeout | 20 | 5 | 50 | 15 |
| Existing invalid entry timeout | 0 | 40 | 20 | 0 |
| Existing invalid entry timeout | 40 | 20 | 0 | 20 |

## RELATED DOCUMENTATION

invalid-authentication-entry-timeout | 526
firewall-authentication-forced-timeout | 498

# LDAP Functionality in Integrated User Firewall

IN THIS SECTION


- Role of LDAP in Integrated User Firewall | 194
- LDAP Server Configuration and Base Distinguished Name | 194
- LDAP's Authentication Method | 195
- LDAP Server's Username, Password, and Server Address | 195
- Caching and Calculation of User-to-Group Mappings | 195
- Updating Group Information in the Authentication Entry Table | 196
- LDAP Server Status and Statistics | 196
- Active Directory Autodiscovery | 196


This topic includes the following sections:

## Role of LDAP in Integrated User Firewall

In order to get the user and group information necessary to implement the Integrated User Firewall feature, the SRX Series evice uses the Lightweight Directory Access Protocol (LDAP). The device acts as an LDAP client communicating with an LDAP server. In a common implementation scenario of the integrated user firewall feature, the domain controller acts as the LDAP server. The LDAP module in the device, by default, queries the Active Directory in the domain controller.

The device downloads user and group lists from the LDAP server. The device also queries the LDAP server for user and group updates. The device downloads a first-level, user-to-group mapping relationship and then calculates a full user-to-group mapping.

The use of "LDAP" in this section applies specifically to LDAP functionality within the integrated user firewall feature.

## LDAP Server Configuration and Base Distinguished Name

Most of the LDAP server configuration is optional, leveraging the common implementation scenario where the domain controller acts as the LDAP server. The device periodically (every two minutes) queries the LDAP server to get the user and group information changed since the last query.

### LDAP's Authentication Method

By default, the LDAP authentication method uses simple authentication. The client's username and password are sent to the LDAP server in plaintext. Keep in mind that the password is clear and can be read from the network.

To avoid exposing the password, you can use simple authentication within an encrypted channel [namely Secure Sockets layer (SSL)], as long as the LDAP server supports LDAP over SSL (LDAPS). After enabling SSL, the data sent from the LDAP server to the device is encrypted. To enable SSL, see the **user-group-mapping** statement.

### LDAP Server's Username, Password, and Server Address

The LDAP server's username, password, IP address, and port are all optional, but they can be configured.

- If the username and password are not configured, the system uses the configured domain controller's username and password.
- If the LDAP server's IP address is not configured, the system uses the address of one of the configured Active Directory domain controllers.
- If the port is not configured, the system uses port 389 for plaintext or port 636 for encrypted text.

### Caching and Calculation of User-to-Group Mappings

The device caches user-to-group mappings in its local database when the **show services user-identification active-directory-access user-group-mapping** operation is performed. This command displays the users who belong to a group or the groups to which a user belongs.

Three events cause a user-to-group mapping to be removed from the cache:

- A source-identity is removed from a referenced firewall policy (because only source-identities referenced in a policy are stored in the authentication table).
- The LDAP configuration is deleted from the customer's configuration, so all cached Active Directory user-to-group mappings for the domain are removed.
- The user-to-group mapping is deleted from the LDAP server.

The device periodically queries to get user and group information from the LDAP server in real time. The user list and the group list show only cached users or groups, not all users or groups in the LDAP server. From this information, the device calculates one-level mapping relationships. The user list, group list, and mapping are cached in the local database.

### Updating Group Information in the Authentication Entry Table

The device queries to get the changed users and groups based on the prior query results from the LDAP server. The device updates the local database and triggers an authentication entry update. Only user/group mappings that are already cached are updated. Other users and groups that are not in the database do not have their mapping relationships cached.

### LDAP Server Status and Statistics

You can verify the LDAP connection status by issuing the **show services user-identification active-directory-access user-group-mapping status** command.

You can see counts of queries made to the LDAP server by issuing the **show services user-identification active-directory-access statistics user-group-mapping** command.

### Active Directory Autodiscovery

The integrated user firewall feature provides the IP address and Active Directory name of the domain. The auto-discovery feature can use the Active Directory's global catalog feature and then query DNS for a list of global catalogs. The global catalogs in the list are typically provided in a weighted order based on criteria such as network location, system-set weights based on global catalog server size, and so on. Once the customer has the list of Active Directories, the customer can configure it for both event log reading and LDAP search.

SEE ALSO


- show services user-identification active-directory-access statistics | 795
- show services user-identification active-directory-access user-group-mapping | 799
- user-group-mapping | 663


# Configure Integrated User Firewall

IN THIS SECTION


- Example: Configuring Integrated User Firewall on SRX Series | 197
- Configuring Integrated User Firewall on NFX Devices | 209

- Example: Configuring Integrated User Firewall on SRX Series devices to Use Web-Redirect for Unauthenticated and Unknown Users | **211**
- Example: Configuring Integrated User Firewall on SRX Series devices to Use Web-Redirect-to-HTTPS to Authenticate Unauthenticated and Unknown Users | **216**


As the name denotes, integrated user firewall provides simpler user firewall functionality without the need of Unified Access Control (UAC) integration with network access control (NAC). Integrated user firewall collects user information through Lightweight Directory Access Protocol (LDAP), and by enforcing policies, access is allowed or denied.

## Example: Configuring Integrated User Firewall on SRX Series

**IN THIS SECTION**


- Requirements | **197**
- Overview | **198**
- Configuration | **198**
- Verification | **206**


This example shows how to implement the integrated user firewall feature by configuring a Windows Active Directory domain, an LDAP base, unauthenticated users to be directed to captive portal, and a security policy based on a source identity. All configurations in this example for the captive portal are over the Transport Layer Security (TLS).

### Requirements

This example uses the following hardware and software components:

- One SRX Series device
- Junos OS Release 12.1X47-D10 or later for SRX Series devices

No special configuration beyond device initialization is required before configuring this feature.

## Overview

In a typical scenario for the integrated user firewall feature, domain and non-domain users want to access the Internet through an SRX Series device. The SRX Series device reads and analyzes the event log of the domain controllers configured in the domain. Thus, the SRX Series device detects domain users on an Active Directory domain controller. Active Directory domain generates an authentication table as the Active Directory authentication source for the integrated user firewall. The SRX Series device uses this information to enforce the policy to achieve user-based or group-based access control.

For any non-domain user or domain user on a non-domain device, the network administrator can specify a captive portal to force the user to submit to firewall authentication (if the SRX Series device supports captive portal for the traffic type. For example, HTTP). After the user enters a name and password and passes firewall authentication, the SRX Series device gets firewall authentication user-to-group mapping information from the LDAP server and can enforce user firewall policy control over the user accordingly.

Starting with Junos OS Release 17.4R1, you can use IPv6 addresses for Active Directory domain controllers in addition to IPv4 addresses. To illustrate this support, this example uses 2001:db8:0:1:2a0:a502:0:1da as the address for the domain controller.

> **NOTE:** You cannot use the Primary Group, whether by its default name of Domain Users or any other name, if you changed it, in integrated user firewall configurations.
>
> When a new user is created in Active Directory (AD), the user is added to the global security group Primary Group which is by default Domain Users. The Primary Group is less specific than other groups created in AD because all users belong to it. Also, it can become very large.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set services user-identification active-directory-access domain example.net user-group-mapping ldap base
    DC=example,DC=net user administrator password $ABC123
set services user-identification active-directory-access domain example.net user administrator password
    $ABC123
set services user-identification active-directory-access domain example.net domain-controller ad1 address
    2001:db8:0:1:2a0:a502:0:1da
set access profile profile1 authentication-order ldap
set access profile profile1 authentication-order password
```

```
set access profile profile1 ldap-options base-distinguished-name CN=Users,DC=example,DC=net
set access profile profile1 ldap-options search search-filter sAMAccountName=
set access profile profile1 ldap-options search admin-search distinguished-name
    CN=Administrator,CN=Users,DC=example,DC=net
set access profile profile1 ldap-options search admin-search password $ABC123
set access profile profile1 ldap-server 192.0.2.3
set access profile profile1 ldap-server 192.0.2.3 tls-type start-tls
set access profile profile1 ldap-server 192.0.2.3 tls-peer-name peername
set access profile profile1 ldap-server 192.0.2.3 tls-timeout 3
set access profile profile1 ldap-server 192.0.2.3 tls-min-version v1.2
set access profile profile1 ldap-server 192.0.2.3 no-tls-certificate-check
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 match source-identity unauthenticated-user
set security policies from-zone trust to-zone untrust policy p1 match source-identity unknown-user
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    access-profile profile1
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    domain example.net
set security policies from-zone trust to-zone untrust policy p2 match source-address any
set security policies from-zone trust to-zone untrust policy p2 match destination-address any
set security policies from-zone trust to-zone untrust policy p2 match application any
set security policies from-zone trust to-zone untrust policy p2 match source-identity “example.net\user1”
set security policies from-zone trust to-zone untrust policy p2 then permit
set security user-identification authentication-source active-directory-authentication-table priority 125
```

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To establish a Windows Active Directory domain, to configure captive portal, and to configure another security policy, perform the steps in this section.

Once configured, when traffic arrives, the SRX Series device consults the user firewall process, which in turn consults the Active Directory authentication source to determine whether the source is in its authentication table. If the user firewall hits an authentication entry, the SRX Series device checks the policy configured in Step 4 for further action. If the user firewall does not hit any authentication entry, the SRX Series device checks the policy configured in Step 3 to enforce the user to do captive portal.

1. Configure the LDAP base distinguished name.

   ```
   [edit services user-identification]
   ```

```
user@host# set active-directory-access domain example.net user-group-mapping ldap base
    DC=example,DC=net user administrator password $ABC123
```

2. Configure a domain name, the username and password of the domain, and the name and IP address of the domain controller in the domain.

```
[edit services user-identification]
user@host# set active-directory-access domain example.net user administrator password $ABC123
user@host# set active-directory-access domain example.net domain-controller ad1 address
    2001:db8:0:1:2a0:a502:0:1da
```

3. Configure an access profile and set the authentication order and LDAP options.

```
[edit access profile profile1]
user@host# set authentication-order ldap
user@host# set authentication-order password
user@host# set ldap-options base-distinguished-name CN=Users,DC=example,DC=net
user@host# set ldap-options search search-filter sAMAccountName=
user@host# set ldap-options search admin-search distinguished-name
    CN=Administrator,CN=Users,DC=example,DC=net
user@host# set ldap-options search admin-search password $ABC123
user@host# set ldap-server 192.0.2.3
user@host# set ldap-server 192.0.2.3 tls-type start-tls
user@host# set ldap-server 192.0.2.3 tls-peer-name peername
user@host# set ldap-server 192.0.2.3 tls-timeout 3
user@host# set ldap-server 192.0.2.3 tls-min-version v1.2
user@host# set ldap-server 192.0.2.3 no-tls-certificate-check
```

> **NOTE:** When the **no-tls-certificate-check** option is configured, the SRX Series device ignores the validation of the server's certificate and accepts the certificate without checking.

4. Configure a policy for the source-identity "unauthenticated-user" and "unknown-user" and enable the firewall authentication captive portal. Configuring the source identity is required in case there is no authentication sources configured, it is disconnected.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set match source-address any
```

```
user@host# set match destination-address any
user@host# set match application any
user@host# set match source-identity unauthenticated-user
user@host# set match source-identity unknown-user
user@host# set then permit firewall-authentication user-firewall access-profile profile1
user@host#set then permit firewall-authentication user-firewall domain example.net
```

5. Configure a second policy to enable a specific user.

```
[edit security policies from-zone trust to-zone untrust policy p2]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set match source-identity “example.net\user1”
user@host# set then permit
```

> **NOTE:** When you specify a source identity in a policies statement, prepend the domain name and a backslash to the group name or username. Enclose the combination in quotation marks.

6. Set the Active Directory authentication table as the authentication source for integrated user firewall information retrieval and specify the sequence in which user information tables are checked.

```
[edit security]
user@host# set user-identification authentication-source active-directory-authentication-table priority 125
```

**NOTE:** You must set the Active Directory authentication table as the authentication source for integrated user firewall information retrieval and specify the sequence in which user information tables are checked using the command **set security user-identification authentication-source active-directory-authentication-table priority *value***.

The default value of this option is 125. The default priority for all the authentication sources is as follows:

- Local authentication: 100
- Integrated user firewall: 125
- User role firewall: 150
- Unified Access Control (UAC): 200

The field **priority** specifies the sources for the Active Directory authentication table. The value set determines the sequence for searching among various supported authentication tables to retrieve a user role. Note that these are the only currently supported values. You can enter any value from 0 through 65,535. The default priority of the Active Directory authentication table is 125. This means that even if you do not specify a priority value, the Active Directory authentication table will be searched starting at sequence of value 125 (integrated user firewall).

For more details, see "Understanding Active Directory Authentication Tables" on page 182 and **active-directory-authentication-table**.

***(Optional) Configuration of PKI and SSL Forward Proxy to Authenticate Users***

**Step-by-Step Procedure**

Optionally, for non-domain users, you can configure public key infrastructure (PKI) to validate integrity, confidentiality, and authenticity of traffic. PKI includes digital certificates issued by the Certificate Authority (CA), certificate validity and expiration dates, details about the certificate owner and issuer, and security policies.

> **NOTE:** For any non-domain user or domain user on a non-domain machine, the administrator specifies a captive portal to force the user to do firewall authentication (if the SRX Series device supports captive portal for the traffic type). After the user enters a name and password and passes firewall authentication, the SRX Series device gets firewall authentication user/group information and can enforce the user firewall policy to control the user accordingly. In addition to captive portal, if the IP address or user information is not available from the event log, the user can again log in to the Windows PC to generate an event log entry. Then the system generates the user's authentication entry accordingly.

To enable the SRX Series device to authenticate the users through HTTPs, the SSL forward proxy must be configured and enabled. You need to generate a local certificate, add an SSL termination profile, add an SSL proxy profile, and reference the SSL proxy profile in the security policy. If the SSL forward proxy is not enabled, the SRX Series device cannot authenticate users who are using HTTPS, but for users who are using HTTP, FPT, and Telnet, the authentication can be performed as expected.

To generate PKI and enable SSL forward proxy, perform the following steps:

1. Generate a PKI public/private key pair for a local digital certificate.

```
user@host# request security pki generate-key-pair certificate-id ssl-inspect-ca size 2048 type rsa
```

2. Manually generate a self-signed certificate for the given distinguished name.

```
user@host# request security pki local-certificate generate-self-signed certificate-id ssl-inspect-ca
  domain-name www.mycompany.net subject "CN=www.mycompany.com,OU=IT,O=MY
  COMPANY,L=Sunnyvale,ST=CA,C=US" email security-admin@mycompany.net
```

3. Define the access profile to be used for SSL termination services. This option is available only on SRX5400, SRX5600, and SRX5800 devices.

```
user@host# set services ssl termination profile for_userfw server-certificate ssl-inspect-ca
```

4. Configure the loaded certificate as root-ca in the SSL proxy profile. This option is available only on SRX5400, SRX5600, and SRX5800 devices.

```
user@host# set services ssl proxy profile ssl-inspect-profile root-ca ssl-inspect-ca
```

5. Specify the **ignore-server-auth-failure** option if you do not want to import the entire CA list and you do not want dropped sessions. This option is available only on SRX5400, SRX5600, and SRX5800 devices.

```
user@host# set services ssl proxy profile ssl-inspect-profile actions ignore-server-auth-failure
```

6. Add an SSL termination profile into security policies. This option is available only on SRX5400, SRX5600, and SRX5800 devices.

```
user@host# set security policies from-zone untrust to-zone trust policy p1 then permit firewall-authentication
  user-firewall ssl-termination-profile for_userfw
```

***Results***

From configuration mode, confirm your integrated user firewall configuration by entering the **show services user-identification active-directory-access** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show services user-identification active-directory-access
    domain example.net {
        user {
            administrator;
            password "$ABC123"; ## SECRET-DATA
        }
        domain-controller ad1 {
            address 2001:db8:0:1:2a0:a502:0:1da;
        }
        user-group-mapping {
            ldap {
                base DC=example,DC=net;
                user {
                    administrator;
                    password "$ABC123"; ## SECRET-DATA
                }
            }
        }
    }
```

From configuration mode, confirm your policy configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security policies
    from-zone trust to-zone untrust {
        policy p1 {
            match {
                source-address any;
                destination-address any;
                application any;
                source-identity [ unauthenticated-user unknown-user ];
            }
            then {
                permit {
                    firewall-authentication {
                        user-firewall {
                            access-profile profile1;
                            domain example.net;
                        }
                    }
                }
            }
        }
        policy p2 {
            match {
                source-address any;
                destination-address any;
                application any;
                source-identity “example.net\user1”;
            }
            then {
                permit;
            }
        }
    }
```

From configuration mode, confirm your access profile configuration by entering the **show access profile profile1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access profile profile1
authentication-order [ ldap password ];
ldap-options {
```

```
    base-distinguished-name CN=Users,DC=example,DC=net;
    search {
        search-filter sAMAccountName=;
        admin-search {
            distinguished-name CN=Administrator,CN=Users,DC=example,DC=net;
            password "$ABC123"; ## SECRET-DATA
        }
    }
}
ldap-server {
    192.0.2.3 {
        tls-type start-tls;
        tls-timeout 3;
        tls-min-version v1.2;
        no-tls-certificate-check;
        tls-peer-name peername;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**


- Verifying Connectivity to a Domain Controller | **206**
- Verifying the LDAP Server | **207**
- Verifying Authentication Table Entries | **207**
- Verifying IP-to-User Mapping | **207**
- Verifying IP Probe Counts | **207**
- Verifying User-to-Group Mapping Queries | **208**


Confirm that the configuration is working properly.

***Verifying Connectivity to a Domain Controller***

**Purpose**

Verify that at least one domain controller is configured and connected.

**Action**

From operational mode, enter the **show services user-identification active-directory-access domain-controller status** command.

**Meaning**

The domain controller is shown to be connected or disconnected.

***Verifying the LDAP Server***

**Purpose**

Verify that the LDAP server is providing user-to-group mapping information.

**Action**

From operational mode, enter the **show services user-identification active-directory-access user-group-mapping status** command.

**Meaning**

The LDAP server address, port number, and status are displayed.

***Verifying Authentication Table Entries***

**Purpose**

See which groups users belong to and the users, groups, and IP addresses in a domain.

**Action**

From operational mode, enter the **show services user-identification active-directory-access active-directory-authentication-table all** command.

**Meaning**

The IP addresses, usernames, and groups are displayed for each domain.

***Verifying IP-to-User Mapping***

**Purpose**

Verify that the event log is being scanned.

**Action**

From operational mode, enter the **show services user-identification active-directory-access statistics ip-user-mapping** command.

**Meaning**

The counts of the queries and failed queries are displayed.

***Verifying IP Probe Counts***

**Purpose**

Verify that IP probes are occurring.

**Action**

From operational mode, enter the **show services user-identification active-directory-access statistics ip-user-probe** command.

**Meaning**

The counts of the IP probes and failed IP probes are displayed.

***Verifying User-to-Group Mapping Queries***

**Purpose**

Verify that user-to-group mappings are being queried.

**Action**

From operational mode, enter the **show services user-identification active-directory-access statistics user-group-mapping** command.

**Meaning**

The counts of the queries and failed queries are displayed.

SEE ALSO

*Understanding the Three-Tiered User Firewall Features*

policies | **555**

*show services user-identification active-directory-access active-directory-authentication-table*

show services user-identification active-directory-access domain-controller status | **792**

show services user-identification active-directory-access statistics | **795**

show services user-identification active-directory-access user-group-mapping | **799**

## Configuring Integrated User Firewall on NFX Devices

In a typical scenario for the integrated user firewall feature, domain users want to access the Internet through an NFX device. The device reads and analyzes the event log of the domain controllers configured in the domain. Thus, the device detects domain users on an Active Directory domain controller. Active Directory domain generates an authentication table as the Active Directory authentication source for the integrated user firewall. The device uses this information to enforce the policy to achieve user-based or group-based access control.

> **NOTE:** When a new user is created in Active Directory (AD), the user is added to the global security group Primary Group which is by default Domain Users. The Primary Group is less specific than other groups created in AD because all users belong to it. Also, it can become very large.
>
> You cannot use the Primary Group, whether by its default name of Domain Users or any other name, if you changed it, in integrated user firewall configurations.

To establish a Windows Active Directory domain and to configure another security policy:

1. Configure the LDAP base distinguished name.

```
[edit services user-identification]
user@host# set active-directory-access domain example.com user-group-mapping ldap base
    DC=example,DC=com
```

2. Configure a domain name, the username and password of the domain, and the name and IP address of the domain controller in the domain.

```
[edit services user-identification]
user@host# set active-directory-access domain example.com user administrator password $ABC123
user@host# set active-directory-access domain example.net domain-controller ad1 address
    2001:db8:0:1:2a0:a502:0:1da
```

3. Configure a second policy to enable a specific user.

```
[edit security policies from-zone trust to-zone untrust policy p2]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set match source-identity ""example.com\user1""
```

```
user@host# set then permit
```

**NOTE:** When you specify a source identity in a policies statement, prepend the domain name and a backslash to the group name or username. Enclose the combination in quotation marks.

4. Set the Active Directory authentication table as the authentication source for integrated user firewall information retrieval and specify the sequence in which user information tables are checked.

```
[edit security]
user@host# set user-identification authentication-source active-directory-authentication-table priority 125
```

**NOTE:** You must set the Active Directory authentication table as the authentication source for integrated user firewall information retrieval and specify the sequence in which user information tables are checked using the command **set security user-identification authentication-source active-directory-authentication-table priority *value***.

The default value of this option is 125. The default priority for all the authentication sources is as follows:

- Local authentication: 100
- Integrated user firewall: 125
- User role firewall: 150

The field **priority** specifies the sources for the Active Directory authentication table. The value set determines the sequence for searching among various supported authentication tables to retrieve a user role. Note that these are the only currently supported values. You can enter any value from 0 through 65,535. The default priority of the Active Directory authentication table is 125. This means that even if you do not specify a priority value, the Active Directory authentication table will be searched starting at sequence of value 125 (integrated user firewall).

For more details, see *Active Directory Authentication Tables* and **active-directory-authentication-table**.

To verify that the configuration is working properly:

1. Verify that at least one domain controller is configured and connected by entering the **show services user-identification active-directory-access domain-controller status** command.
2. Verify that the LDAP server is providing user-to-group mapping information by entering the **show services user-identification active-directory-access user-group-mapping status** command..
3. Verify the authentication table entries by entering the **show services user-identification active-directory-access active-directory-authentication-table all** command. The IP addresses, usernames, and groups are displayed for each domain.
4. Verifying IP-to-user mapping by entering the **show services user-identification active-directory-access statistics ip-user-mapping** command. The counts of the queries and failed queries are displayed.
5. Verify that IP probes are occurring by entering the **show services user-identification active-directory-access statistics ip-user-probe** command.
6. Verify that user-to-group mappings are being queried by entering the **show services user-identification active-directory-access statistics user-group-mapping** command.

SEE ALSO

*Understanding Integrated User Firewall Domain PC Probing*

## Example: Configuring Integrated User Firewall on SRX Series devices to Use Web-Redirect for Unauthenticated and Unknown Users

**IN THIS SECTION**


- Requirements | **212**
- Overview | **212**
- Configuration | **212**
- Verification | **215**


This example shows how to use web-redirect for unauthenticated users and unknown users to redirect to the authentication page through http.

## Requirements

This example uses the following hardware and software components:

- One SRX Series device
- Junos OS Release 15.1X49-D70 or later for SRX Series devices

No special configuration beyond device initialization is required before configuring this feature.

## Overview

The fwauth access profile redirects **web-redirect** requests of pass-through traffic to HTTP webauth (in JWEB httpd server). Once authentication is successful, fwauth creates a firewall authentication for the user firewall.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [**edit**] hierarchy level, and then enter **commit** from configuration mode.

```
set system services web-management http
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24 web-authentication http
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 match source-identity unauthenticated-user
set security policies from-zone trust to-zone untrust policy p1 match source-identity unknown-user
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    access-profile profile1 web-redirect
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    domain ad03.net
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the integrated user firewall to use web-redirect for unauthenticated users requesting access to HTTP-based resources:

1. Enable Web-management support for HTTP traffic.

```
[edit system services]
user@host# set system services web-management http
```

2. Configure interfaces and assign IP addresses. Enable Web authentication on ge-0/0/1 interface.

```
[edit interfaces]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24 web-authentication http
```

3. Configure security policies that specifies an unauthenticated-user or unknown-user as the source-identity.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set match source-identity unauthenticated-user
user@host# set match source-identity unknown-user
```

> **NOTE:** Starting with Junos OS 17.4R1, you can assign IPv6 addresses in addition to IPv4 addresses when you configure source addresses. To configure IPv6 source address, issue **any** or **any-IPv6** command at [edit security policies from-zone trust to-zone untrust policy policy-name match source-address] hierarchy level.

4. Configure a security policy that permits firewall authentication of a user firewall with **web-redirect** as the action and specifies a pre configured access profile for the user.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set then permit firewall-authentication user-firewall access-profile profile1 web-redirect
```

5. Configure a security policy that specifies the domain name.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set then permit firewall-authentication user-firewall domain ad03.net
```

**Results**

From configuration mode, confirm your configuration by entering the **show system services** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show system services
    web-management {
        http {
            port 123;
        }
    }
```

From configuration mode, confirm your integrated user-firewall configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 192.0.2.0/24 {
                    web-authentication http;
                }
                }
            }
        }
```

From configuration mode, confirm your integrated user-firewall configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security policies
    from-zone trust to-zone untrust {
        policy p1 {
            match {
                source-address any;
                destination-address any;
                application any;
                source-identity unauthenticated-user;
                source-identity unknown-user;
            }
            then {
                permit {
                    firewall-authentication {
```

```
                user-firewall {
                    access-profile profile1;
                    web-redirect;
                    domain ad03.net;
                }
            }
        }
    }
}
```

From configuration mode, confirm your policy configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

***Verify the Configuration.***

**Purpose**

Verify that the configuration is correct.

**Action**

From operational mode, enter the **show security policies** command.

## Sample Output

```
user@host> show security policies

    Default policy: permit-all

    From zone: PCzone, To zone: Tunnelzone
        Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
            Source addresses: any

                Destination addresses: any

                Applications: junos-ftp, junos-tftp, junos-dns-tcp, junos-dns-udp
```

```
Action: permit
```

**Meaning**

Display the security policy that permits firewall authentication of a user firewall with web-redirect as the action.

SEE ALSO

Overview of Integrated User Firewall | 177

Example: Configuring Integrated User Firewall on SRX Series | 197

## Example: Configuring Integrated User Firewall on SRX Series devices to Use Web-Redirect-to-HTTPS to Authenticate Unauthenticated and Unknown Users

IN THIS SECTION

- Requirements | 216
- Overview | 217
- Configuration | 218

This example shows how to use web-redirect-to-https for unauthenticated and unknown users attempting to access an HTTPS site to enable them to authenticate through the SRX Series device's internal webauth server.

You can also use web-redirect-https to authenticate users attempting to access an HTTP site, although not shown in this example.

### Requirements

This example uses the following hardware and software components:

- One SRX Series device
- Junos OS Release 15.1X49-D70 or later for SRX Series devices

## Overview

The web-redirect-https feature allows you to securely authenticate unknown and unauthenticated users attempting to access either HTTP or HTTPS resources by redirecting the user's browser to the SRX Series services gateway's internal HTTPS webauth server for authentication. That is, the webauth server sends an HTTPS response to the client system redirecting its browser to connect to the webauth server for user authentication. The interface on which the client's request arrives is the interface to which the redirect response is sent. HTTPS, in this case, secures the authentication process, not the user's traffic.

After the user has been authenticated, a message is displayed to inform the user about the successful authentication. The browser is redirected to launch the user's original destination URL, whether to an HTTP or HTTPS site, without requiring the user to retype that URL. The following message is displayed:

```
Redirecting to the original url, please wait.
```

**NOTE:** If the user's target resource is to an HTTPS URL, for this process to succeed the configuration must include an SSL termination profile that is referenced in the applicable security policy. An SSL termination profile is not required if the target is an HTTP URL.

Use of this feature allows for a richer user login experience. For example, instead of a pop-up prompt asking the user to enter their user name and password, users are presented with the login page in a browser. Use of web-redirect-https has the same effect as if the user typed the Web authentication IP address in a client browser. In that sense, web-redirect-https provides a seamless authentication experience; the user does not need to know the IP address of the Web authentication source, but only the IP address of the resource that they are attempting to access.

For integrated user firewall, the security policy configuration statement includes the source-identity tuple, which allows you to specify a category of users to whom the security policy applies, in this case unauthenticated and unknown users. Specifying "any" as the value of the source-address tuple allows the source-identity tuple value to control the match.

**NOTE:** For security reasons, it is recommended that you use the web-redirect-https for authentication instead of web-redirect, which is also supported. The web-redirect authentication feature uses HTTP for the authentication process, in which case the authentication information is sent in the clear and is therefore readable.

This example assumes that the user is attempting to access an HTTPS resource such as https://mymailsite.com.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [**edit**] hierarchy level, and then enter **commit** from configuration mode.

```
set system services web-management https pki-local-certificate my-test-cert
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24 web-authentication https
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 match source-identity unauthenticated-user
set security policies from-zone trust to-zone untrust policy p1 match source-identity unknown-user
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    domain mydomain.net
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    access-profile profile1 web-redirect-to-https
set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication user-firewall
    ssl-termination-profile my-ssl-profile
set services ssl termination profile my-ssl-profile server-certificate my-test-cert
set access profile profile1 ldap-server 198.51.100.0/24 tls-type start-tls
set access profile profile1 ldap-server 198.51.100.0/24 tls-peer-name peer1
set access profile profile1 ldap-server 198.51.100.0/24 tls-timeout 3
set access profile profile1 ldap-server 198.51.100.0/24 tls-min-version v1.1
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure web-redirect-to-https for unauthenticated users or unknown users requesting access to HTTPS-based resources, enter the following statement.

1. Enable Web-management support for HTTPS traffic.

   ```
   [edit system services]
   user@host# set system services web-management https pki-local-certificate my-test-cert
   ```

   Note that this example applies to HTTPS user traffic, but web-redirect-to-https authentication is also supported for authenticated users whose traffic is to an HTTP URL site, although that specific scenario is not shown here. In that case, an SSL termination profile is not required.

2. Configure interfaces and assign IP addresses. Enable Web authentication on ge-0/0/1 interface.

```
[edit interfaces]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24 web-authentication https
```

3. Configure a security policy that specifies unauthenticated-user and unknown-user as the source-identity tuple values.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set match source-identity unauthenticated-user
user@host# set match source-identity unknown-user
```

> **NOTE:** Starting with Junos OS 17.4R1, you can assign IPv6 addresses in addition to IPv4 addresses when you configure source addresses. To configure IPv6 source address, issue **any** or **any-IPv6** command at the [edit security policies from-zone trust to-zone untrust policy policy-name match source-address] hierarchy level.

4. Configure the security policy to permit firewall authentication of a user firewall with **web-redirect-to-https** as the action and that specifies a preconfigured access profile for the user.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set then permit firewall-authentication user-firewall access-profile profile1 web-redirect-to-https
```

5. Configure the domain name for the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set then permit firewall-authentication user-firewall domain mydomain.net
```

6. Configure the security policy to reference the SSL termination profile to be used.

> **NOTE:** If you have an existing appropriate SSL termination profile that provides the services needed for your implementation, you can use it. Otherwise, follow Step 7 to create one.

```
[edit security policies from-zone trust to-zone untrust policy p1]
user@host# set then permit firewall-authentication user-firewall ssl-termination-profile my-ssl-profile
```

7. Specify the profile to be used for SSL termination services.

```
[edit services]
user@host# set ssl termination profile my-ssl-profile server-certificate my-cert-type
```

8. Define the TLS type to configure the LDAP over StartTLS.

```
[edit access]
user@host# set profile profile1 ldap-server 198.51.100.0/24 tls-type start-tls
```

9. Configure the peer host name to be authenticated.

```
[edit access]
user@host# set access profile profile1 ldap-server 198.51.100.0/24 tls-peer-name peer1
```

10. Specify the timeout value on the TLS handshake. You can enter 3 through 90 seconds.

```
[edit access]
user@host# set access profile profile1 ldap-server 198.51.100.0/24 tls-timeout 3
```

11. Specify TLS version (v1.1 and v1.2 are supported) as the minimum protocol version enabled in connections.

```
[edit ]
user@host# set access profile profile1 ldap-server 198.51.100.0/24 tls-min-version v1.1
```

**Results**

From configuration mode, confirm your configuration by entering the **show system services** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show system services
    web-management {
        https {
```

```
        pki-local-certificate my-test-cert;
    }
```

From configuration mode, confirm your integrated user-firewall configuration by entering the **show services ssl** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show services ssl
    termination {
        profile my-ssl-profile {
            server-certificate my-cert-type;
                }
            }
```

From configuration mode, confirm your integrated user-firewall configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 192.0.2.0/24 {
                    web-authentication {
                        https;
                    }
                }
            }
        }
```

From configuration mode, confirm your integrated user-firewall configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security policies
    from-zone trust to-zone untrust {
        policy p1 {
            match {
                source-address any;
                destination-address any;
                application any;
                source-identity unauthenticated-user;
```

```
            source-identity unknown-user;
        }
        then {
            permit {
                firewall-authentication {
                    user-firewall {
                        access-profile profile1;
                        web-redirect-to-https;
                        domain mydomain.net;
                        ssl-termination-profile my-ssl-profile;
                    }
                }
            }
        }
    }
```

From configuration mode, confirm your access profile configuration by entering the **show access profile profile1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access profile profile1
    ldap-server {
        198.51.100.0/24 {
            tls-type start-tls;
            tls-timeout 3;
            tls-min-version v1.1;
            tls-peer-name peer1;
        }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

SEE ALSO

Example: Configuring Integrated User Firewall on SRX Series | 197

LDAP Functionality in Integrated User Firewall | 194

# Configure Captive Portal for Unauthenticated Browsers

IN THIS SECTION


- Understanding SRX Series Assured Captive Portal Support for Unauthenticated Browser Users | 223
- Understanding the Forced Timeout Setting Assigned to Active Directory Authentication Entries for Users Authenticated Through Captive Portal | 226


Generally, an SRX Series device redirects an unauthenticated user to the captive portal for authentication. While redirecting to the captive portal, the background process such as Microsoft updates triggers the captive portal before it triggers HTTP/HTTPS browser-based user's access, which makes the browser to display "401 Unauthorized" page without presenting authentication portal. The auth-only-browser and auth-user-agent parameters give you control to handle HTTP/HTTPS traffic.

## Understanding SRX Series Assured Captive Portal Support for Unauthenticated Browser Users

When an unauthenticated user requests access to an SRX Series protected resource using an HTTP/HTTPS browser, the SRX Series device presents the user with a captive portal interface to allow the user to authenticate. Normally, this process occurs without interference. However, prior to introduction of this feature, HTTP/HTTPS-based workstation services running in the background, such as Microsoft updates and control checks, could trigger captive portal authentication before the HTTP/HTTPS browser-based user's access request did. The situation posed a race condition. If a background process triggered captive portal first, the SRX Series device presented it with a "401 Unauthorized" page. The service discarded the page without informing the browser, and the browser user was never presented with the authentication portal. The SRX Series device did not support simultaneous authentication from the same source (IP address) on different SPUs.

Starting with Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1, the SRX Series device supports simultaneous HTTP/HTTPS pass-through authentication across multiple SPUs, including support for web-redirect authentication. If an HTTP/HTTPS packet arrives while the SPU is querying the CP, the SRX Series device queues the packet to be handled later.

Additionally, the following two parameters are made available to give you greater control over how HTTP/HTTPS traffic is handled.

- auth-only-browser—Authenticate only browser traffic. If you specify this parameter, the SRX Series device distinguishes HTTP/HTTPS browser traffic from other HTTP/HTTPS traffic. The SRX Series device does not respond to non-browser traffic. You can use the auth-user-agent parameter in conjunction with this control to further ensure that the HTTP traffic is from a browser.
- auth-user-agent—Authenticate HTTP/HTTPS traffic based on the User-Agent field in the HTTP/HTTPS browser header. You can specify one user-agent value per configuration. The SRX Series device checks the user-agent value that you specify against the User-Agent field in the HTTP/HTTPS browser header for a match to determine if the traffic is HTTP/HTTPS browser-based.

  You can use this parameter with the auth-only-browser parameter or alone for both pass-through and user-firewall firewall-authentication.

  You can specify only one string as a value for auth-user-agent. It must not include spaces and you do not need to enclose the string in quotation marks.

> **NOTE:** Starting with Junos OS 17.4R1, you can assign IPv6 addresses in addition to IPv4 addresses when you configure source addresses. To configure IPv6 source address, issue **any** or **any-IPv6** command at [edit security policies from-zone trust to-zone untrust policy policy-name match source-address] hierarchy level.

Here are some examples of how to configure security policies to use the auth-only-browser and auth-user-agent firewall authentication features.

**For Pass-Through Authentication**

Configures a security policy for pass-through authentication that uses the auth-only-browser parameter.

```
user@host# set security policies from-zone trust to-zone untrust policy p1 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p1 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p1 match application any
user@host# set security policies from-zone trust to-zone untrust policy p1 then permit firewall-authentication
    pass-through auth-only-browser access-profile my-access-profile1t
```

Configures a security policy for pass-through authentication that uses the auth-user-agent parameter without auth-only-browser.

```
user@host# set security policies from-zone trust to-zone untrust policy p2 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p2 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p2 match application any
```

```
user@host# set security policies from-zone trust to-zone untrust policy p2 then permit firewall-authentication
    pass-through auth-user-agent Opera1 access-profile my-access-profile2
```

Configures a security policy for pass-through authentication that uses the auth-only-browser with the auth-user-agent parameter.

```
user@host# set security policies from-zone trust to-zone untrust policy p3 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p3 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p3 match application any
user@host# set security policies from-zone trust to-zone untrust policy p3 then permit firewall-authentication
    pass-through auth-only-browser auth-user-agent Opera1 my-access-profile3
```

**For User Firewall Authentication**

Configures a security policy for user-firewall authentication that uses the auth-only-browser parameter.

```
user@host# set security policies from-zone trust to-zone untrust policy p4 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p4 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p4 match application any
user@host# set security policies from-zone trust to-zone untrust policy p4 then permit firewall-authentication
    user-firewall auth-only-browser access-profile my-access-profile4t
```

Configures a security policy for user-firewall authentication that uses the auth-user-agent parameter without auth-only-browser.

```
user@host# set security policies from-zone trust to-zone untrust policy p5 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p5 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p5 match application any
user@host# set security policies from-zone trust to-zone untrust policy p5 then permit firewall-authentication
    user-firewall auth-user-agent Opera1 access-profile my-access-profile5
```

Configures a security policy for user-firewall authentication that uses the auth-only-browser with the auth-user-agent parameter.

```
user@host# set security policies from-zone trust to-zone untrust policy p6 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p6 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p6 match application any
user@host# set security policies from-zone trust to-zone untrust policy p6 then permit firewall-authentication
    user-firewall auth-only-browser auth-user-agent Opera1 access-profile my-access-profile6
```

SEE ALSO

auth-only-browser | 416
auth-user-agent | 418

## Understanding the Forced Timeout Setting Assigned to Active Directory Authentication Entries for Users Authenticated Through Captive Portal

This topic covers the effect of the firewall authentication forced timeout setting as it applies to active directory authentication entries for users who authenticate through captive portal.

When a user authenticates through captive portal, an authentication table entry is generated for that user based on the information that the SRX Series device obtains from the firewall authentication module. At that point, the default traffic-based authentication timeout logic is applied to the entry.

As an administrator, it is important for you to have control over how long non-domain users who authenticate through captive portal remain authenticated. The firewall authentication forced timeout feature gives you that control. Use of it ensures that non-domain users do not remain authenticated indefinitely. For example, assume that the flow of traffic is continuous to and from the device of a non-domain user authenticated through captive portal. Given the behavior of the default traffic-based authentication timeout, the non-domain user would remain authenticated indefinitely.

When the firewall authentication forced timeout value is configured, it is used in conjunction with the traffic-based timeout logic.

Here is how timeout settings, including firewall authentication forced timeout, affect active directory authentication entries for users authenticated through captive portal. In all of the following instances, an authentication entry was generated for a user based on firewall authentication information after the user authenticated through captive portal.

- The firewall authentication forced timeout is set for 3 hours.

  Traffic continues to be received and generated by a device associated with an authentication entry for a user. After 3 hours the authentication entry expires, although at that time there are sessions anchored in Packet Forwarding Engine for the authentication entry.

- If set, the firewall authentication forced timeout has no effect.

  An authentication entry does not have sessions anchored to it. It expires after the time set for the authentication entry timeout, for example, 30 minutes.

- The firewall authentication forced timeout configuration is deleted.

Firewall authentication forced timeout has no effect on new authentication entries. Firewall authentication forced timeout remains enforced for existing authentication entries to which it applied before it was deleted. That is, for those authentication entries, the original forced timeout setting remains in effect.

- The firewall authentication forced timeout configuration setting is changed.

  The new tine-out setting is applied to new incoming authentication entries. Existing entries keep the original, former setting.

- The firewall authentication forced timeout is set to 0, disabling it.

  If the firewall authentication forced timeout is set to a new value, that value is assigned to all incoming authentication entries. There is no firewall authentication forced timeout setting for existing authentication entries.

- The firewall authentication forced timeout value is not configured.
  - The SRX Series device generates an authentication entry for a user. The default traffic-based timeout logic is applied to the authentication entry.
  - The active directory timeout value is configured for 50 minutes. A traffic-based timeout of 50 minutes is applied to an authentication entry.
  - The active directory timeout is not configured. The default traffic-based timeout of 30 minutes is applied to an authentication entry.

SEE ALSO

firewall-authentication-forced-timeout | 498

invalid-authentication-entry-timeout | 526

Understanding the Invalid Authentication Table Entry Timeout Setting | 190

# Manage Event logs to Generate IP Address-to-User Mapping

IN THIS SECTION

Understanding How the WMIC Reads the Event Log on the Domain Controller | 228

Using Firewall Authentication as an Alternative to WMIC | 230

Understanding Integrated User Firewall Domain PC Probing | 231

SRX Series and NFX Series device gathers IP address, user, and group information from Windows Active Directory domain controller event logs and adds to the active directory authentication table. Authentication entries become a source for authentication.

# Understanding How the WMIC Reads the Event Log on the Domain Controller

**IN THIS SECTION**


- Windows Management Instrumentation Client | **228**
- WMIC Reads the Event Log on the Domain Controller | **229**
- Specifying IP Filters to Limit IP-to-User Mapping | **229**
- Event Log Verification and Statistics | **230**


This topic includes the following sections:

## Windows Management Instrumentation Client

When you configure the integrated user firewall feature on a device, the device establishes a Windows Management Instrumentation (WMI)/Distributed Component Object Module (DCOM) connection to the domain controller. The device acts as a WMI client (WMIC). It reads and monitors the security event log on the domain controller. The device analyzes the event messages to generate IP address-to-user mapping information.

All configuration regarding the WMIC is optional; it will function with default values. After the domain is configured (by the **set services user-identification active-directory-access domain** statement), the WMIC starts to work. The WMIC connection to the domain controller uses the same user credentials as those configured for the domain.

**CAUTION:** Integrated user firewall uses NTLMv2 as the default WMIC authentication protocol for security reasons. NTLMv1 exposes the system to attacks in which authentication hashes could be extracted from NTLMv1 authentication responses.

For compatibility with integrated user firewall, you must apply the latest version of the Microsoft SP2 patch if you are running an older version of Windows OS, including Windows 2000, Windows XP, and Windows 2003.

### WMIC Reads the Event Log on the Domain Controller

The following behaviors apply to reading the event log:

- The device monitors the event log at a configurable interval, which defaults to 10 seconds.
- The device reads the event log for a certain timespan, which you can configure. The default timespan is one hour. Each time at WMIC startup, the device checks the last timestamp and the timespan. If the last timestamp is older than the current timespan, then the timespan takes effect. After the WMIC and the UserID process start working, the timespan does not apply; the device simply reads the latest event log.
- The device can read the event log to obtain IPv6 addresses in addition to IPv4 addresses.
- During WMIC startup, the device has a maximum count of events it will read from the event log, and that maximum is not configurable.
  - On SRX300, SRX320, SRX340, and SRX345 devices, the maximum count is 100,000.
  - On SRX5400, SRX5600, and SRX5800 devices, the maximum count is 200,000.

  During WMIC startup, this maximum count is used with the timespan setting, so that if either limit is reached, the WMIC stops reading the event log.
- After a failover, the device reads the event log from the latest event log timestamp.
- In a chassis cluster environment, the WMIC works on the primary node only.

### Specifying IP Filters to Limit IP-to-User Mapping

You can specify IP filters to limit the IP address-to-user mapping information that the device generates from the event log.

To understand when a filter is useful for such mapping, consider the following scenario. A customer deploys 10 devices in one domain, and each device controls a branch. All 10 devices read all 10 branch user login event logs in the domain controller. However, the device is configured to detect only whether the user is

authenticated on the branch it controls. By configuring an IP filter on the device, the device reads only the IP event log under its control.

You can configure a filter to include or exclude IP addresses or prefixes. You can specify a maximum of 20 addresses for each filter.

### Event Log Verification and Statistics

You can verify that the authentication table is getting IP address and user information by issuing the **show services user-identification active-directory-access active-directory-authentication-table all** command. A list of IP address-to-user mappings is displayed for each domain. The table contains no group information until LDAP is running.

You can see statistics about reading the event log by issuing the **show services user-identification active-directory-access ip-user-mapping statistics domain** command.

SEE ALSO

> *show services user-identification active-directory-access active-directory-authentication-table*

## Using Firewall Authentication as an Alternative to WMIC

**IN THIS SECTION**


- WMIC Limitations | **230**
- Firewall Authentication as a Backup Method for IP Address-to-User Mappings | **231**


This topic includes the following sections:

### WMIC Limitations

The primary method for the integrated user firewall feature to get IP address-to-user mapping information is for the device to act as a WMI client (WMIC). However, the WMIC has limitations, such as the following:

- On Windows XP or Server2003, the Windows firewall does not allow the WMIC request to pass through because of the dynamic port allocation of the Distributed Component Object Model (DCOM). Therefore,

for these operating systems when Windows firewall is enabled, the PC does not respond to the WMIC probe.

- Because the event-log-reading and PC probe functions both use WMI, using a global policy to disable the WMI-to-PC probe also affects event log reading.

Because these cases might result in the failure of the PC probe, a backup method for getting IP address-to-user mappings is needed. That method is to use firewall authentication to identify users.

### Firewall Authentication as a Backup Method for IP Address-to-User Mappings

If you want to use firewall authentication to identify users for the integrated user firewall feature, specify a domain name in the **set security policies from-zone trust to-zone untrust policy <policy-name> then permit firewall-authentication user-firewall domain <domain-name>** statement.

If a domain is configured in that statement, fwauth recognizes that the domain is for a domain authentication entry, and will send the domain name to the fwauth process along with the authentication request. After it receives the authentication response, fwauth deletes that domain authentication entry. The fwauth process sends the source IP address, username, domain, and other information to the USERID process, which verifies that it is a valid domain user entry. The subsequent traffic will hit this user firewall entry.

> **NOTE:** The Active Directory authentication entry that comes from the fwauth process is not subject to the IP filters.

SEE ALSO

*user-firewall*

## Understanding Integrated User Firewall Domain PC Probing

**IN THIS SECTION**


- Overview of Domain PC Probing | **232**
- Probing Domain PCs for User Information | **232**
- Probe Response | **233**

Probe Configuration | 234
Probe Rate and Statistics | 234

This topic includes the following sections:

## Overview of Domain PC Probing

At a high level, the integrated user firewall feature gathers IP address, user, and group information from Windows Active Directory domain controller event logs and LDAP services. This information is used to generate Active Directory authentication table entries on a device. Authentication entries serve as the authentication source for security policies that enforce user-based or group-based access control.

PC probing acts as a supplement of event log reading. When a user logs in to the domain, the event log contains that information. The PC probe is triggered only when there is no IP-to-address mapping from the event log.

Domain information constantly changes as users log in and out of domain PCs. The integrated user firewall probe functionality provides a mechanism for tracking and verifying information in the authentication tables by directly probing domain PCs for IP address-to-user mapping information. New and changed information identified by the probe serves to update Active Directory authentication table entries, which is critical to maintaining firewall integrity.

The IP address filter also impacts the PC probe. Once you configure the IP address filter, only the IP address specified in the filter is probed.

## Probing Domain PCs for User Information

The integrated user firewall feature tracks the online status of users by probing domain PCs. If a user is not online or is not an expected user, the Active Directory authentication table is updated as appropriate. The following probe behaviors apply:

**On-demand probing**—On-demand probing occurs when a packet is dropped due to a missing entry in the Active Directory authentication table. In this case, an entry is added in pending state to the authentication table, and the domain PC identified by the source IP field of the dropped packet is probed for IP address and user information. The entry remains in pending state until a response is received from the probe.

**Manual probing**—Manual probing is used to verify and troubleshoot the online status of a user or a range of users, and is at the discretion of the system administrator. To initiate a manual probe, use the **request services user-identification active-directory-access ip-user-probe address ip-address *address* domain**

***domain-name*** command. If a domain name is not specified, the probe looks at the first configured domain for the IP address. To specify a range, use the appropriate network address.

**NOTE:** Manual probing can cause entries to be removed from the Active Directory authentication table. For example, if there is no response from your PC due to a network issue, such as when the PC is too busy, the IP address entry of the PC is marked as *invalid* and your access is blocked.

If the device cannot access a domain PC for some reason, such as a network configuration or Windows firewall issue, the probe fails.

### Probe Response

Based on the domain PC probe response, updates are made to the Active Directory authentication table, and associated firewall policies take effect. If no response is received from the probe after 90 seconds, the authentication entry times out. The timed-out authentication entry is the pending state authentication entry, which is generated when you start the PC probe.

If the probe is successful, the state of the authentication entry is updated from pending to valid. If the probe is unsuccessful, the state of the authentication entry is marked as invalid. The invalid entry has the same lifetime as a valid entry and is overwritten by upcoming fwauth (firewall authentication process) authentication results or by the event log. Table 10 on page 233 lists probe responses and corresponding authentication table actions.

**Table 10: Probe Responses and Associated Active Directory Authentication Table Actions**

| Probe Response from Domain PC | Active Directory Authentication Table Action |
|---|---|
| Valid IP address and username | Add IP-related entry. |
| Logged on user changed | Update IP-related entry. |
| Connection timeout | Update IP-related entry as invalid. |
| Access denied | Update IP-related entry as invalid. |
| Connection refused | Update IP-related entry as invalid. |
| Authentication failed<br><br>(The configured username and password have no privilege to probe the domain PC.) | Update IP-related entry as invalid. |

### Probe Configuration

On-demand probing is enabled by default. To disable on-demand probing, use the **set services user-identification active-directory-access no-on-demand-probe** statement. Delete this statement to reenable probing. When on-demand probing is disabled, manual probing is available.

The probe timeout value is configurable. The default timeout is 10 seconds. To configure the timeout value, use the following statement:

```
user@host# set services user-identification active-directory-access wmi-timeout seconds
```

If no response is received from the domain PC within the **wmi-timeout** interval, the probe fails and the system either creates an invalid authentication entry or updates the existing authentication entry as invalid. If an authentication table entry already exists for the probed IP address, and no response is received from the domain PC within the **wmi-timeout** interval, the probe fails and that entry is deleted from the table.

> **NOTE:** To probe domain PCs, you must configure the integrated user firewall feature with the username and password credentials. You do not necessarily need a username and password account for each PC; instead you could set up one administrator account with privileges to access information on multiple PCs.

### Probe Rate and Statistics

The maximum probe rate for the integrated user firewall feature is set by default and cannot be changed. For SRX 5400, SRX 5600, and SRX 5800 devices, the probe rate is 600 times per minute. For branch SRX Series devices, the probe rate is 100 times per minute. Probe functionality supports 5000 users, or up to 10 percent of the total supported authentication entries, whichever is smaller. Supporting 10 percent means that at any time, the number of IP addresses waiting to be probed cannot exceed 10 percent. For more information about the number of supported Active Directory authentication table entries, see “Understanding Active Directory Authentication Tables” on page 182.

High-level statistics covering probe activity are available for the total number of probes and the number of failed probes. Table 10 on page 233 describes the reasons for probe failures. To display probe statistics, use the **show services user-identification active-directory-access statistics ip-user-probe** command.

```
user@host> show services user-identification active-directory-access statistics ip-user-probe
Domain: www.example1.net
    Total user probe number                       : 176116
    Failed user probe number                      : 916
```

```
Domain: www.example2.net
    Total user probe number                         : 17632
    Failed user probe number                        : 342
```

# Logging User Identity Information Based on Zones

**IN THIS SECTION**


- Understanding How to Include User Identity Information in the Session Log File Based on the Source Zone | **235**
- Example: Configuring Integrated User Firewall to Write User Identity to the Session Log Based On the Source Zone | **237**


The integrated user firewall zone-based feature directs the system to log the user identify information based on the source zone configured in the security policy. The log information includes all users who belong to the zone and their traffic matches the security policy.

## Understanding How to Include User Identity Information in the Session Log File Based on the Source Zone

This topic covers the integrated user firewall feature that allows you to configure the system to write to the session log the user's identity by user name or group name without having to use the source identity (source-identity) tuple in the security policy. Knowing the user's identity by name, as written to the log, not just by the IP address of the user's device, gives you clearer visibility into their activity and allows you to resolve security problems faster and more easily. Relying on the source zone (from-zone) to trigger user identity logging rather than on the source identity widens the scope of users whose source identity is logged.

Typically, for each security policy, you must specify in the policy the source and destination IP addresses and the zones against which traffic is matched. You must also specify an application that the traffic is matched to. If traffic matches these criteria, then the security policy's action is applied to the traffic issued from the user's device. However, no user identity information is written to the session log.

Optionally, instead of relying exclusively on the IP address of the user's device to identify the source of the traffic, you can specify the user identity—that is, the user name or the group name—in the source-identity tuple of a security policy. This approach gives you greater control over resource access by narrowing down application of the security policy's actions to a single, identified user or a group of users, if other security policy matching conditions are met. However, use of the source-identity tuple constrains application of the policy to traffic from a single user or user group.

It may happen that you want the system to write to the session log the user identity for all users from whom traffic originated based on the zone to which they belong (from-zone). In this case, you do not want to narrow the traffic match and security policy application to a single user or a user group, which configuring the source-identity tuple would do.

The zone-based user identity feature allows you to direct the system to write to the log user identity information for any user who belongs to a zone that is configured with the source-identity-log statement when that zone is used as the source zone in a matching security policy.

> **NOTE:** For the source-identity-log feature to take effect, you must also configure logging of the session initialize (session-init) and session end (session-close) events as part of the security policy's actions.

Table 11 on page 236 identifies the platforms that support this feature.

**Table 11: Supported Platforms**

| Supported SRX Series Device Platforms |
| --- |
| SRX320 |
| SRX550M |
| SRX1500 series |
| |
| SRX500 series |

# Example: Configuring Integrated User Firewall to Write User Identity to the Session Log Based On the Source Zone

IN THIS SECTION


- Requirements | 237
- Overview | 237
- Configuration | 239
- Verification | 240


This example shows how to configure the integrated user firewall zone-based user identity feature that directs the system to log user identity information based on the source zone (from-zone) configured in the security policy. The zone-based user identity feature widens the scope of users whose identity information is written to the log to include all users who belong to the zone whose traffic matches the security policy.

## Requirements

This feature is supported starting with Junos OS 15.1X49-D60 and Junos OS Release 17.3R1. You can configure and run this feature on any of the currently supported SRX Series devices beginning with Junos OS 15.1X49-D60.

## Overview

This example shows how to configure integrated user firewall to log user identity information in the session log based on the source zone in the security policy. For this to occur, the zone specified as the source zone must be configured for source identity logging. For zone-based user identity logging, the security policy's actions must include session create (session-init) and session close (session-init) events.

When all conditions are met, the user's name is written to the log at the beginning of the session (or session initialization) and at the beginning of the close of the session (or session tear-down). Note that if a security policy denies the user access to the resource, an entry identifying the user by name is written to the log, that is, if session close is configured.

When you use the zone-based user identity feature, it is the source zone (from-zone) in the security policy that initiates the user identity logging event.

Prior to introduction of this feature, it was necessary to include the source identity tuple (source-identity) in a security policy to direct the system to write user identity information to the log—that is, the user name

or the group name. The user identity was written to the log if the source-identity tuple was configured in any of the policies in a zone pair that matched the user's traffic and the session close log was configured.

However, the source identity feature is specific to an individual user or a group of users, and it constrains application of the security policy in that regard.

It is the user name that is stored in the local Active Directory table which the system writes to the log when the policy's source zone is configured for user-identity logging. The SRX Series device previously obtained the user identity information by reading the domain controller event log. The SRX Series device stored that information in its Active Directory table.

You can use the source-identity tuple in a security policy that also specifies as its source zone a zone that was configured for user identity logging. Because integrated user firewall collects the names of the groups that a user belongs to from Microsoft Domain Controllers only when integrated user firewall relies on the source identity tuple, if you use the zone-based user identity logging feature without also configuring source-identity, the log will contain only the name of the user requesting access and not the groups that the user belongs to.

After you configure a zone to support source identity logging, the zone is reusable as the from-zone specification in any security policy for which you want user identity information logged.

To summarize, the user's name is written to the log if:

- The user belongs to the zone configured for source identity logging.
- The user Issues a resource access request whose generated traffic matches a security policy whose source zone (from-zone) tuple specifies a qualifying zone.
- The security policy includes as part of its actions the session initialize (session-init) and session end (session-close) events.

The source identity log function benefits include the ability to:

- Cover a wide range of users in a single specification—that is, all users who belong to a zone that is configured for source identity logging.
- Continue to use an address range for the source address in a security policy without forfeiting user identity logging.
- Reuse a zone that is configured for source identity logging in more than one security policy.

  Because it is configured independent of the security policy, you can specify the zone as the source zone in one or more policies.

> **NOTE:** The user identity is not logged if you specify a zone configured for zone-based user identity logging as the destination zone rather than as the source zone.

For this function to work, you must configure the following information:

- The source identity log statement configured for a zone that is used as the source zone (from-zone) in the intended security policy.
- A security policy that specifies:
  - A qualifying zone as its source zone.
  - The session-init and the session-close events as part of its actions.

## Configuration

**IN THIS SECTION**


- Configuring a Zone to Support Source Identity Logging and Using It in a Security Policy | **239**
- Results | **240**


To configure the source identity logging feature, perform these tasks:

### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security zones security-zone trust source-identity-log
set security policies from-zone trust to-zone untrust policy appfw-policy1 match source-address any
    destination-address any application junos-ftp
set security policies from-zone trust to-zone untrust policy appfw-policy1 then permit
set security policies from-zone trust to-zone untrust policy appfw-policy1 then log session-init
set security policies from-zone trust to-zone untrust policy appfw-policy1 then log session-close
```

### *Configuring a Zone to Support Source Identity Logging and Using It in a Security Policy*

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Configure source identity logging for the trust zone. When this zone is used as the source zone in a security policy, the system writes the user identity information to the session log for all users to whom the security policy applies.

```
[edit security]
user@host# set zones security-zone trust source-identity-log
```

2. Configure a security policy called appfw-policy1 that specifies the zone trust as the term for its source zone. Source identity logging is applied to any user whose traffic matches the security policy's tuples.

   This security policy allows the user to access the junos-ftp service. When the session is established for the user, the user's identity is logged. It is also logged at the close of the session.

```
[edit security]
user@host# set policies from-zone trust to-zone untrust policy appfw-policy1 match source-address any
    destination-address any application junos-ftp
user@host#  set policies from-zone trust to-zone untrust policy appfw-policy1 then permit
```

3. Configure the appfw-policy1 security policy's actions to include logging of the session initiation and session close events.

   **NOTE:** You must configure the security policy to log session initiation and session close events for the source identity log function to take effect. The user identity information is written to the log in conjunction with these events.

```
[edit security]
user@host# set policies from-zone trust to-zone untrust policy appfw-policy1 then log session-init
user@host# set policies from-zone trust to-zone untrust policy appfw-policy1 then log session-close
```

***Results***

From configuration mode, confirm your configuration by entering the **show security zones** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

## Verification

**IN THIS SECTION**


- Verify that the User Identity Information Was Logged | **242**

This section shows the session log generated for the user session. The log output:

- Shows the user name, user1, which appears at the outset of session open and then again at the outset of session close.

  The security policy configuration that caused the user name to be written to the log specifies the zone trust as its source zone. The zone trust was configured for source identity logging.
- Includes information obtained from the user's request traffic, the policy matching criteria, and the NAT setup.
- Contains identity information about the user, which is obtained from the Active Directory database. That information includes the role parameter for "MyCompany/Administrator", which shows the groups that the user belongs to.

In this scenario, the user requested access to the Juniper Networks junos-ftp service, which the log also records. Table 12 on page 241 calls out the parts of the log that are specific to the source identity log function configuration:

**Table 12: Session Log Components Specific to the Source Identity Log Function**

| | |
|---|---|
| session create<br><br>This is the session initiation which begins the first section of the log that records the session setup information.<br><br>The user's name, user1, is displayed at the beginning of the session create log recording. | user1 RT_FLOW_SESSION_CREATE |
| Session create is followed by standard information that defines the session based on the user's traffic that matches security policy tuples. | |
| source address, the source port, the destination address, the destination port. | source-address="198.51.100.13/24"<br>source-port="635"<br>destination-address="198.51.100.10/24"<br>destination-port="51" |
| application service<br><br>This is the application service that the user requested access to and which the security policy permitted. | service-name="junos-ftp" |
| source zone, destination zone<br><br>Further down the log are the zone specifications which show trust as the source zone and untrust as the destination zone as defined. | source-zone-name="trust"<br>destination-zone-name="untrust" |

**Table 12: Session Log Components Specific to the Source Identity Log Function *(continued)***

| | |
|---|---|
| session close<br><br>This is the session close initiation, which begins the second part of the log record that covers session tear-down and close.<br><br>The user's name, user1, is displayed at the beginning of the session close record. | user1 RT_FLOW - RT_FLOW_SESSION_CLOSE |

***Verify that the User Identity Information Was Logged***

**Purpose**

Note that integrated user firewall collects groups configured as the source-identity only from Microsoft Domain Controllers. If you use the zone-based user-identity feature without configuring source-identity, the log will contain only the user's name, that is, no group informations is recorded. In that case, the "roles=" section of the log will show "N/A". In the following example, it is assumed that the source-identity tuple was used and the "roles=" section shows a long list of the groups that the user "Administrator" belongs to.

**Action**

Display the log information.

## Sample Output

```
<14>1 2015-01-19T15:03:40.482+08:00 user1 RT_FLOW - RT_FLOW_SESSION_CREATE
[user@host2636 192.0.2.123 source-address=”198.51.100.13“ source-port=”635”
destination-address=”198.51.100.10” destination-port=”51” service-name=”junos-ftp”
 nat-source-address=”203.0.113.10” nat-source-port=”12349” nat-destination-address
 ="198.51.100.13" nat-destination-port="3522" nat-rule-name="None"
dst-nat-rule-name="None" protocol-id="6" policy-name="appfw-policy1"
source-zone-name="trust" destination-zone-name="untrust" session-id-22="12245"
username="MyCompany/Administrator " roles="administrators, Users, Enterprise Admins,
 Schema Admins, ad, Domain Users, Group Policy Creator Owners, example-team, Domain
 Admins" packet-incoming-interface="ge-0/0/0.1" application="UNKNOWN"
nested-application="UNKNOWN" encrypted="UNKNOWN"] session created 192.0.2.1/21
junos-ftp 10.1.1.12/32898->10.3.1.10/21 junos-ftp 10.1.1.1/547798->10.1.2.10/21
None None 6 appfw-policy1 trust untrust 20000025 MyCompany/Administrator
(administrators, Users, Enterprise Admins, Schema Admins, ad, Domain Users, Group
 Policy Creator Ownersexample-team, Domain Admins) ge-0/0/0.0 UNKNOWN UNKNOWN
UNKNOWN
<14>1 2015-01-19T15:03:59.427+08:00 user1 RT_FLOW - RT_FLOW_SESSION_CLOSE
[user@host2636 192.0.2.123 reason="idle Timeout” source-address=”198.51.100.13“
source-port=”635”" destination-address=”198.51.100.10” destination-port=”51"
```

```
service-name="junos-ftp" nat-source-address="203.0.113.10" nat-source-port="12349"
  nat-destination-address ="198.51.100.13" "nat-destination-port="3522"
src-nat-rule-name="None" dst-nat-rule-name="None" protocol-id="6"
policy-name="appfw-policy1" source-zone-name="trust"
destination-zone-name="untrust"session-id-32="20000025" packets-from-client="3"
bytes-from-client="180"
packets-from-server="0" bytes-from-server="0" elapsed-time="19"
application="INCONCLUSIVE" nested-application="INCONCLUSIVE" username=" J
“MyCompany /Administrator” roles="administrators, Users, Enterprise Admins,
Schema Admins, ad, Domain Users, Group Policy Creator Owners, example-team,
Domain Admins" packet-incoming-interface="ge-0/0/0.1" encrypted="UNKNOWN"]
session closed idle Timeout: 111.1.1.10/1234>10.1.1.11/21 junos-ftp
10.1.1.12/32898->10.3.1.10/21 1 None None 6 appfw-policy1 trust untrust 20000025
3(180) 0(0) 19
INCONCLUSIVE INCONCLUSIVE MyCompany/Administrator (administrators, Users, Enterprise
 Admins, Schema Admins, ad, Domain Users, Group Policy Creator Owners, example-team,
 Domain Admins) ge-0/0/0.1 UNKNOWN
```

SEE ALSO

source-identity-log (Security) | 616

# Control Network Access Using Device Identity Authentication

**IN THIS SECTION**


- Understanding Access Control to Network Resources Based on Device Identity Information | 244
- Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature | 247
- Understanding the Device Identity Authentication Table and Its Entries | 252
- Understanding How the SRX Series Obtains the Authenticated Device Identity Information From Windows Active Directory for Network Access Control | 257
- Understanding the Device Identity XML Solution for Third-Party NAC Authentication Systems | 258
- Example: Configuring the SRX Series Device Identity Feature in an Active Directory Environment | 260

Based on identity and attributes of the device you can control the access to your network by configuring device identify feature.

## Understanding Access Control to Network Resources Based on Device Identity Information

**IN THIS SECTION**


- Why Use Device Identity Information to Control Access to Your Network | 244
- Background | 244


You can use the integrated user firewall device identity authentication feature to control access to network resources based on the attributes, or characteristics, of the device used. After you configure device identity authentication feature, you can configure security polices that allow or deny traffic from the identified device based on the policy action.

### Why Use Device Identity Information to Control Access to Your Network

For various reasons, you might want to control access to your network resources based on the identity of the user's device rather than on the identity of the user. For example, you might not know the identity of the user. You might allow your users to use their own devices (BYOD) to access network resources and you do not want to use captive portal authenticate. Some companies might have older switches that do not support 802.1, or they might not have a network access Control (NAC) system. The integrated user firewall device identity authentication feature was designed to offer a solution to these and other similar situations by enabling you to control network access based on attributes of the user's device.

### Background

Fundamentally, the device receives or obtains the device identity information from the authentication source in the same manner that it obtains the user identity information, depending on the authentication source. If Microsoft Windows Active Directory is the authentication source, the device retrieves the device information from the Active Directory domain controller. In the case of third-party Network Access Control (NAC) systems, the device receives the information from the authentication source through the RESTful Web services API that the device exposes to it for this purpose. After the device obtains the device identity information, it creates an entry for it in the device identity authentication table.

The purpose of obtaining the device information and entering it into the device identity authentication table is to control user access to network resources based on the device's identity. For this to occur, you must also configure security policies that identify the device, based on the specified device identity profile, and specify the action to be taken on traffic that issues from that device.

In broad terms, the process in which the device identity information is obtained and stored in the device identity information table entails the following actions on the part of the device:

- Getting the device identity information.

  Depending on the authentication source, the device uses one of the following two methods to obtain the device identity information:

  - Active Directory—For Active Directory, the device can extract the device information from the domain controller's event log and then connect to the Active Directory LDAP server to obtain the names of the groups that the device belongs to. The device uses the information that it obtained from the event log to locate the device's information in the LDAP directory.
  - Third-party NAC systems—These authentication systems use the POST service of the RESTful Web services API, called Web API. The device implements the API and exposes to the authentication systems to allow them to send the device identity information to the SRX Series device.

    The API has a formal XML structure and restrictions that the authentication source must adhere to in sending this information to the device.

- Creating an entry for the device in the device identity authentication table.

  After the SRX Series device obtains the device identity information, it creates an entry for it in the device identity authentication table. The device identify authentication table is separate from the Active Directory authentication table or any of the other local authentication tables used for third party authentication sources. Too, unlike local user authentication tables which are particular to an authentication source or feature, the device identity authentication table holds device identity information for all authentication sources. However, only one authentication source, such as Active Directory, can be active at a time. The SRX Series device allows only authentication source to be used at a time to constrain the demand on the system to process information.

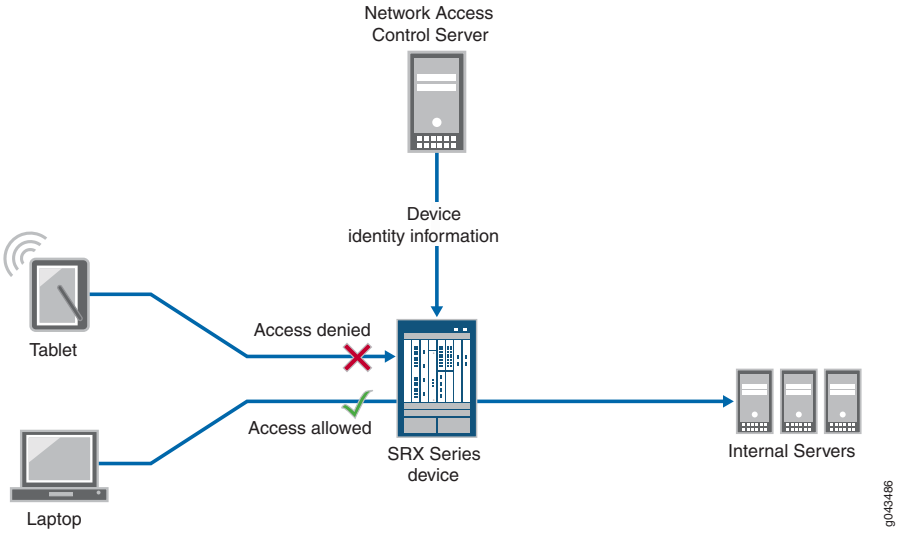
The device identity authentication feature supports various types of authentication systems, such as Active Directory or a third-party authentication source. That is, the device identity authentication feature provides a generic solution that stores device identity information in the same table regardless of the authentication source.

Starting with Junos OS Release 17.4R1, the SRX Series supports IPv6 addresses for user firewall (UserFW) authentication. The device identity table can include entries with IPv6 addresses when active directory is the authentication source.

Figure 16 on page 246 shows the communication between the SRX Series and a third-party NAC authentication source that is used for device identity authentication. The SRX Series device receives the device identity information from the NAC system and stores it in its local device identity authentication

table. A security policy that specifies a device identity profile is applicable to one or more devices. If a device matches the device identity profile and other parts of the security policy, the security policy is applied to traffic issuing from that device.

**Figure 16: Using a Third-Party Network Access Control (NAC) System for Device Identity Authentication**




Use of a device identity profile in a security policy is optional.

If no device identity profile is specified in the security policy's source-end-user-profile field, "any" profile is *assumed*. However, you can not use the keyword "any" in the source-end-user-profile field of a security policy. It is a reserved keyword.

**Release History Table**

| Release | Description |
|---|---|
| 17.4R1 | Starting with Junos OS Release 17.4R1, the SRX Series supports IPv6 addresses for user firewall (UserFW) authentication. The device identity table can include entries with IPv6 addresses when active directory is the authentication source. |

# Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature

IN THIS SECTION


- Device Identity | 247
- Device Identity Profile Contents | 248
- Predefined Device Identity Attributes | 250
- Characteristics of Device Identity Profiles, and Attributes and Target Scaling | 250


The *device identity profile*, referred to in the CLI as the ***end-user-profile***, is a key component of the integrated user firewall device identity authentication feature. It identifies the device and specifies its attributes. The device identity authentication feature allows you to control access to your network resources based on the identity of the device used and not the identity of the user of that device. This feature supports Microsoft Windows Active Directory and third-party network access control (NAC) systems as authentication sources.

This topic focuses on device identity and the device identity profile.

## Device Identity

The device identity essentially consists of the IP address of the device, its name, its domain, and the groups that the device belongs to.

For example, the following output shows information about the device, which is referred to from the device identity profile.

This example shows that the device identity authentication table contains entries for two devices. For each entry, it shows the IP address of the device, the name assigned to the device, and the groups that the device belongs to. Note that both devices belong to the group grp4.

```
Source IP       Device ID      Device-Groups
192.0.2.1       lab-computer1   grp1, grp3, grp4
198.51.100.1    dev-computer2   grp5, grp6, grp4
```

### Device Identity Profile Contents

The device identity profile is a collection of attributes that are characteristics of a specific group of devices, or of a specific device, depending on the attributes configured in the profile. The Packet Forwarding Engine of the device maps the IP address of a device to the device identity profile.

A device identity profile specifies the name of the device and information that includes the IP address of the device, groups to which the device belongs, and attributes of the device which are collectively referred to as the host attributes.

> **NOTE:** The only attributes that you can configure using the CLI are the name of the device and the groups that it belongs to. The other attributes are configured using the third-party RESTful web services API, which is used by NAC systems or Active Directory LDAP.

When traffic from a device arrives at the SRX Series device or NFX Series device, the device obtains the IP address of the device from the first packet of the traffic and uses it to search the device identity authentication table for a matching device identity entry. Then it matches that device identity profile with a security policy whose **source-end-user-profile** field specifies the device identity profile name. If a match is found, the security policy is applied to traffic issuing from the device.

The same device identity profile can also apply to other devices sharing the same attributes. However, for the same security policy to apply, the device and its traffic must match all other fields in the security policy.

A device identity profile must contain the domain name. It might contain more than one set of attributes, but it must contain at least one. Consider the following two sets of attributes that belong to the profile called marketing-main-alice.

The profile contains the following set of attributes:

- alice-T430, as the name of the device.
- MARKETING and WEST-COAST, as the groups that the device belongs to.
- example.net as the name of the domain that it belongs to.

The profile also the following attributes that characterize the device:

- laptop, as the category of the device (device-category)
- Lenovo, as the device vendor (device-vendor)
- ThinkPad T430, as the type of device (device-type)

In cases such as the marketing-main-alice profile that includes the name given to the device, the profile applies exclusively to that device.

However, now suppose that another profile called marketing-west-coast-T430 was configured and that it contains the same attributes as the marketing-main-alice profile with one exception: the name given to the device in the marketing-main-alice profile was not included as an attribute in the marketing-west-coast-T430 profile. In this case, the attributes contained in the profile now make up a group profile. Application of the profile is widened to include all Lenovo ThinkPad T430 devices (which are laptops) that fit the rest of the characteristics, or attributes, defined in the profile.

Devices are covered by the profile if all other attributes match: devices that belong to either the MARKETING or WEST-COAST groups, which the marketing-west-coast-T430 profile specifies as its groups, or to both groups, match the profile.

As mentioned previously, a device identity profile can contain more than one group. A device can also belong to more than one group.

To illustrate further, note that the group device identity profile called marketing-west-coast-T430 also applies to the device called alice-T430 because that device belongs to both the MARKETING and the WEST-COAST groups and it matches all other attributes defined in the profile. Of course, the marketing-main-alice device identity profile still applies to the device called alice-T430. Therefore, the device called alice-T430 belongs to at least two groups, and it is covered by at least two device identity profiles.

Suppose that another profile called marketing-human-resources was defined with all of the attributes of the marketing-west-coast-T430 device identity profile but with these differences: the new device identity profile includes a group called HUMAN-RESOURCES and it does not include the group called WEST-COAST. However, it does contain the MARKETING group.

Because the device called alice-T430 belongs to the MARKETING group, which remains as a group in marketing-human-resources profile, the alice-T430 device also matches the marketing-human-resources device identity profile. Now the alice-T430 device matches three profiles. If the names of any of these profiles is specified in a security policy's source-end-user-profile and the alice-T430 device matches all of the other fields in the security profile, then that profile's action is applied to traffic from that device.

The previous examples of device identity profiles illustrate the following points:

- A profile can be defined to identify only one device or it can be defined to apply to many devices.
- A device identity profile can contain more than one group to which a given device belongs.
- A device can match more than one device identity profile by matching the characteristics, or attributes, including at least one of the groups, configured for the profile.

The flexible use of device identity profiles will become evident when you configure security policies that are designed to include the source-end-user-profile field, in particular when you want the policy's action to be applied to a number of devices.

### Predefined Device Identity Attributes

The SRX Series device provides the predefined device identity policy attributes that are configured using the third-party RESTful web services API, which is used by NAC systems or Active Directory LDAP.

- device-identity
- device-category
- device-vendor
- device-type
- device-os
- device-os-version

You specify values for these attributes in a device identity profile.

### Characteristics of Device Identity Profiles, and Attributes and Target Scaling

This section describes how the SRX Series and NFX Series devices treat device identity attributes and profiles. It also gives device-independent and device-dependent scaling numbers for these entities.

The following attribute and profile characteristics apply to their use on all supported SRX Series and NFX Series devices.

- The maximum length of the following entities is 64 bytes: device identity profile names (referred to in the CLI as **end-user-profile**) attribute names, attribute-values.
- You can not overlap values in a range if you configure more than one digital value range for the same attribute.
- When the device matches a device identity profile to a security policy, all of the attributes in the profile are taken into account. Here is how they are treated:
  - If the device identity profile contains multiple values for an attribute, the values of that attribute are treated individually. It is said that they are ORed.

    For the security policy to be applied to the device, the following conditions must be met. The device must match:

    - One of the values for each attribute that has multiple values.
    - The rest of the attribute values specified in the device identity profile.
    - The security policy field values.
  - All individual attributes that have a single value are treated separately and considered together as a collection of values—that is, the AND operation is applied to them. The device uses its standard policy-matching criteria in handling these attributes.

Table 13 on page 251 shows the platform-independent scaling values used in the device identity authentication feature.

**Table 13: Platform-Independent Scaling**

| Item | Maximum |
|---|---|
| Values per attribute | 20 |
| Attributes per profile | 100 |
| Device identity profile specification per security policy (source-end-user-profile) | 1 |

Table 14 on page 251 shows the platform-dependent scaling values used in the device identity authentication feature..

**Table 14: Platform-Dependent Scaling**

| Platform | Maximum Number of Profiles | Maximum Total Number of Attribute Values |
|---|---|---|
| SRX5000 Series | 4000 | 32000 |
| SRX Series 1500 | 1000 | 8000 |
| SRX Series 550M | 500 | 4000 |
| SRX Series 300 and SRX Series 320 | 100 | 1000 |
| SRX Series 340 and SRX Series 345 | 100 | 1000 |
| SRX Series 4100-4XE | 1000 | 8000 |
| SRX Series 4200-8XE | 2000 | 16000 |
| vSRX | 500 | 4000 |
| NFX150 | 100 | 1000 |

The following changes to device identity profiles and their use in security policies do not cause the device to perform a session scan:

- Updates to a profile which is referenced in a security policy.
- Updates to the profile configuration.

- Updates to attributes that are made through the RESTful web services API, which is used by NAC systems, or Active Directory LDAP.

## Understanding the Device Identity Authentication Table and Its Entries

**IN THIS SECTION**


- The Device Identity Authentication Table | **252**
- Why the Device Identity Authentication Table Content Changes | **253**
- Security Policy Matching and Device Identity Profiles | **256**


The device contains a number of local authentication tables used for user authentication for various purposes. For example, the device contains a local Active Directory authentication table for user authentication when Microsoft Windows Active Directory is used as the authentication source.

When you configure the device to use the integrated user firewall device identity authentication feature for authentication based on the device identity and its attributes, the device creates a new table called the device identity authentication table.

To gain a complete view of the device identity authentication feature, it helps to understand this table, its contents, and its relationship to other entities.

This topic covers the device identity authentication table and its device entries, and how the table contents change based on several factors.

### The Device Identity Authentication Table

Unlike other local authentication tables, the device identity authentication table does not contain information about a user but rather about the user's device. Moreover, unlike user authentication tables, it does not contain information about devices authenticated by one authentication source. Rather, it serves as a repository for device identity information for all devices regardless of their authentication source. For example, it might contain entries for devices authenticated by Active Directory or third-party NAC authentication sources.

A device identity authentication table entry contains the following parts:

- The IP address of the device.
- The name of the domain that the device belongs to.

- The groups with which the device is associated.
- The device identity.

  The device identity is actually that of a device identity profile (referred to in the CLI as **end-user-profile**). This type of profile contains a group of attributes that characterize a specific individual device or a specific group of devices, for example, a specific type of laptop.

Starting in Junos OS 17.4R1, the SRX Series device supports IPv6 addresses for user firewall module (UserFW) authentication. This feature allows IPv6 traffic to match any security policy configured for source identity. Previously, if a security policy was configured for source identity and “any” was specified for its IP address, the UserFW module ignored the IPv6 traffic.

IPv6 addresses are supported for the following authentication sources:

- Active directory authentication table
- Device identity with Active Directory authentication
- Local authentication table
- Firewall authentication table

### Why the Device Identity Authentication Table Content Changes

The device identity entries in the device identity authentication table are changed when certain events occur: when the user authentication entry with which the device identity entry is associated expires, when security policy changes occur in regard to referencing a group that the device belongs to, when the device is added to or removed from groups, or when groups that it belongs to are deleted and that change is made to the Windows Active Directory LDAP server.

- When the User Identity Entry with Which a Device Identity Entry Is Associated Expires

  When the device generates an entry for a device in the device identity authentication table, it associates that entry with a user identity entry in a local authentication table for the specific authentication source that authenticated the user of the device, such as Active Directory. That is, it ties the device identity entry in the device identity authentication table to the entry for the user of the device in the user authentication table.

  When the user authentication entry with which the device identity entry is associated expires and is deleted from the user authentication table, the device identity entry is deleted silently from the device identity authentication table. That is, no message is issued to inform you of this event.

- When Security Policy Changes Occur in Regard to Referencing a Group to Which the Device Belongs

  To control access to network resources based on device identity, you create a device identity profile that you can refer to in a security policy. In addition to other attributes, a device identity profile contains the names of groups. When a device identity profile is referenced by a security policy, the groups that it contains are referred to as *interested groups*.

A group qualifies as an *interested group* if it is referenced by a security policy—that is, if it is included in a device identity profile that is specified in the **source-end-user-devic**e field of a security policy. If a group is included in a device identity profile that is not currently used in a security policy, it is not included in the list of interested groups. A group can move in and out of the list of groups referenced by security policies.

- When a Device Is Added to or Removed from a Group or a Group Is Deleted

  To keep the device identity entries in the local device identity authentication table current, the SRX Series or NFX Series monitors the Active Directory event log for changes. In addition to determining whether a device has logged out of or in to the network, it can determine changes to any groups that the device might belong to. When changes occur to the groups that a device belongs to—that is, when a device is added to or removed from a group or the group is deleted—the device modifies the contents of the affected device entries in its own device identity authentication table to reflect the changes made in the Microsoft Windows Active Directory LDAP server.

The device identity authentication table is updated according to changes to groups with which the device is associated in the LDAP server, as illustrated in Table 15 on page 254.

**Table 15: Group Changes for Devices in the Active Directory LDAP and the SRX Series or NFX Series Response**

| Changes Made to LDAP | SRX Series or NFX Series LDAP Message and UserID Daemon Action |
|---|---|
| Group information for a device has changed. The device has been added to or removed from a group, or a group that the device belongs to has been deleted. | The Active Directory LDAP module sends notification of the change to the SRX Series or NFX Series UserID daemon, directing it to revise information in its local device identity authentication table.<br><br>The device processes these messages every 2 minutes. |
| The device entry in LDAP is deleted. | The Active Directory LDAP module sends notification of the change to the UserID daemon, directing it to revise information in its local device identity authentication table.<br><br>The device processes these messages every 2 minutes. |

The SRX Series or NFX Series device UserID daemon is informed of the changes. Whether or not a group that a device belongs to is specified in a security policy has bearing on what information is stored in device identity authentication table entries for the affected device. Table 16 on page 255 shows the activity that occurs when a group is added to or deleted from the Active Directory LDAP.

**Table 16: Changes to Device Identity Entries Based on Security Policy Specifications**

| Device Identity Profile Changes | Device-Group Mapping Behavior | SRX Series or NFX Series UserID Daemon Response |
|---|---|---|
| A new group that was added to the Active Directory LDAP is added to the SRX Series device identity profile. | The device gets the list of devices that belong to the new group and its subgroups from the Active Directory LDAP server. It adds the list to its local LDAP directory. | The UserID daemon determines whether the device identity authentication table includes entries for the set of affected devices. If so, it updates the group information for these entries.<br><br>For example, here is the entry for device1 before it was updated to include the new group and after the group was added:<br><br>• device1, g1<br>• device1, g1, g2 |
| A group is deleted from the Active Directory LDAP. The device deletes the group from the device identity profile. | The device gets the list of devices that belong to the deleted group from its local LDAP database.<br><br>It deletes the device-group mapping from the local LDAP directory. | The UserID daemon checks the device identity authentication table for entries that belong to the group. It removes the group from affected entries.<br><br>For example, here is the entry for device1 before the group was deleted and after the group was deleted:<br><br>• device1, g1, g2<br>• device1, g1 |

Table 17 on page 255 elaborates on the contents of device authentication entries for several devices that are affected by deletion of a group.

**Table 17: Changes to Device Identity Authentication Table Resulting from LDAP and Security Policy Changes**

| Changes to Device identity Authentication Table Entries | | |
|---|---|---|
| IP Address | Device Information | Group |
| Original Entries | | |
| 192.0.2.10 | device1 | group1, group2 |
| 192.0.2.11 | device2 | group3, group4 |

**Table 17: Changes to Device Identity Authentication Table Resulting from LDAP and Security Policy Changes *(continued)***

| Changes to Device identity Authentication Table Entries | | |
|---|---|---|
| IP Address | Device Information | Group |
| 192.0.2.12 | device3 | group2 |
| Same Entries After group2 Is Deleted | | |
| 192.0.2.10 | device1 | group1 |
| 192.0.2.11 | device2 | group3, group4 |
| 192.0.2.12 | device3 | *This entry no longer contains groups.* |

### Security Policy Matching and Device Identity Profiles

The device follows the standard rules for matching traffic against security policies. The following behavior pertains to the use of a device identity profile in a security policy for determining a match:

- Use of a device identity profile in a security policy is optional.
  - If no device identity profile is specified in the source-end-user-profile field, **any** profile is assumed.
  - You cannot use the keyword **any** in the **source-end-user-profile** field of a security policy.

    If you use the source-end-user-profile field in a security policy, you must reference a specific profile. The device from which the access attempt is issued must match the profile's attributes.
- Only one device identity profile can be specified in a single security policy.
- A security policy rematch is triggered when the **source-end-user-profile** field value of the security policy is changed. No rematch is triggered when an attribute value of a profile is changed.

**Release History Table**

| Release | Description |
|---|---|
| 17.4R1 | Starting in Junos OS 17.4R1, the SRX Series device supports IPv6 addresses for user firewall module (UserFW) authentication. This feature allows IPv6 traffic to match any security policy configured for source identity. Previously, if a security policy was configured for source identity and "any" was specified for its IP address, the UserFW module ignored the IPv6 traffic. |

## Understanding How the SRX Series Obtains the Authenticated Device Identity Information From Windows Active Directory for Network Access Control

You can use the integrated user firewall device identity authentication feature to control access to your network resources based on the identity and attributes of the device used rather than the user identity. Information about a device is stored in the device identity authentication table. You can specify the name of a device identity profile that contains the device attributes in the source-end-user-profile field of a security policy. If all conditions are met, the security policy's actions are applied to traffic issuing from that device.

For you to be able to use device identity profiles in security policies, the SRX Series or NFX Series device must obtain the device identity information for authenticated devices. The device creates the device identity authentication table to use to store device identity entries. It searches the table for a device match when traffic arrives from a device. This topic considers the process followed when Active Directory is used as the authentication source.

An Active Directory domain controller authenticates users when they log in to the domain, and it writes a record of that event to the Windows event log. It also writes a record to the event log when a user logs out of the domain. The domain controller event log provides the device with information about authenticated devices that are currently active in the domain and when those devices are logged out from it.

The UserID daemon takes the following actions:

1. It reads the Active Directory domain controller event logs to obtain the IP addresses of devices logged into the domain and authenticated by Windows.

   The UserID daemon in the device Routing Engine implements a Windows Management Instrumentation (WMI) client with Microsoft Distributed COM/Microsoft RPC stacks and an authentication mechanism to communicate with a Windows Active Directory domain controller in an Active Directory domain. Using event log information retrieved from the Active Directory controller, the process obtains the IP addresses of active Active Directory devices. The process monitors Active Directory event log changes using the same WMI DCOM interface to adjust its device identity information in its local authentication table to reflect any changes made to the Active Directory server.

2. It uses the device IP addresses that it obtained from the event log to obtain information about the groups that a device belongs to. To obtain this group information, the device connects to the LDAP service in the Active Directory controller using the LDAP protocol for this purpose.

As a result of this process, the device is able to generate entries for the devices in the device identity authentication table. After it generates an entry for a device in the device identity authentication table, the device associates that entry with the appropriate user entry in its local Active Directory authentication table. You can then reference the device identity profile entries in security policies to control access to resources.

Behavior and Constraints

- The process of reading the event log consumes domain controller CPU resources which may lead to high CPU usage in the domain controller. For this reason, the Active Directory domain controller should have a high-performance configuration of at least 4 cores and 8 gigabytes of memory.
- The domain controller event log records a maximum length of 16 bytes of the device ID, including a null terminator. Therefore, the maximum length of the device ID that the device obtains from the domain controller is 15 bytes.
- If the domain controller clears the event log or if the data written to the event log is missing or delayed, the device identity mapping information might be inaccurate. If the SRX Series or NFX Series device is unable to read the event log or if it contains null data, the device reports an error condition in its own log.
- If the device identity information table reaches capacity, it cannot add new device identity entries. In that case, traffic from the device is dropped.

## Understanding the Device Identity XML Solution for Third-Party NAC Authentication Systems

**IN THIS SECTION**


- XML Web API Implementation on SRX Series and NFX Series Devices | **259**
- Ensuring the Integrity of Data Sent from the NAC Service to the SRX Series or NFX Series Device | **259**
- Data Size Restrictions and Other Constraints | **259**


The integrated user firewall device identity authentication feature enables you to control access to network resources based on the identity of a device. You can use one of the following device identity solutions:

- Microsoft Active Directory as the authentication source.

  If your environment is set up to use Microsoft Active Directory, the SRX Series or NFX Series device obtains the device IP address and groups from the Active Directory domain controller and LDAP service.

- Network access control (NAC) authentication system.

  If your network environment is configured for a NAC solution and you decide to take this approach, the NAC system sends the device identity information to the SRX Series or NFX Series device. The RESTful Web services API enables you to send the device information to the device in a formal XML structure.

**WARNING:** If you take this approach, you must verify that your NAC solution works with the device.

### XML Web API Implementation on SRX Series and NFX Series Devices

The RESTful Web services API enables you to send the device identity information to the SRX Series or NFX Series device in a formal XML structure. It allows your NAC solution to integrate with the device and efficiently send the device information to it. You must adhere to the formal structure and restrictions in sending information to the device using the API.

### Ensuring the Integrity of Data Sent from the NAC Service to the SRX Series or NFX Series Device

The following requirements ensure that the data sent from the NAC service is not compromised:

- The API implementation is restricted to processing only HTTP/HTTPS POST requests. Any other type of request that it receives generates an error message.
- The API daemon analyzes and processes HTTP/HTTPS requests from only the following dedicated URL:

  ```
  /api/userfw/v1/post-entry
  ```

- The HTTP/HTTPS content that your NAC solution posts to the SRX Series device must be consistently formatted correctly. The correct XML format indicates a lack of compromise, and it ensures that user identity information is not lost.

### Data Size Restrictions and Other Constraints

The following data size restrictions and limitations apply to the data posted to the SRX Series or NFX Series device:

- The NAC authentication system must control the size of the data that it posts. Otherwise, the Web API daemon is unable to process it. The Web API daemon can process a maximum of 2 megabytes of data.
- The following limitations apply to XML data for role and device posture information. The Web API daemon discards XML data sent to it that exceeds these amounts (that is, the overflow data):
  - The device can process a maximum of 209 roles.
  - The device supports only one type of posture with six possible posture tokens, or values. Identity information for an individual user can have only one posture token.

# Example: Configuring the SRX Series Device Identity Feature in an Active Directory Environment

IN THIS SECTION


- Requirements | 260
- Overview | 261
- Configuration | 264
- Verification | 270


This example shows how to configure the integrated user firewall device identity authentication feature to control access to network resources based on the identity of an authenticated device, not its user. This example uses Microsoft Active Directory as the authentication source. It covers how to configure a device identity profile that characterizes a device, or set of devices, and how to reference that profile in a security policy. If a device matches the device identity and the security policy parameters, the security policy's action is applied to traffic issuing from that device.

For various reasons, you might want to use the identity of a device for resource access control. For example, you might not know the identity of the user. Also some companies might have older switches that do not support 802.1, or they might not have a network access control (NAC) system. The device identity authentication feature was designed to offer a solution to these and other similar situations by enabling you to control network access based on the device identity. You can control access for a group of devices that fit the device identity specification or an individual device.

## Requirements

This example uses the following hardware and software components:

- An SRX Series Services Gateway device running Junos OS Release 15.1X49-D70.
- Microsoft Active Directory with a domain controller and the Lightweight Directory Access Protocol (LDAP) server

  The Active Directory domain controller has a high-performance configuration of 4 cores and 8 gigabytes of memory.

> **NOTE:** The SRX Series obtains the IP address of a device by reading the domain controller event log. The process that reads the event log consumes domain controller CPU resources, which might lead to high CPU usage. For this reason, the Active Directory domain controller should have a high-performance configuration of at least 4 cores and 8 gigabytes of memory.

- A server on the internal corporate network.

### Overview

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, the SRX Series provides support for controlling access to network resources based on the identity of a device authenticated by Active Directory or a third-party network access control (NAC) system. This example uses Active Directory as the authentication source.

> **NOTE:** You must configure the authentication source for this feature to work.

This example covers the following configuration parts:

- Zones and their interfaces

  You must configure the zones to which the source and destination entities specified in the security policy belong. If you do not configure them, the security policy that references the device identity profile will be invalid.

- A device identity profile

  You configure the device identity profile apart from the security policy; you refer to it from a security policy. A device identity profile specifies a device identity that can be matched by one or more devices. For Active Directory, you can specify only the device-identity attribute in the profile.

  In this example, the device-identity attribute specification is company-computers.

  > **NOTE:** The device identity profile is referred to as **end-user-profile** in the CLI.

- A security policy

  You configure a security policy whose action is applied to traffic issuing from any device that matches the device identity profile attributes and the rest of the security policy's parameters.

> **NOTE:** You specify the name of the device identity profile in the security policy's ***source-end-user-profile*** field.

- Authentication source

  You configure the authentication source to be used to authenticate the device. This example uses Active Directory as the device identity authentication source.

  If Active Directory is the authentication source, the SRX Series obtains identity information for an authenticated device by reading the Active Directory domain's event log. The device then queries the LDAP interface of Active Directory to identify the groups that the device belongs to, using the device's IP address for the query.

  For this purpose, the device implements a Windows Management Instrumentation (WMI) client with Microsoft Distributed COM/Microsoft RPC stacks and an authentication mechanism to communicate with the Windows Active Directory controller in the Active Directory domain. It is the device wmic daemon that extracts device information from the event log of the Active Directory domain.

  The wmic daemon also monitors the Active Directory event log for changes by using the same WMI DCOM interface. When changes occur, the device adjusts its local device identity authentication table to reflect those changes.

  Starting with Junos OS Release 17.4R1, you can assign IPv6 addresses to Active Directory domain controllers and the LDAP server. Prior to Junos OS Release 17.4R1, you could assign only IPv4 addresses.

***Topology***

In this example, users who belong to the marketing-zone zone want to access resources on the internal corporate servers. Access control is based on the identity of the device. In this example, company-computers is specified as the device identity. Therefore, the security policy action is applied only to devices that fit that specification and match the security policy criteria. It is the device that is either granted or denied access to the server resources. Access is not controlled based on user identification.

Two SRX Series zones are established: one that includes the network devices (marketing-zone) and one that includes the internal servers (servers-zone). The SRX Series device interface ge-0/0/3.1, whose IP address is 192.0.2.18/24, is assigned to the marketing-zone zone. The SRX Series device interface ge-0/0/3.2, whose IP address is 192.0.2.14/24, is assigned to the servers-zone zone.

This examples covers the following activity:

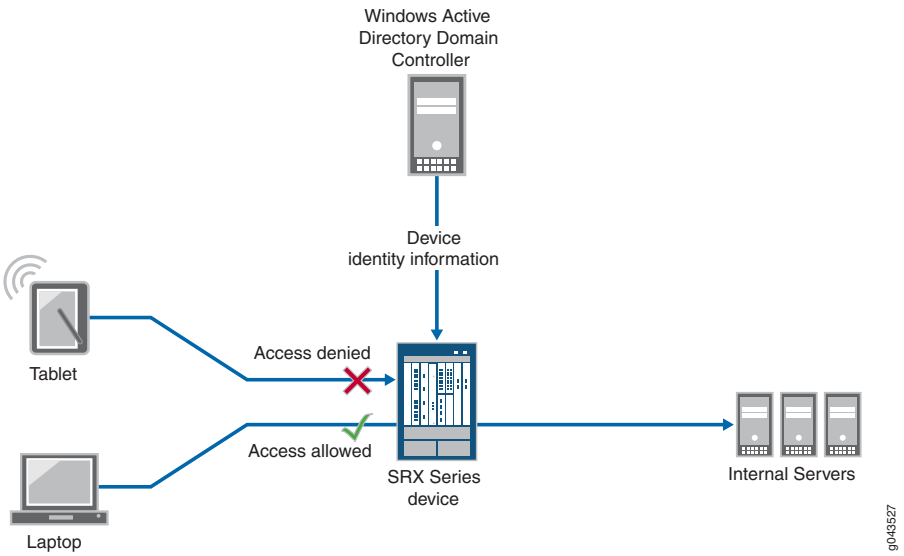
1. The SRX Series device connects to the Active Directory domain controller using the WMI DCOM interface to obtain information about devices authenticated by Active Directory.

   When a user logs in to the network and is authenticated, information about the user's device is written to the event log.

2. The SRX Series extracts the device information from the event log of the Active Directory domain controller.
3. The SRX Series uses the extracted information to obtain a list of the groups that the device belongs to from the Active Directory LDAP server.
4. The SRX Series creates a local device identity authentication table and stores the device identity information that it obtained from the domain controller and LDAP server in the table.
5. When traffic from a device arrives at the SRX Series device, the SRX Series checks the device identity authentication table for a matching entry for the device that issued the traffic.
6. If the SRX Series finds a matching entry for the device that is requesting access, it checks the security policy table for a security policy whose **source-end-user-profile** field specifies a device identity profile with a device-identity specification that matches that of the device requesting access.
7. The matching security policy is applied to traffic issuing from the device.

Figure 17 on page 264 show the topology for this example.

Figure 17: Topology for the Device Identity Feature with Active Directory as the Authentication Source




## Configuration

**IN THIS SECTION**


- Configuring the Integrated User Firewall Device Identity Authentication Feature in an Active Directory Environment | **265**
- Results | **267**


To configure the device identity feature in an Active Directory environment, perform these tasks:

### CLI Quick Configuration

To quickly configure this example, copy the following commands to a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/3.1 family inet address 192.0.2.18/24
set interfaces ge-0/0/3.2 family inet address 192.0.2.14/24
set security zones security-zone marketing-zone interfaces ge-0/0/3.1 host-inbound-traffic system-services
    all
set security zones security-zone marketing-zone interfaces ge-0/0/3.1 host-inbound-traffic protocols all
set security zones security-zone servers-zone interfaces ge-0/0/3.2 host-inbound-traffic system-services all
```

```
set security zones security-zone servers-zone interfaces ge-0/0/3.2 host-inbound-traffic protocols all
set services user-identification device-information authentication-source active-directory
set services user-identification device-information end-user-profile profile-name marketing-west-coast
    domain-name example.net
set services user-identification device-information end-user-profile profile-name marketing-west-coast attribute
    device-identity string company-computers
set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access match
    source-address any destination-address any
set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access match application
    any
set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access match
    source-end-user-profile marketing-west-coast
set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access then permit
set services user-identification active-directory-access domain example.net user user1 password pswd
set services user-identification active-directory-access domain example.net domain-controller dc-example
    address 203.0.113.0
set services user-identification active-directory-access domain example.net ip-user-mapping discovery-method
    wmi event-log-scanning-interval 30
set services user-identification active-directory-access domain example.net ip-user-mapping discovery-method
    wmi initial-event-log-timespan 1
set services user-identification active-directory-access domain example.net user-group-mapping ldap
    authentication-algorithm simple
set services user-identification active-directory-access domain example.net user-group-mapping ldap address
    198.51.100.9 port 389
set services user-identification active-directory-access domain example.net user-group-mapping ldap base
    dc=example,dc=net
set services user-identification active-directory-access authentication-entry-timeout 100
set services user-identification active-directory-access wmi-timeout 60
```

***Configuring the Integrated User Firewall Device Identity Authentication Feature in an Active Directory Environment***

**Step-by-Step Procedure**

This procedure includes the configuration statements required to configure the SRX Series device to support the device identity authentication feature in an Active Directory environment.

1. Configure the interfaces to be used for the marketing-zone and the servers-zone.

```
[edit interfaces]
user@host# set ge-0/0/3.1 family inet address 192.0.2.18/24
user@host# set ge-0/0/3.2 family inet address 192.0.2.14/24
```

2. Configure the marketing-zone and the servers-zone and assign interfaces to them.

```
[edit security zones]
user@host# set security-zone marketing-zone interfaces ge-0/0/3.1 host-inbound-traffic system-services
    all
user@host# set security-zone marketing-zone interfaces ge-0/0/3.1 host-inbound-traffic protocols all
user@host# set security-zone servers-zone interfaces ge-0/0/3.2 host-inbound-traffic system-services all
user@host# set security-zone servers-zone interfaces ge-0/0/3.2 host-inbound-traffic protocols all
```

3. Configure the authentication source to specify Microsoft Active Directory. You must specify the authentication source for the device identity feature to work. This is a required value.

```
[edit services user-identification]
user@host# set device-information authentication-source active-directory
```

4. Configure the device identity specification for the device identity profile, which is also referred to as **end-user-profile**.

```
[edit services user-identification]
user@host# set device-information end-user-profile profile-name marketing-west-coast domain-name
    example.net
user@host#set device-information end-user-profile profile-name marketing-west-coast attribute
    device-identity string company-computers
```

5. Configure a security policy, called mark-server-access, that references the device identity profile called marketing-west-coast. The security policy allows any device that belongs to the marketing-zone zone (and that matches the device identity profile specification) access to the target server's resources.

```
[edit security policies]
user@host# set from-zone marketing-zone to-zone servers-zone policy mark-server-access match
    source-address any destination-address any
user@host# set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access
    match source-end-user-profile marketing-west-coast
user@host# set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access
    match application any
user@host# set security policies from-zone marketing-zone to-zone servers-zone policy mark-server-access
    then permit
```

6. Configure the SRX Series device to communicate with Active Directory and to use the LDAP service.

   To get the group information necessary to implement the device identity authentication feature, the SRX Series device uses the Lightweight Directory Access Protocol (LDAP). The SRX Series acts as an LDAP client communicating with an LDAP server. Typically, the Active Directory domain controller

acts as the LDAP server. The LDAP module in the device, by default, queries the Active Directory in the domain controller.

```
[edit services user-identification]
user@host# set active-directory-access domain example.net user user1 password pswd
user@host# set active-directory-access domain example.net domain-controller dc-example address
  203.0.113.0
user@host# set active-directory-access domain example.net ip-user-mapping discovery-method wmi
  event-log-scanning-interval 30
user@host# set active-directory-access domain example.net ip-user-mapping discovery-method wmi
  initial-event-log-timespan 1
user@host set active-directory-access domain example.net user-group-mapping ldap address 198.51.100.9
  port 389
user@host# set active-directory-access domain example.net user-group-mapping ldap base dc=example,dc=net
user@host# set active-directory-access domain example.net user-group-mapping ldap
  authentication-algorithm simple
user@host# set active-directory-access authentication-entry-timeout 100
user@host# set active-directory-access wmi-timeout 60
```

***Results***

Enter **show interfaces** in configuration mode.

```
user@host#show interfaces
ge-0/0/3 {
    unit 1 {
        family inet {
            address 192.0.2.18/24;
        }
    }
    unit 2 {
        family inet {
            address 192.0.2.14/24;
        }
    }
}
```

Enter **show security zones** in configuration mode.

```
user@host#show security zones
security-zone marketing-zone {
    interfaces {
        ge-0/0/3.1 {
```

```
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
security-zone servers-zone {
    interfaces {
        ge-0/0/3.2 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
```

Enter **show services user-identification device-information end-user-profile** in configuration mode.

```
user@host#show services user-identification device-information end-user-profile
domain-name example.net
attribute device-identity {
    string company-computers;
}
```

Enter **show services user-identification device-information authentication-source** in configuration mode.

```
user@host#show services user-identification device-information authentication-source
active-directory;
```

Enter **show security policies** in configuration mode.

```
user@host#show security policies
from-zone marketing-zone to-zone servers-zone {
```

```
    policy mark-server-access {
        match {
            source-address any;
            destination-address any;
            application any;
            source-end-user-profile {
                marketing-west-coast;
            }
        }
        then {
            permit;
        }
    }
}
```

Enter **show services user-identification active-directory-access** in configuration mode.

```
user@host#show services user-identification active-directory-access
domain example-net {
    user {
        user1;
        password $ABC123; ## SECRET-DATA
    }
    ip-user-mapping {
        discovery-method {
            wmi {
                event-log-scanning-interval 30;
                initial-event-log-timespan 1;
            }
        }
    }
    user-group-mapping {
        ldap {
            base dc=example,DC=net;
            address 198.51.100.9 {
                port 389;
            }
        }
    }
}
```

Enter **show services user-identification active-directory-access domain example-net** in configuration mode.

```
user@host#show services user-identification active-directory-access domain example-net
user {
    user1;
    password $ABC123 ## SECRET-DATA
}
domain-controller dc-example {
    address 203.0.113.0;
}
```

### Verification

**IN THIS SECTION**


- Verify the Device Identity Authentication Table Contents | **270**


#### *Verify the Device Identity Authentication Table Contents*

**Purpose**

Verify that the device identity authentication table contains the expected entries and their groups.

**Action**

In this case, the device identity authentication table contains three entries. The following command displays extensive information for all three entries.

Enter **show services user-identification device-information table all extensive** command in operational mode to display the table's contents.

## Sample Output

```
Domain: example.net
Total entries: 3
  Source IP: 192.0.2.19
    Device ID: example-dev1
    Device-Groups: device_group1,
    device_group2,device_group3,
    device_group4, device_group5
    device-identity: company-computers
```

```
    Location1: us1
    Referred by: mark-server-access
  Source IP: 192.0.2.22
    Device ID: example-dev2
    Device-Groups: device_group06,
    device_group7, device_group8,
    device_group9, device_group10
    device-identity: company-computers
    Location1: us1
    Referred by: mark-server-access
Source IP: 192.0.2.19
   Device ID: example-dev3
   Device-Groups: device_group1, device_group2,
   device_group3, device_group4, device_group5
   device-identity: company-computers
   Location1: us1
   Referred by: mark-server-access
```

**Meaning**

The table should contain entries with information for all authenticated devices and the groups that they belong to.

**Release History Table**

| Release | Description |
|---|---|
| 17.4R1 | Starting with Junos OS Release 17.4R1, the SRX Series supports IPv6 addresses for user firewall (UserFW) authentication. The device identity table can include entries with IPv6 addresses when active directory is the authentication source. |
| 17.4R1 | Starting in Junos OS 17.4R1, the SRX Series device supports IPv6 addresses for user firewall module (UserFW) authentication. This feature allows IPv6 traffic to match any security policy configured for source identity. Previously, if a security policy was configured for source identity and “any” was specified for its IP address, the UserFW module ignored the IPv6 traffic. |

5

CHAPTER

# Identity Management User Firewall

Configure Juniper Identity Management Service to Obtain User Identity Information | 275

# Configure Juniper Identity Management Service to Obtain User Identity Information

IN THIS SECTION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- Configuring Advanced Query Feature for Obtaining User Identity Information from JIMS | 280
- Example: Configuring the Advanced Query Feature for Obtaining User Identity Information from JIMS | 285
- Example: Configuring Filter for Advanced Query Feature | 291


Juniper Identity Management Service (JIMS) is a standalone Windows service application that collects and maintains a large database of user, device, and group information from Active Directory domains. JIMS enables the device to rapidly identify thousands of users in a large, distributed enterprise.

## Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS

IN THIS SECTION


- Overview | 275
- Establishing a Connection to JIMS to Obtain User Identity Information | 276
- Querying JIMS for User Identity Information | 277
- Filters | 278
- Caveats and Limitations | 279


### Overview

Juniper Identity Management Service (JIMS) is a software agent and repository that collects user name, device identity, and group information from various sources. JIMS supports Microsoft active directory and Microsoft Exchange Server.

The SRX Series or NFX Series device relies on JIMS to obtain user identity information much in the same way that it does LDAP.

If you configure the advanced user query feature, the device:

- Can query JIMS for identity information.
- Populate identity management authentication table with the information that is obtained from JIMS.
- Use the populated identity management authentication table to authenticate a user or a device requesting access to a protected resource.

If JIMS does not contain information for a user, you can push that information to the device. The user must first authenticate to the device through captive portal.

The advanced query feature also allows you to push authentication entries to the JIMS server for users for whom there are no entries in JIMS but who have successfully authenticated to the device through captive portal.

User identity information that JIMS sends in response to the device queries includes:

- IP address of the user's device.
- User name.
- Domain that the user's device belongs to.
- Roles that the user belongs to, such mycompany-pc. CEO. user-authenticated.
- If the device is online and the state of the device, such as "Healthy".
- End-user-attributes, such as device-identity, value (device name), and groups that the device belongs to.

### Establishing a Connection to JIMS to Obtain User Identity Information

The device obtains user identity information by querying JIMS either in batch mode to obtain information for groups of users or through queries for individual users. For the device to query JIMS, you must establish an HTTPS connection between the device and the JIMS server.

> **NOTE:** HTTP connections are used only for debugging purposes.

Defining the connection entails configuring the following information:

- Connection parameters.
- Authentication information that allows the device to authenticate to JIMS.

The device obtains an access token after it authenticates to the JIMS server. The device must use this token to query JIMS for user information.

- You can also configure this information for connection to a *secondary*, backup server.

Starting in Junos OS Release 18.3R1, IPv6 addresses are supported to connect JIMS primary server and secondary server, in addition to existing IPv4 address support.

The device attempts to connect to the primary server first and in case of failed attempt, it switches to the secondary server. Even after connecting to the secondary server, the device periodically probes the failed primary server and reverts to the primary server when it is available again.

Starting with Junos OS Release 18.1R1, you can configure an IPv6 address for Web API function to allow the JIMS to initiate and establish a secure connection. The Web API supports the IPv6 user or device entries obtained from JIMS. Prior to Junos OS Release 18.1R1, only IPv4 addresses were supported.

### Querying JIMS for User Identity Information

There are three ways to obtain user identity information from JIMS:

- Initial batch query at startup—When the device is started, it sends a batch query message to JIMS to obtain all available user identity information for active directory users that it expects at that time, if you have configured the device connection to the JIMS server.
- Follow-on batch queries—Following its initial receipt of user identity information, the device queries JIMS periodically for batches of newly generated user identity information. For this to occur, you configure an interval for the periodic queries and specify the number of user identity records to be sent in return per batch. Starting with Junos OS Release 18.1R1, the device can query JIMS for IPv6 user or device information. Prior to Junos OS Release 18.1R1, only IPv4 addresses were supported.
- Query for individual user information—You can configure the advanced query feature to allow you to query the JIMS server for identity information for an individual user based on the IP address of the user's device, if that information is missing from a batch response. Starting with Junos OS Release 18.1R1, the device can query JIMS for IPv6 user or device information when IPv6 traffic arrives on the device.

> **NOTE:** If an entry for the specified IP address does not exist, JIMS returns an HTTP 404 "Not Found" message.

> **NOTE:** When the device requests user information from JIMS initially, it specifies a timestamp. JIMS sends user information in response going back to the timestamp specification, and it includes a cookie to the device in the response to indicate the context. The device sends that cookie with its next query instead of a timestamp.

You can refresh the user identity information in your identity management authentication table obtained from JIMS. You can obtain everything that was received automatically when you started the device and from subsequent batch queries and individual IP queries up to the present.

For this purpose, you clear the authentication table by disabling the advanced query feature configuration. Afterward, you can reconfigure the advanced query feature to retrieve all available user identities.

Starting with Junos OS Release 18.1R1, devices can search the identity management authentication table for information based on IPv6 addresses. Prior to Junos OS Release 18.1R1, the devices read only IPv4 addresses. The device supports the use of IPv6 addresses associated with source identities in security policies. If an IPv4 or IPv6 entry exists, policies matching that entry are applied to the traffic and access is either allowed or denied.

### Filters

The advanced query feature provides an optional filter function that you can use to control at a granular level the user information records that you want to receive in response to queries. You can configure filters based on IP addresses and domains. Filters allow you to define specifically users whose information you want JIMS to return to you in response to queries.

You can configure filters composed of:

- A range of IP addresses. You can specify a range of IP addresses for:
  - Users whose information you want to receive.
  - Users for whom you do not want information.

  Starting in Junos OS Release 18.3R1, SRX Series devices support IPv6 addresses to configure the filters based on IP addresses, in addition to existing IPv4 addresses.

  You use address books to create the IP address filters. You configure address sets, each of which must not contain more than twenty IP addresses to be included in the address book.
- Domain names.

  You can specify the names of up to twenty-five active directory domains.

You can configure a filter that includes all three specifications: a range of IP addresses to include, a range of IP addresses to be excluded, and the names of one or more domains.

> **NOTE:** Filters are contextual. That is, you can use a different filter configuration for different requests. If you change the filter configuration, the new filter applies to subsequent queries exclusively. It has no bearing on prior query requests

### Caveats and Limitations

The following warnings and caveats apply to the advanced query feature:

- Before you use this feature, you must disable **active-directory-access** and **authentication-source** options under the **user-identification** hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and Web API functions are configured and committed.
- The CPU usage and resource consumption is affected by the device's reading and processing of user identity records. The impact might last several minutes.
- If user identity information is cleared from JIMS or it is missing for other reasons or delayed, the device could receive inaccurate IP address and user mapping information.
- When the device firewall authentication function pushes to JIMS entries for users successfully authenticated through captive portal, it does not update the authentication entry time-out state for the Juniper Identity Management Service server.

The following limitations apply to the advanced query feature:

- Generation of authentication entries in the identity management authentication table can be affected by a delay in the JIMS server's response time or the number of user identity records to be retrieved.
- As noted, if configuration of a filter is changed, the new filter is used only in subsequent retrievals of user identities.
- You can configure only IPv4 addresses for configuring the address ranges.

SEE ALSO


query-api | 585

batch-query | 426

filter | 487

primary | 575

secondary | 603

token-api | 639

connect-method | 464

push-to-identity-management | 580

# Configuring Advanced Query Feature for Obtaining User Identity Information from JIMS

IN THIS SECTION


- Configuring the Web API Process | 280
- Configuring the Advanced Query Feature for Obtaining User Identity Information from JIMS | 281
- Configuring Device Identity Authentication Source, and Security Policy to Match the User Identity Information Obtained from JIMS | 283


This configuration shows how to configure the advanced query feature for obtaining user identity information from Juniper Identity Management Service (JIMS) and to configure security policy to match the source identity.

This topic describes:

## Configuring the Web API Process

Configuring the Web API allows JIMS to initialize a connection to the device.

Use the following steps to configure the Web API process:

1. Configure the Web API process (webapi) username and password for the account.

```
[edit system services ]
user@host# set webapi user name  password password
```

2. Configure the Web API client address–that is, the IP address of the JIMS webserver's data port.

```
[edit system services ]
user@host# set webapi client ip address
```

> **NOTE:** Starting with Junos OS Release 18.1R1, SRX Series devices support IPv6 addresses to configure the Web API client address, in addition to existing IPv4 addresses.

3. Configure the Web API process HTTPS service port. If you enable the Web API service on the default TCP port 8080 or 8443, you must enable host inbound traffic on that port.

```
[edit system services ]
user@host# set webapi https ip address
```

4. Configure the Web API process to use the HTTPS default certificate.

```
[edit system services ]
user@host# set webapi https default-certificate
```

5. Configure the trace level for the Web API process. The supported trace levels are notice, warn, error, crit, alert, and emerg. The default value is error.

```
[edit system services ]
user@host# set webapi debug-level alert
```

### Configuring the Advanced Query Feature for Obtaining User Identity Information from JIMS

By configuring the advanced user query feature, the device can query JIMS and add identity information in the local active directory authentication table.

Use the following steps to configure the advanced query feature:

1. Configure the IP address of the primary JIMS server.

```
[edit services user-identification]
user@host# set identity-management connection primary address ip-address
```

2. Configure the client ID that the SRX Series device provides to the JIMS primary server as part of its authentication.

```
[edit services user-identification]
user@host# set identity-management connection primary client-id client-id
```

3. Configure the client secret that the device provides to the JIMS primary server as part of its authentication.

```
[edit services user-identification]
```

```
user@host# set identity-management connection primary client-secret client-secret
```

4. Configure the IP address for the secondary JIMS server.

```
[edit services user-identification]
user@host# set identity-management connection secondary address ip-address
```

5. Configure the client ID that the device provides to the JIMS secondary server as part of its authentication to it.

```
[edit services user-identification]
user@host# set identity-management connection secondary client-id client-id
```

6. Configure the client secret that the device provides to the JIMS secondary server as part of its authentication to it.

```
[edit services user-identification]
user@host# set identity-management connection primary client-secret client-secret
```

7. Configure the maximum number of user identity items that the device accepts in one batch in response to the query.

```
[edit services user-identification]
user@host# set identity-management batch-query items-per-batch items-per-batch
```

8. Configure the interval in seconds after which the device issues a query request for newly generated user identities.

```
[edit services user-identification]
user@host# set identity-management batch-query query-interval query-interval
```

9. Configure active directory domains of interest to the SRX Series device. You can specify up to twenty domain names for the filter.

```
[edit services user-identification]
user@host# set identity-management filter domain domain
```

10. Configure the address book name to include the IP filter.

```
[edit services user-identification]
user@host# set identity-management filter include-ip address-book address-book
```

11. Configure the referenced address set.

```
[edit services user-identification]
user@host# set identity-management filter include-ip address-set address-set
```

12. Configure the trace option file name.

```
[edit services user-identification]
user@host# set identity-management traceoptions file file name
```

13. Configure trace file size.

```
[edit services user-identification]
user@host# set identity-management traceoptions file file size
```

14. Configure the level of debugging output.

```
[edit services user-identification]
user@host# set identity-management traceoptions level all
```

15. Configure the trace identity management for all modules.

```
[edit services user-identification]
user@host# set identity-management traceoptions flag all
```

### Configuring Device Identity Authentication Source, and Security Policy to Match the User Identity Information Obtained from JIMS

Specify the device identity authentication source and the security policy. The device obtains the device identity information for authenticated devices from the authentication source. The device searches the device identity authentication table for a device match when traffic issuing from a user's device arrives at the device. If it finds a match, the device searches for a matching security policy. If it finds a matching security policy, the security policy's action is applied to the traffic.

Use the following steps to configure device identity authentication source:

1. Specify the device identity authentication source.

```
[edit services user-identification ]
user@host# set device-information authentication-source network-access-controller
```

2. Configure the device identity profile.

```
[edit services user-identification ]
user@host# set device-information end-user-profile profile-name profile-name domain-name domain-name
```

3. Configure the domain name to which the device belongs.

```
[edit services user-identification ]
user@host# set device-information end-user-profile profile-name profile-name attribute device-identity
   string string-value
```

Use the following steps to configure the security policy:

1. Create a source address for a security policy.

```
[edit security ]
user@host# set policies from-zone untrust to-zone trust policy  name match source-address any
```

2. Create a destination address for a security policy.

```
[edit security ]
user@host# set policies from-zone untrust to-zone trust policy  name match destination-address any
```

3. Configure the port-based application to match the policy.

```
[edit security ]
user@host# set policies from-zone untrust to-zone trust policy  name match application any
```

4. Define a username or a role (group) name that the JIMS sends to the device. Example: "jims-dom1.local\user1".

```
[edit security ]
```

```
user@host# set policies from-zone untrust to-zone trust policy name match source-identity username or
    group
```

5. Permit the packet if policy matches.

```
[edit security ]
user@host# set policies from-zone untrust to-zone trust policy name then permit
```

6. Configure the session initiation time.

```
[edit security ]
user@host# set policies from-zone untrust to-zone trust policy name then log session-init
```

7. Configure the session close time.

```
[edit security ]
user@host# set policies from-zone untrust to-zone trust policy name then log session-close
```

## Example: Configuring the Advanced Query Feature for Obtaining User Identity Information from JIMS

**IN THIS SECTION**


- Requirements | 286
- Overview | 286
- Configuration | 286
- Verification | 289


Configuring the advanced query feature on the SRX Series device to connect automatically to Juniper Identity Management Service (JIMS). This feature helps to make requests to obtain the authentication information through batch query.

JIMS is implemented to provide a robust and scalable user identification, and IP Address mapping implementation (including endpoint context and machine ID), it collects user's identities from different authentication sources on behalf of SRX Series device. For the advanced query feature, the SRX Series device works as the HTTPS client and sends HTTPS requests to JIMS on port 8082.

## Requirements

No special configuration beyond device initialization is required before configuring this feature.

## Overview

This example uses the following hardware and software components:

- JUNOS Software Release 15.1X49-D100.

Before you begin, you need the following information:

- The IP address of the JIMS server.
- The port number on the JIMS server for receiving HTTPS requests.
- The client ID from the JIMS server for advanced queries.
- The client secret from the JIMS server for advanced queries.
- The traceoptions from the JIMS server for advanced queries.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set services user-identification identity-management connection connect-method http
set services user-identification identity-management connection port 8082
set services user-identification identity-management connection primary address 12.0.0.15
set services user-identification identity-management connection primary client-id otest
set services user-identification identity-management connection primary client-secret "$ABC123"
set services user-identification identity-management connection secondary address 12.0.0.2
set services user-identification identity-management connection secondary client-id otest
set services user-identification identity-management connection secondary client-secret "$ABC123"
set services user-identification identity-management batch-query query-interval 60
set services user-identification identity-management ip-query query-delay-time 0
set services user-identification identity-management traceoptions file jimslog
set services user-identification identity-management traceoptions file size 10m
```

```
set services user-identification identity-management traceoptions level all
set services user-identification identity-management traceoptions flag all
```

**Step-by-Step Procedure**

To configure the advanced query feature on SRX Series device:

1. Configure JIMS as the authentication source for advanced query requests. The SRX Series device requires this information to contact the server.

   ```
   [edit services user-identification]
   user@host# set identity-management connection connect-method http
   ```

2. Configure the port number of the JIMS server to which the SRX Series device sends HTTPS requests.

   ```
   [edit services user-identification]
   user@host# set identity-management connection port 8082
   ```

3. Configure the primary address of the JIMS server.

   ```
   [edit services user-identification]
   user@host# set identity-management connection primary address 12.0.0.15
   ```

4. Configure the client ID and client secret to obtain access token.

   ```
   [edit services user-identification]
   user@host# set identity-management connection primary client-id otest
   user@host# set identity-management connection primary client-secret "$ABC123"
   ```

5. Configure the secondary address of the JIMS server.

   ```
   [edit services user-identification]
   user@host# set identity-management connection secondary address 12.0.0.2
   ```

6. Configure the client ID and client secret to obtain access token.

   ```
   [edit services user-identification]
   user@host# set identity-management connection secondary client-id otest
   user@host# set identity-management connection secondary client-secret "$ABC123"
   ```

7. Configure the batch query interval to periodically query JIMS for user identity information.

```
[edit services user-identification]
user@host# set identity-management batch-query query-interval 60
```

8. Configure the delay time in seconds before the SRX Series device sends the individual user query. In this example, there is no delay.

```
[edit services user-identification]
user@host# set identity-management ip-query query-delay-time 0
```

9. Configure the traceoptions.

```
[edit services user-identification]
user@host# set identity-management traceoptions file jimslog
user@host# set identity-management traceoptions file size 10m
user@host# set identity-management traceoptions level all
user@host# set identity-management traceoptions flag all
```

***Results***

From configuration mode, confirm your configuration by entering the **show services user-identification** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. To disable the ip-query use configuration **set services user-identification identity-management ip-query no-ip-query**.

```
[edit]
user@host# show services user-identification
   identity-management {
   connection {
      connect-method http;
         port 8082;
      primary {
         address 12.0.0.15;
            client-id otest;
            client-secret "$ABC123"; ## SECRET-DATA
            }
            secondary {
               address 12.0.0.2;
                  client-id otest;
               client-secret "$ABC123"; ## SECRET-DATA
```

```
            }
          }
    batch-query {
      query-interval 60;
      }
      ip-query {
        query-delay-time 0;
        }
        traceoptions {
          file jimslog size 10m;
          level all;
          flag all;
        }
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**


- Verifying the user-identification identity-management status | **289**
- Verifying the user-identification identity-management counters | **290**


Confirm that the configuration is working properly.

***Verifying the user-identification identity-management status***

**Purpose**

Verify that the JIMS server is online and which server is responding to queries from the SRX Series device.

**Action**

From operational mode, enter the **show services user-identification identity-management status** command.

```
Primary server :
    Address                     : 12.0.0.15
    Port                        : 443
    Connection method           : HTTPS
```

```
    Connection status            : Online
    Last received status message : OK (200)
    Access token                 : jjrOS4unS5d6KOTAvN8VlTsflhZBQmOm9jVsrwS
    Token expire time            : 2017-12-22 08:51:38
Secondary server :
    Address                      : 12.0.0.2
    Port                         : 443
    Connection method            : HTTPS
    Connection status            : Online
    Last received status message : OK (200)
    Access token                 : MLefNf00jG503D7H95neF1ip59JOC3jPgcl4oWQ
    Token expire time            : 2017-12-22 08:51:28
```

**Meaning**

The output provides data about the JIMS server status.

***Verifying the user-identification identity-management counters***

**Purpose**

Display counters for batch and IP queries sent to the JIMS device and responses received from the JIMS server. The batch query is displayed separately for the primary server and the secondary server, if more than one is configured.

**Action**

From operational mode, enter the **show services user-identification identity-management counters** command.

```
Primary server :
    Address                          : 12.0.0.15
    Batch query sent number          : 8
    Batch query total response number : 8
    Batch query error response number : 0
    Batch query last response time    : 2017-12-22 01:04:34
    IP query sent number             : 4
    IP query total response number    : 4
    IP query error response number    : 0
    IP query last response time       : 2017-12-22 01:02:25
Secondary server :
    Address                          : 12.0.0.2
    Batch query sent number          : 0
    Batch query total response number : 0
    Batch query error response number : 0
    Batch query last response time    : 0
```

```
IP query sent number              : 0
IP query total response number    : 0
IP query error response number    : 0
IP query last response time       : 0
```

**Meaning**

The output provides the batch and IP queries data.

## Example: Configuring Filter for Advanced Query Feature

**IN THIS SECTION**


- Requirements | **291**
- Overview | **291**
- Configuration | **292**
- Verification | **294**


An SRX Series device supports IP filters and domain filters when querying Juniper Identity Management Service (JIMS). The advanced query feature provides an optional filter function to receive the user information in response to queries.

This example shows how to configure the filters for obtaining the user information.

### Requirements

Before you begin:

- Configure the advanced query feature. See "Configuring Advanced Query Feature for Obtaining User Identity Information from JIMS" on page 280.

### Overview

You can configure filters to query JIMS server at a more granular level to obtain user identity information based on IP addresses. You can set filters to include the IP address ranges, which SRX Series devices

require or exclude the IP address ranges that they do not require when collecting the user identity information. You can also filter domains.

A filter can include and exclude up to twenty IP address ranges. Therefore, an address set that contains more than twenty address ranges causes the filter configuration to fail. To specify the ranges, specify the name of a predefined address set which includes them, and also which is included in an existing address book.

A domain can include up to 20 domain names for a filter.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

In this example, define an address book, and specify the security address for the address book. Specify an IP address with a prefix. Define an address set name and specify the address. Include and exclude the IP addresses in the address book. Add the address set to include and exclude the IP addresses. Add a domain name to filter the domain.

```
set security address-book mybook address addr1 192.0.2.0/24
set security address-book mybook address-set myset address addr1
set services user-identification identity-management filter include-ip address-book mybook
set services user-identification identity-management filter include-ip address-set myset
set security address-book mybook2 address addr2 198.51.100.0/24
set security address-book mybook2 address-set myset2 address addr2
set services user-identification identity-management filter exclude-ip address-book mybook2
set services user-identification identity-management filter exclude-ip address-set myset2
set services user-identification identity-management filter domain host.example.com
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a filter for advanced query feature:

1. Define an address book name, specify security address for the address book, and add an IPv4 address with a prefix.

   ```
   [edit ]
   user@host# set security address-book mybook address addr1 192.0.2.0/24
   user@host# set security address-book mybook2 address addr2 198.51.100.0/24
   ```

2. Specify an address set name and specify the address.

```
[edit ]
user@host# set security address-book mybook address-set myset address addr1
user@host# set security address-book mybook2 address-set myset2 address addr2
```

3. Configure the address book to include and exclude the IP address.

```
[edit ]
user@host# set services user-identification identity-management filter include-ip address-book mybook
user@host# set services user-identification identity-management filter exclude-ip address-book mybook2
```

4. Define the address set to include or exclude the IP address.

```
[edit ]
user@host# set services user-identification identity-management filter include-ip address-set myset
user@host# set services user-identification identity-management filter exclude-ip address-set myset2
```

5. Specify a domain name to filter the domain.

```
[edit ]
user@host# set services user-identification identity-management filter domain host.example.com
```

***Results***

From configuration mode, confirm your configuration by entering the **show services user-identification** and **show security address-book** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services user-identification
identity-management {
    filter {
        domain {
            host.example.com;
        }
        include-ip {
            address-book mybook;
            address-set myset;
        }
```

```
        exclude-ip {
            address-book mybook2;
            address-set myset2;
        }
    }
}
```

```
[edit]
user@host# show security address-book
mybook {
    address addr1 192.0.2.0/24;
    address-set myset {
        address addr1;
    }
}
mybook2 {
    address addr2 198.51.100.0/24;
    address-set myset2 {
        address addr2;
    }
}
```

## Verification

**IN THIS SECTION**


- Verifying Filter for Advanced Query Feature | **294**


### *Verifying Filter for Advanced Query Feature*

**Purpose**

Verify that the authentication table displays the user information that you want to receive in response to queries.

**Action**

From operational mode, enter **show services user-identification authentication-table authentication-source all** command.

```
show services user-identification authentication-table authentication-source
all
```

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: host.example.com
Total entries: 10
Source IP       Username       groups(Ref by policy)          state
192.0.2.10      jasonlee                                      Valid
192.0.2.9       jasonlee                                      Valid
192.0.2.8       jasonlee                                      Valid
192.0.2.7       jasonlee                                      Valid
192.0.2.6       jasonlee                                      Valid
192.0.2.5       jasonlee                                      Valid
192.0.2.4       jasonlee                                      Valid
192.0.2.3       jasonlee                                      Valid
192.0.2.2       jasonlee                                      Valid
192.0.2.1       jasonlee                                      Valid

node1:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: host.example.com
Total entries: 10
Source IP       Username       groups(Ref by policy)          state
192.0.2.10      jasonlee                                      Valid
192.0.2.9       jasonlee                                      Valid
192.0.2.8       jasonlee                                      Valid
192.0.2.7       jasonlee                                      Valid
192.0.2.6       jasonlee                                      Valid
192.0.2.5       jasonlee                                      Valid
192.0.2.4       jasonlee                                      Valid
192.0.2.3       jasonlee                                      Valid
192.0.2.2       jasonlee                                      Valid
192.0.2.1       jasonlee                                      Valid
```

**Meaning**

The output displays the user information in response to queries.

6

CHAPTER

# User Authentication and Enforcement with Clearpass


Integrated ClearPass Authentication and Enforcement Overview | **299**

Configure Integrated ClearPass Authentication and Enforcement | **306**

Enforce Security Policies using ClearPass | **334**

Filter and Transmit Threat and Attack Logs to ClearPass | **366**

Configure ClearPass and JIMS at the Same Time | **376**

# Integrated ClearPass Authentication and Enforcement Overview

IN THIS SECTION


- Understanding the Integrated ClearPass Authentication and Enforcement Feature | 299
- Understanding the Invalid Authentication Table Entry Timeout Setting | 302


The SRX Series and NFX Series devices associate with ClearPass to control the user access from the user level based on their usernames or by the groups that they belong to, not the IP address of the device.

## Understanding the Integrated ClearPass Authentication and Enforcement Feature

IN THIS SECTION


- Why You Need to Protect Your Environment With the Integrated ClearPass Authentication and Enforcement Feature | 300
- How the Integrated ClearPass Authentication and Enforcement Feature Can Protect Your Network Environment | 300

This topic introduces the integrated ClearPass authentication and enforcement feature in which the device and Aruba ClearPass collaborate to protect your network resources by enforcing security at the user identity level and controlling user access to the Internet. The ClearPass Policy Manager (CPPM) can authenticate users across wired, wireless, and VPN infrastructures. The integrated ClearPass feature allows the CPPM and the device to collaborate in multiple environments in which they are deployed together.

### Why You Need to Protect Your Environment With the Integrated ClearPass Authentication and Enforcement Feature

The proliferation of mobile devices and cloud services and securing them has become a fundamental strategic part of enterprise cybersecurity. Use of company smartphones poses one of the biggest IT security risks to businesses. The integrated ClearPass feature protects against malicious intrusions introduced through use of mobile devices and multiple concurrently connected devices.

In a work environment that supports mobile devices, knowing the identity of the user whose device is associated with an attack or threat provides IT administrators with improved advantage in identifying the source of the attack and stemming future potential attacks that follow the same strategy.

Attackers can gain access to nearby company-owned mobile devices and install malware on them that they can then use to capture data at any time. Whether reconnaissance or malicious, attacks against network resources are commonplace in today's computing environment. Attackers can launch information-gathering ventures, stop business activity, and steal sensitive corporate data.

Today's network environments are more open to attacks of various kinds because they support *anywhere*, *anytime*, *any device* access, to a greater or lesser degree, and they allow a user to use multiple concurrently network-connected devices.

The integrated ClearPass authentication and enforcement feature can protect you against attacks and intrusions by allowing you to configure security policies that identify users by their usernames or by the groups that they belong to. It also identifies threats and attacks perpetrated against your network environment and provides this information to the CPPM. As administrator of the CPPM, you can better align your security enforcement to protect against possible future attacks of the same kind. If a user is logged in to the network with more than one device, you can keep track of their activity based on their identity, not only by their devices, and you can more easily control their network access and any egregious activity on their behalf, whether intended or not.

### How the Integrated ClearPass Authentication and Enforcement Feature Can Protect Your Network Environment

The integrated ClearPass authentication and enforcement feature gives you granular control at the user level, not the device's IP address, over user access to protected resources and the Internet. As administrator of the device, you can now specify in the source-identity parameter of *identity-aware* security policies a username or a role (group) name that the CPPM posts to the device. You are no longer restricted to relying

solely on the IP address of the device as a means of identifying the user. Honing in on the user of the device, rather than only the device, enhances your control over security enforcement.

In addition to providing the SRX Series device with authenticated user information, the CPPM can map a device type to a role and assign users to that role. It can then send that role mapping to the SRX Series device. This capability allows you to control through security policies a user's access to resources when they are using a *specific type of device*.

For example, suppose that the administrator of the CPPM configured a role called marketing-company-device and mapped to that role both company devices and members of the Marketing department. As administrator of the device, you could specify that role in a security policy as if it were a group. The security policy would then apply to all users mapped to the role, inherently controlling their network activity when they use that type of device type.

The integrated ClearPass feature delivers the protection of the SCREENS, IDP and UTM features to defend your network against a wide range of attack strategies. In addition to protecting the company's network resources, the device can make available to the CPPM log records generated by these protective security features in response to attack or attack threats. Knowing about threats and specific attacks that have already occurred can help IT departments to identify noncompliant systems and exposed areas of the network. With this information, they can harden their security by enforcing device compliance and strengthening protection of their resources.

SRX Series security policies protect the company's resources and enforce access control at a fine-grain level, taking advantage of the user authentication and identity information sent to the device from the CPPM. The CPPM acts as the authentication source. It uses its own internal RADIUS server to authenticate users. It can also rely on an external authentication source to perform the authentication for it, such as an external RADIUS server or Active Directory.

The CPPM authentication is triggered by requests from NAS devices such as switches and access controllers. The CPPM uses the XML portion of the RESTful Web services that the device exposes to it to send in POST request messages to the device authenticated user identity and device posture information.

The device and Aruba ClearPass simplify the complex and complicated security tasks required to safeguard company resources and enforce Internet access policy for mobile devices. This security is essential in a network environment that supports the mobile experience and that gives the user latitude to use a wide range of devices, including their own systems, smartphones, and tablets.

Starting with Junos OS Release 15.1X49-D130, the SRX Series device supports the use of IPv6 addresses associated with source identities in security policies. If IPv4 or IPv6 entry exists, policies matching that entry are applied to the traffic and access is allowed or denied.

# Understanding the Invalid Authentication Table Entry Timeout Setting

IN THIS SECTION


- Timeout Setting for Invalid Authentication Entries | 302
- How the Invalid Authentication Entry Timeout Works for Windows Active Directory | 303
- How the Invalid Authentication Entry Timeout Works for SRX Series and NFX Series Aruba ClearPass | 304


## Timeout Setting for Invalid Authentication Entries

Starting in Junos OS Release 15.1X49-D100, for SRX Series devices and vSRX, you can protect invalid user authentication entries in an authentication table from expiring before the user can be validated by configuring a timeout setting that is specific to invalid entries. The invalid authentication entry timeout setting is separate from the common authentication entry timeout setting that is applied to valid entries.

Authentication entries in both the Windows Active Directory authentication table and the ClearPass authentication table contain a timeout value after which the entry expires. Prior to introduction of this feature, a single, common timeout setting was applied to valid and invalid authentication entries. That is, if an invalid authentication entry was created in either of these tables, the current setting of the common timeout for the table—which applied to all of the table's entries—was applied to it.

For both the Active Directory authentication table and the ClearPass authentication table, the invalid entry could expire before the user's identity could be validated. Here is what could cause that event to occur in each case:

- Windows Active Directory uses a mechanism to probe an unauthenticated user's device for user identity authentication information based on the IP address of the device. It is not uncommon for Windows to trigger a WMI probe that fails because it occurs before the user logs in. After an unsuccessful probe, the system generates an entry in the authentication table with an INVALID state for the IP address of the device. If you configured a value for the invalid timeout setting, that timeout is applied to the entry. If you did not configure a value for the invalid entry timeout setting, then its default timeout of 30 minutes is applied.

  The invalid authentication entry timeout setting is separate from the common authentication entry timeout setting that is applied to valid entries.

  Starting in Junos OS Release 17.4R1, the integrated user firewall supports IPv6 device addresses in the Windows Active Directory authentication table. Prior to Junos OS Release 17.4R1, only IPv4 addresses were supported.

- For the ClearPass feature, if an unauthenticated user attempts to join the network and the IP address of the user's device is not found—that is, it is not in the Packet Forwarding Engine—the device queries Aruba ClearPass for the user's information. If the query is unsuccessful, the system generates an INVALID authentication entry for the user. If you configured a value for the invalid timeout setting, that timeout is applied to the entry. If you did not configure the invalid entry timeout, then its default timeout of 30 minutes is applied to the new entry.

  **NOTE:** The invalid entry timeout is also applied to entries whose state is changed from valid or pending to INVALID.

You configure the timeout setting to be applied to invalid authentication entries in the Windows Active Directory authentication table and the ClearPass authentication table separately. If you do not configure a timeout setting, the invalid authentication entry timeout default value of 30 minutes is applied. The application and effect of the timeout value is determined differently for these authentication sources.

### How the Invalid Authentication Entry Timeout Works for Windows Active Directory

Use the following command to configure the invalid authentication entry timeout setting for entries in the Windows Active Directory authentication table. In this example, the invalid authentication entry timeout value is set to 40 minutes. That timeout value is applied to new invalid entries.

```
user@host# set services user-identification active-directory-access invalid-authentication-entry-timeout 40
```

The new timeout value is also applied to existing invalid entries but within the context of the current timeout value assigned to them and the timeout state. Suppose that the authentication table contains existing invalid entries to which an invalid authentication entry timeout setting or the default was previously applied. In this case, the new invalid entry timeout setting has effect on the timeout for these entries, but in a different way. For these entries, the original timeout setting—the time that has expired since the original timeout value was applied–and the new timeout setting collude to produce the resulting timeout value that is applied to the existing entry.

As Table 8 on page 192 shows, in some cases the resulting timeout is extended, in some cases it is shortened, and in some cases it causes the original timeout to expire and the invalid authentication entry to which is applies to be deleted.

**Table 18: How New Invalid Authentication Entry Timeout Settings Affect Timeout Settings for Existing Invalid Entries in the Active Directory Authentication Table**

| Original Invalid Entry Timeout Setting for Existing Entry | Elapse Time | New Invalid Entry Timeout Configuration Setting | Resulting Timeout Setting for Existing Invalid Entry |
|---|---|---|---|
| 20 minutes | 5 minutes | 50 minutes | 45 minutes |
| 50 minutes | 10 minutes | 20 minutes | 10 minutes |
| 50 minutes | 40 minutes | 20 minutes | Timeout expired and entry is removed from the authentication table |
| 40 minutes | 20 minutes | 0 | 0 |

**NOTE:** Just as the new invalid timeout entry is imposed on that of old invalid entries, producing various and unique results, a new invalid entry is subject to the same rules and effects when the invalid entry timeout value is changed.

### How the Invalid Authentication Entry Timeout Works for SRX Series and NFX Series Aruba ClearPass

Use the following command to configure the invalid authentication entry timeout for entries in the ClearPass authentication table. In this example, invalid authentication entries in the ClearPass authentication table expires 22 minutes after they are created.

```
user@host# set services user-identification authentication-source aruba-clearpass
    invalid-authentication-entry-timeout 22
```

- When you initially configure the invalid authentication entry timeout value for ClearPass, it is applied to any invalid authentication entries that are generated *after* it was configured. However, all existing invalid authentication entries retain the default timeout of 30 minutes.
- If you do not configure the invalid authentication entry timeout setting, the default timeout of 30 minutes is applied to all invalid authentication entries.

  If you configure the invalid authentication entry timeout setting and delete it later, the default value is applied to new invalid authentication entries generated after the deletion. However, any existing invalid authentication entries to which a configured value had been applied previously retain that value.

- If you change the setting for the invalid authentication entry timeout value, the new value is applied to all invalid authentication entries that were created *after* the value was changed. However, all existing invalid authentication entries retain the former invalid authentication entry timeout setting applied to them. Those entries to which the default value of 30 minutes had been applied previously retain that setting.
- When the pending or valid state of an entry is changed to invalid, the invalid authentication entry timeout setting is applied to it.

  When the state of an invalid authentication entry is changed to pending or valid, the invalid authentication entry timeout setting is no longer applicable to it. The timeout value set for the common authentication entry timeout is applied to it

Table 9 on page 193 shows how a new invalid entry timeout value affects new and existing invalid entries.

**Table 19: How New Invalid Authentication Entry Timeout Settings Affect Timeout Settings for Invalid Entries in the ClearPass Authentication Table**

| Invalid Entry Timeout Setting | Intial Invalid Entry Timeout Setting | Elapse Time | New Invalid Entry Timeout Configuration Setting | Final Timeout Setting for Existing Invalid Entry |
|---|---|---|---|---|
| New invalid authentication entry | | | 50 | 50 |
| Existing invalid entry timeout | 20 | 5 | 50 | 15 |
| Existing invalid entry timeout | 0 | 40 | 20 | 0 |
| Existing invalid entry timeout | 40 | 20 | 0 | 20 |

## RELATED DOCUMENTATION

invalid-authentication-entry-timeout | 526

firewall-authentication-forced-timeout | 498

# Configure Integrated ClearPass Authentication and Enforcement

IN THIS SECTION


- Understanding How ClearPass Initiates a Session and Communicates User Authentication Information Using the Web API | 306
- Example: Configuring the SRX Series Integrated ClearPass Feature to Allow the Device to Receive User Authentication Data from ClearPass | 310
- Understanding the Integrated ClearPass Authentication and Enforcement User Query Function | 321
- Example: Configuring the Integrated ClearPass Authentication and Enforcement User Query Function | 324


SRX Series and NFX Series devices collaborate with ClearPass to control the user access from the user level by their usernames or by the groups that they belong to, not the IP address of the device. The device Web API acts as an HTTP server and sends user identity information from ClearPass to the device for authentication. Also, the user query function helps to query an individual user for user identity information.

## Understanding How ClearPass Initiates a Session and Communicates User Authentication Information Using the Web API

IN THIS SECTION


- Web API | 307
- ClearPass Authentication Table | 307
- Using HTTPS or HTTP for the Connection Protocol Between ClearPass and the Device | 308
- Ensuring the Integrity of Data Sent from ClearPass to the Device | 308
- Data Size Restrictions and Other Constraints | 308
- Posture States and the Posture Group | 309

The integrated ClearPass authentication and enforcement feature enables the SRX Series or NFX Series device and Aruba ClearPass to collaborate in protecting your company's resources by enforcing security at the user identity level in environments in which they are deployed together. The ClearPass Policy Manager (CPPM) can authenticate users across wired, wireless, and VPN infrastructures and post that information to the device, which, in turn, uses it to authenticate users requesting access to your protected resources and to the internet. The device can provide the CPPM with threat and attack logs associated users' devices so that you can better harden your security at the ClearPass end.

### Web API

The device exposes to the CPPM its Web API daemon (webapi) interface that enables the CPPM to integrate with it and efficiently send authenticated user identity information to the device. The Web API daemon acts as an HTTP server in that it implements part of the RESTful Web services that supports concurrent HTTP and HTTPS requests. In this relationship, the CPPM is the client. The Web API daemon is restricted to processing only HTTP/HTTPS requests. Any other type of request it receives generates an error message.

**WARNING:** If you are deploying the integrated ClearPass Web API function and Web management at the same time, you must ensure that they use different HTTP or HTTPS service ports.

However, for security considerations, we recommend that you use HTTPS instead of HTTP. HTTP is supported primarily for debugging purposes.

The Web API daemon runs on the master Routing Engine in a chassis cluster environment. After an Chassis Cluster switchover, the daemon will start automatically on the new master Routing Engine. It has no effect on the Packet Forwarding Engine.

Starting with Junos OS Release 15.1X49-D130, you can configure the IPv6 address for Web API function to allow the ClearPass to initiate and establish a secure connection. Web API supports the IPv6 user entries obtained from CPPM. Prior to Junos OS Release 15.1X49-D130, only IPv4 address was supported.

### ClearPass Authentication Table

After the device receives information posted to it from the CPPM, the device extracts the user authentication and identity information, analyzes it, and distributes it to the appropriate processes for handling. The device creates a ClearPass authentication table on the Packet Forwarding Engine side to hold this user information. When the device receives the information sent to it from ClearPass, the device generates entries in the ClearPass authentication table for the authenticated users. When the device receives an access request from a user, it can check its ClearPass authentication table to verify that the user is authenticated, and then apply the security policy that matches the traffic from the user.

Starting with Junos OS Release 15.1X49-D130, device can receive the IPv6 addresses from CPPM, and the ClearPass authentication table supports IPv6 addresses.

### Using HTTPS or HTTP for the Connection Protocol Between ClearPass and the Device

When you configure the Web API, you specify a certificate key if you are using HTTPS as the connection protocol. To ensure security, the HTTPS default certificate key size is 2048 bytes. If you do not specify a certificate size, the default size is assumed. There are three methods that you can use to specify a certificate:

- Default certificate
- Certificate generated by PKI
- Custom certificate and certificate key

  The SRX Series Web API supports only the Privacy-Enhanced Mail (PEM) format for the certificate and certificate key configuration.

If you enable the Web API on the default ports—HTTP (8080) or HTTPS (8443)—you must enable host inbound traffic on the ports. If you enable it on any other TCP port, you must enable host inbound traffic specifying the parameter **any-service**. For example:

```
user@host# set security zones security-zone trust host-inbound-traffic system-services any-service
```

### Ensuring the Integrity of Data Sent from ClearPass to the Device

The following requirements ensure that the data sent from the CPPM is not compromised:

- The Web API implementation is restricted to processing only HTTP/HTTPS POST requests. Any other type of request that it receives generates an error message.
- The Web API daemon analyzes and processes HTTP/HTTPS requests from only the following dedicated URL:

  `/api/userfw/v1/post-entry`

- The HTTP/HTTPS content that the CPPM posts to the device must be consistently formatted correctly. The correct XML format indicates a lack of compromise, and it ensures that user identity information is not lost.

### Data Size Restrictions and Other Constraints

The following data size restrictions and limitations apply to the CPPM:

- The CPPM must control the size of the data that it posts. Otherwise the Web API daemon is unable to process it. Presently the Web API can process a maximum of 2 megabytes of data.

- The following limitations apply to XML data for role and device posture information. The Web API daemon discards XML data sent to it that exceeds these amounts (that is, the overflow data):
  - The SRX Series device can process a maximum of 209 roles.
  - The SRX Series device supports only one type of posture with six possible posture tokens, or values. Identity information for an individual user can have only one posture token.

> **NOTE:** The CPPM checks the health and posture of a device and it can send that information to the SRX Series or NFX Series device as part of the user information that it posts. You cannot define posture on the SRX Series or NFX Series device. Also, the SRX Series or NFX Series device does not check posture information that it receives.

### Posture States and the Posture Group

User, role, and posture token fields are distinct in the context of the CPPM. Each set of user identity information contains user and role (group) identity and a posture token. Because the SRX Series or NFX Series device supports only user and role (group) fields, the posture token value is mapped to a role by adding the prefix **posture–**. You can then use that role in a security policy as a group and that policy will be applied to all traffic that matches the policy.

The predefined posture identity states are:

- posture-healthy (HEALTHY)
- posture-checkup (CHECKUP)
- posture-transition (TRANSITION)
- posture-quarantine (QUARANTINE)
- posture-infected (INFECTED)
- posture-unknown (UNKNOWN)

SEE ALSO

Example: Enforcing SRX Series Security Policies Using Aruba ClearPass as the Authentication Source | 345

# Example: Configuring the SRX Series Integrated ClearPass Feature to Allow the Device to Receive User Authentication Data from ClearPass

IN THIS SECTION


- Requirements | 310
- Overview | 311
- Configuration | 315


The SRX Series device and the ClearPass Policy Manager (CPPM) collaborate to control access to your protected resources and to the Internet. To carry this out, the SRX Series device must authenticate users in conjunction with applying security policies that match their requests. For the integrated ClearPass authentication and enforcement feature, the SRX Series device relies on ClearPass as its authentication source.

The Web API function, which this example covers, exposes to the CPPM an API that enables it to initiate a secure connection with the SRX Series device. The CPPM uses this connection to post user authentication information to the SRX Series device. In their relationship, the SRX Series device acts as an HTTPS server for the CPPM client.

## Requirements

This section defines the software and hardware requirements for the topology for this example. See Figure 19 on page 315 for the topology design.

The hardware and software components are:

- Aruba ClearPass Policy Manager (CPPM). The CPPM is configured to use its local authentication source to authenticate users.

  **NOTE:** It is assumed that the CPPM is configured to provide the SRX Series device with user authentication and identity information, including the username, a list of the names of any groups that the user belongs to, the IP addresses of the devices used, and the device posture token.

- SRX Series device running Junos OS that includes the integrated ClearPass feature.
- A server farm composed of six servers, all in the servers-zone:

  - marketing-server-protected (203.0.113.23 )
  - human-resources-server (203.0.113.25 )
  - accounting-server (203.0.113.72)
  - public-server (192.0.2.96)
  - corporate-server (203.0.113.71)
  - sales-server (203.0.113.81)
- AC 7010 Aruba Cloud Services Controller running ArubaOS.
- Aruba AP wireless access controller running ArubaOS.

  The Aruba AP is connected to the AC7010.

  Wireless users connect to the CPPM through the Aruba AP.
- Juniper Networks EX4300 switch used as the wired 802.1 access device.

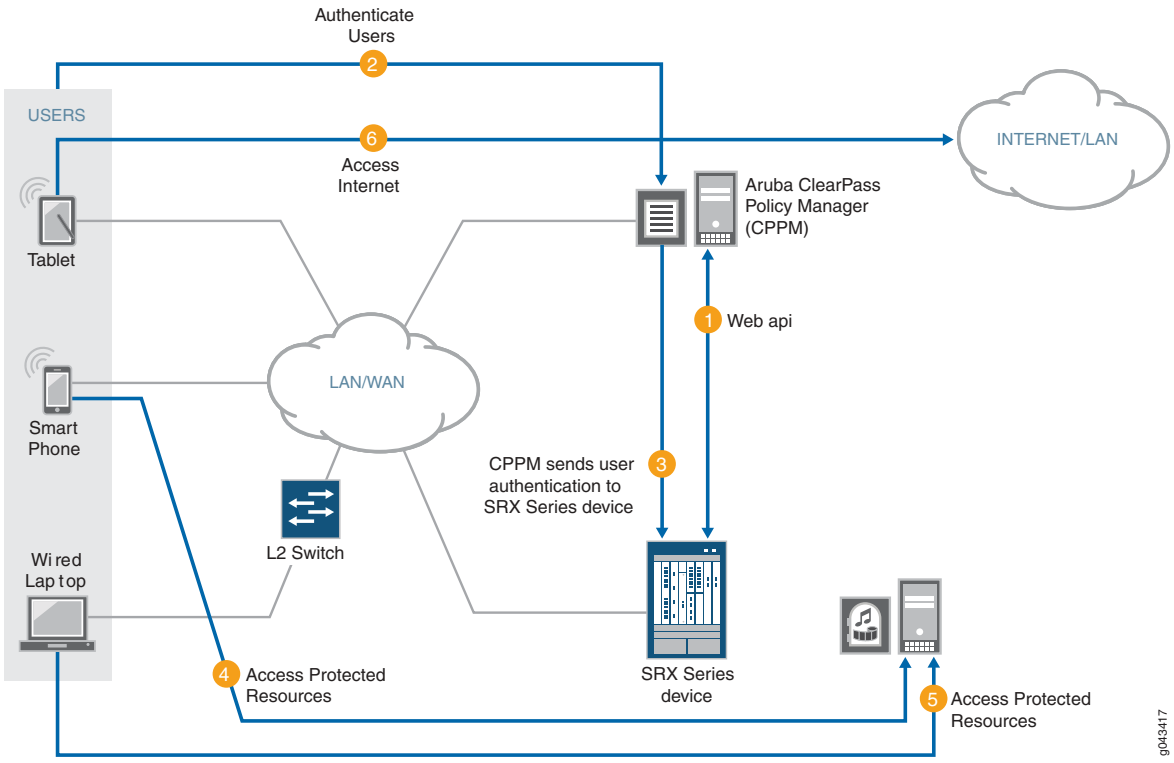
  Wired users connect to the CPPM using the EX4300 switch.
- Six end-user systems:
  - Three wired network-connected PCs running Microsoft OS
  - Two BYOD devices that access the network through the Aruba AP access device
  - One wireless laptop running Microsoft OS

### Overview

You can configure identity-aware security policies on the SRX Series device to control a user's access to resources based on username or group name, not the IP address of the device. For this feature, the SRX Series device relies on the CPPM for user authentication. The SRX Series device exposes to ClearPass its Web API (webapi) to allow the CPPM to integrate with it. The CCPM posts user authentication information efficiently to the SRX Series device across the connection. You must configure the Web API function to allow the CPPM to initiate and establish a secure connection. There is no separate Routing Engine process required on the SRX Series device to establish a connection between the SRX Series device and the CPPM.

Figure 18 on page 312 illustrates the communication cycle between the SRX Series device and the CPPM, including user authentication.

Figure 18: ClearPass and SRX Series Device Communication and User Authentication Process




As depicted, the following activity takes place:

1. The CPPM initiates a secure connection with the SRX Series device using Web API.
2. Three users join the network and are authenticated by the CPPM.
   - A tablet user joins the network across the corporate WAN.
   - A smartphone user joins the network across the corporate WAN.
   - A wireless laptop user joins the network from a wired laptop connected to a Layer 2 switch that is connected to the corporate LAN.
3. The CPPM sends the user authentication and identity information for the users who are logged in to the network to the SRX Series device in POST request messages using the Web API.

   When traffic from a user arrives at the SRX Series device, the SRX Series device:
   - Identifies a security policy that the traffic matches.
   - Locates an authentication entry for the user in the ClearPass authentication table.
   - Applies the security policy to the traffic after authenticating the user.
4. Traffic from the smartphone user who is requesting access to an internal, protected resource arrives at the SRX Series device. Because all of the conditions identified in Step 3 are met and the security policy permits it, the SRX Series device allows the user connection to the protected resource.

5. Traffic from the wired laptop user who is requesting access to a protected resource arrives at the SRX Series device. Because all of the conditions identified in Step 3 are met and the security policy permits it, the SRX Series device allows the user connection to the resource.
6. Traffic from the tablet user who is requesting access to the Internet arrives at the SRX Series device. Because all of the conditions identified in Step 3 are met and the security policy permits it, the SRX Series device allows the user connection to the Internet.

The Web API daemon is not enabled by default for security reasons. When you start up the Web API daemon, by default it opens either the HTTP (8080) or the HTTPS (8443) service port. You must ensure that one of these ports is configured, depending on which version of the HTTP protocol you want to use. We recommend that you use HTTPS for security reasons. Opening these ports makes the system more vulnerable to service attacks. To protect against service attacks that might use these ports, the Web API daemon will start up only after you enable it.

The Web API is a RESTful Web services implementation. However, it does not fully support the RESTful Web services. Rather, it acts as an HTTP or HTTPS server that responds to requests from the ClearPass client.

> **NOTE:** The Web API connection is initialized by the CPPM using the HTTP service port (8080) or HTTPS service port (8443). For ClearPass to be able to post messages, you must enable and configure the Web API daemon.

To mitigate abuse and protect against data tampering, the Web API daemon:

- Requires ClearPass client authentication by HTTP or HTTPS basic user account authentication.
- Allows data to be posted to it only from the IP address configured as the client source. That is, it allows HTTP or HTTPS POST requests only from the ClearPass client IP address, which in this example is 192.0.2.199.
- Requires that posted content conforms to the established XML data format. When it processes the data, the Web API daemon ensures that the correct data format was used.

> **NOTE:** Note that if you deploy Web management and the SRX Series device together, they must run on different HTTP or HTTPS service ports.

See "Understanding How ClearPass Initiates a Session and Communicates User Authentication Information Using the Web API" on page 306 for further information on how this feature protects against data tampering.

The SRX Series UserID daemon processes the user authentication and identity information and synchronizes it to the ClearPass authentication table on the Packet Forwarding Engine. The SRX Series device creates the ClearPass authentication table to be used for information received only from the CPPM. The ClearPass

authentication table does not contain user authentication information from other authentication sources. The SRX Series device checks the ClearPass authentication table to authenticate users attempting to access protected network resources on the Internet using wired or wireless devices and local network resources.

For the CPPM to connect to the SRX Series device and post authentication information, it must be certified using HTTPS authentication. The Web API daemon supports three methods that can be used to refer to an HTTPS certificate: a default certificate, a PKI local certificate, and a customized certificate implemented through the certificate and certificate-key configuration statements. These certificate methods are mutually exclusive.

This example uses HTTPS for the connection between the CPPM and the SRX Series device. To ensure security, the integrated ClearPass feature default certificate key size is 2084 bits.

Whether you use any method—the default certificate, a PKI-generated certificate, or a custom certificate—for security reasons, you must ensure that the certificate size is 2084 bits or greater.

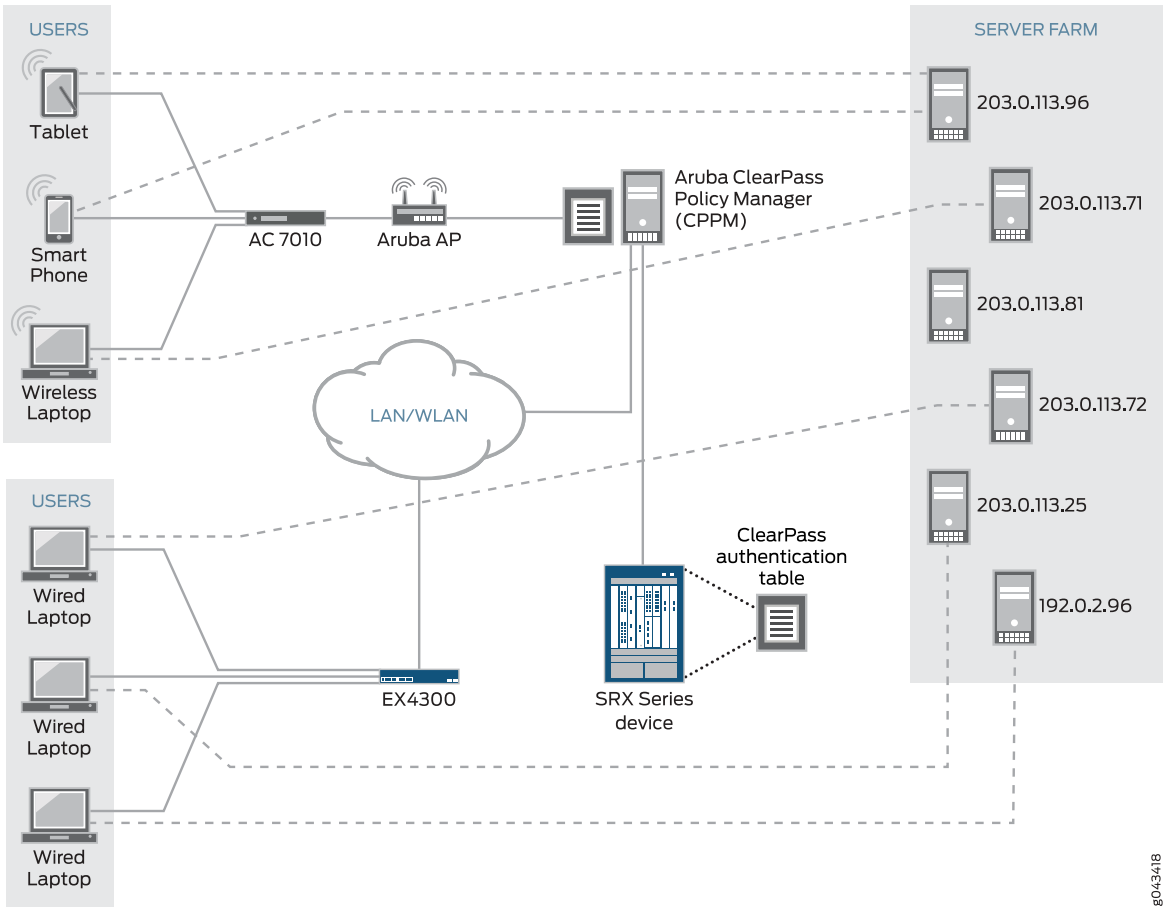
The following example shows how to generate a certificate and key using PKI:

```
user@host>request security pki generate-key-pair certificate-id aruba size 2048
user@host>request security pki local-certificate generate-self-signed certificate-id aruba domain-name
    mycompany.net email jxchan@mycompany.net ip-address 192.51.100.21 subject “CN=John Doe,OU=Sales
    ,O=mycompany.net ,L=MyCity ,ST=CA,C=US"
```

***Topology***

Figure 19 on page 315 shows the topology used for the integrated ClearPass deployment examples.

Figure 19: Integrated ClearPass Authentication and Enforcement Deployment Topology




## Configuration

**IN THIS SECTION**


- Configuring the SRX Series Web API Daemon | **316**
- Configuring the ClearPass Authentication Table Entry Timeout and Priority | **319**


This section covers how to enable and configure the SRX Series Web API.

**NOTE:** You must enable the Web API. It is not enabled by default.

**CLI Quick Configuration**

To quickly configure this example, copy the following statements, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the statements into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set system services webapi user sunny password i4%rgd
set system services webapi client 192.0.2.199
set system services webapi https port 8443
set system services webapi https pki-local-certificate aruba
set system services webapi debug-level alert
set interfaces ge-0/0/3.4 vlan-id 340 family inet address 192.51.100.21
set security zones security-zone trust interfaces ge-0/0/3.4 host-inbound-traffic system-services webapi-ssl
set security user-identification authentication-source aruba-clearpass priority 110
set security user-identification authentication-source local-authentication-table priority 120
set security user-identification authentication-source active-directory-authentication-table priority 125
set security user-identification authentication-source firewall-authentication priority 150
set security user-identification authentication-source unified-access-control priority 200
```

***Configuring the SRX Series Web API Daemon***

**Step-by-Step Procedure**

Configuring the Web API allows the CPPM to initialize a connection to the SRX Series device. No separate connection configuration is required.

It is assumed that the CPPM is configured to provide the SRX Series device with authenticated user identity information, including the username, the names of any groups that the user belongs to, the IP addresses of the devices used, and a posture token.

Note that the CPPM might have configured role mappings that map users or user groups to device types. If the CPPM forwards the role mapping information to the SRX Series device, the SRX Series device treats the role mappings as groups. The SRX Series device does not distinguish them from other groups.

**Step-by-Step Procedure**

To configure the Web API daemon:

1. Configure the Web API daemon (webapi) username and password for the account.

   This information is used for the HTTPS certification request.

   ```
   [edit system services]
   user@host# set webapi user sunny password i4%rgd
   ```

2. Configure the Web API client address–that is, the IP address of the ClearPass webserver's data port.

   The SRX Series device accepts information from this address only.

> **NOTE:** The ClearPass webserver data port whose address is configured here is the same one that is used for the user query function, if you configure that function.

```
[edit system services]
user@host# set webapi client 192.0.2.199
```

> **NOTE:** Starting with Junos OS Release 15.1X49-D130, SRX Series device supports IPv6 addresses to configure the Web API client address. Prior to Junos OS Release 15.1X49-D130, only IPv4 addresses were supported.

3. Configure the Web API daemon HTTPS service port.

   If you enable the Web API service on the default TCP port 8080 or 8443, you must enable host inbound traffic on that port.

   In this example, the secure version of the Web API service is used (webapi-ssl), so you must configure the HTTPS service port, 8443.

   ```
   [edit system services]
   user@host# set webapi https port 8443
   ```

4. Configure the Web API daemon to use the HTTPS default certificate.

   ```
   [edit system services]
   user@host# set webapi https pki-local-certificate aruba
   ```

5. Configure the trace level for the Web API daemon.

   The supported trace levels are notice, warn, error, crit, alert, and emerg. The default value is error.

   ```
   [edit system services]
   user@host# webapi debug-level alert
   ```

6. Configure the interface to use for host inbound traffic from the CPPM.

   ```
   user@host# set interfaces ge-0/0/3.4 vlan-id 340 family inet address 192.51.100.21
   ```

7. Enable the Web API service over HTTPS host inbound traffic on TCP port 8443.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/3.4 host-inbound-traffic system-services webapi-ssl
```

**Results**

From configuration mode, confirm your Web API configuration by entering the **show system services webapi** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user {
    sunny;
    password "$ABC123"; ## SECRET-DATA
}
client {
    192.0.2.199;
}
https {
    port 8443;
    pki-local-certificate aruba;
}
debug-level {
    alert;
}
```

From configuration mode, confirm the configuration for the interface used for host inbound traffic from the CPPM by entering the **show interfaces ge-0/0/3.4** command. If the output does not display the intended configuration, repeat the verification process in this example to correct it.

```
vlan-id 340;
family inet {
    address 192.51.100.21/32;
}
```

From configuration mode, confirm your security zone configuration that allows host-inbound traffic from the CPPM using the secure Web API service (web-api-ssl) by entering the **show security zones security-zone trust** command. If the output does not display the intended configuration, repeat the verification process in this example to correct it.

```
interfaces {
    ge-0/0/3.4 {
        host-inbound-traffic {
            system-services {
                webapi-ssl;
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

***Configuring the ClearPass Authentication Table Entry Timeout and Priority***

**Step-by-Step Procedure**

This procedure configures the following information:

- The timeout parameter that determines when to age out idle authentication entries in the ClearPass authentication table.
- The ClearPass authentication table as the first authentication table in the lookup order for the SRX Series device to search for user authentication entries. If no entry is found in the ClearPass authentication table and there are other authentication tables configured, the SRX Series device will search them, based on the order that you set.

1. Set the timeout value that is used to expire idle authentication entries in the ClearPass authentication table to 20 minutes.

   ```
   [edit services user-identification]
   user@host# set authentication-source aruba-clearpass authentication-entry-timeout 20
   ```

   The first time that you configure the SRX Series device to integrate with an authentication source, you must specify a timeout value to identify when to expire idle entries in the ClearPass authentication table. If you do not specify a timeout value, the default value is assumed.

   - default = 30 minutes
   - range = If set, the timeout value should be within the range [10,1440 minutes]. A value of 0 means that the entry will never expire.

2. Set the authentication table priority order to direct the SRX Series device to search for user authentication entries in the ClearPass authentication table first. Specify the order in which other authentication tables are searched if an entry for the user is not found in the ClearPass authentication table.

**NOTE:** You need to set this value if the ClearPass authentication table is *not* the only authentication table on the Packet Forwarding Engine.

```
[edit security user-identification]
user@host# set authentication-source aruba-clearpass priority 110
user@host# set authentication-source local-authentication-table priority 120
user@host# set authentication-source active-directory-authentication-table priority 125
user@host# set authentication-source firewall-authentication priority 150
user@host# set authentication-source unified-access-control priority 200
```

The default priority value for the ClearPass authentication table is 110. You must change the local authentication table entry from 100 to 120 to direct the SRX Series device to check the ClearPass authentication table first if there are other authentication tables on the Packet Forwarding Engine. Table 20 on page 320 shows the new authentication table search priority.

**Table 20: SRX Series Device Authentication Tables Search Priority Assignment**

| SRX Series Authentication Tables | Set Value |
|---|---|
| ClearPass authentication table | 110 |
| Local authentication table | 120 |
| Active Directory authentication table | 125 |
| Firewall authentication table | 150 |
| UAC authentication table | 200 |

**Results**

From configuration mode, confirm that the timeout value set for aging out ClearPass authentication table entries is correct. Enter the **show services user-identification** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
authentication-source aruba-clearpass {
    authentication-entry-timeout 20;
}
```

SEE ALSO

Understanding the Integrated ClearPass Authentication and Enforcement Feature | 299

Understanding Enforcement of ClearPass User and Group Authentication | 335

Understanding the Integrated ClearPass Authentication and Enforcement User Query Function | 321

## Understanding the Integrated ClearPass Authentication and Enforcement User Query Function

This topic focuses on how you can obtain user authentication and identity information for an individual user when that information is not posted directly to the SRX Series or NFX Series device by the ClearPass Policy Manager (CPPM).

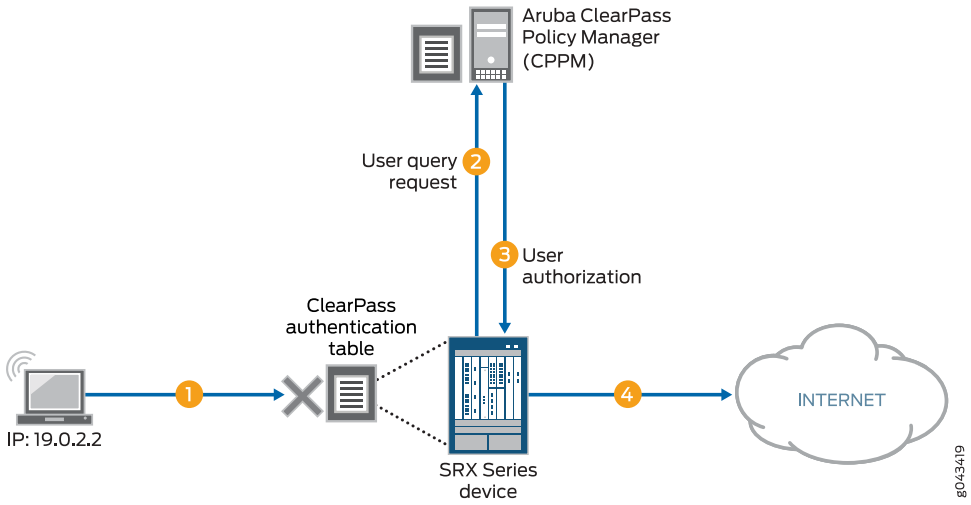
The integrated ClearPass authentication and enforcement feature allows the device and Aruba ClearPass to control access to protected resources and the Internet from wireless and wired devices. For this to occur, ClearPass sends user authentication and identity information to the device. The device stores the information in its ClearPass authentication table. To send this information, usually the CPPM uses the Web API (webapi) services implementation, which allows it to make HTTP or HTTPS POST requests to the device.

It can happen that the CPPM does not send user authentication information for a user, for various reasons. When traffic from that user arrives at the SRX Series or NFX Series device, the device cannot authenticate the user. If you configure the device to enable the user query function, it can query the ClearPass webserver for authentication information for an individual user. The device bases the query on the IP address of the user's device, which it obtains from the user's access request traffic.

If the user query function is configured, the query process is triggered automatically when the device does not find an entry for the user in its ClearPass authentication table when it receives traffic from that user requesting access to a resource or the Internet. The device does not search its other authentication tables. Rather, it sends a query to the CPPM requesting authentication information for the user.
Figure 20 on page 322 depicts the user query process. In this example:

1. A user attempts to access a resource. The device receives the traffic requesting access. The SRX Series device searches for an entry for the user in its ClearPass authentication table, but none is found.
2. The device requests authentication for the user from the CPPM.
3. The CPPM authenticates the user and returns the user authentication and identity information to the device.
4. The device creates an entry for the user in its ClearPass authentication table, and grants the user access to the Internet.

Figure 20: The ClearPass Integration User Query Function




You can control when the device sends its requests automatically by configuring the following two mechanisms:

- The **delay-query-time** parameter

  To determine the value to set for the **delay-query-time** parameter, it helps to understand the events and duration involved in how user identity information is transferred to the device from ClearPass, and how the **delay-query-time** parameter influences the query process.

  A delay is incurred from when the CPPM initially posts user identity information to the device using the Web API to when the device can update its local ClearPass authentication table with that information. The user identity information must first pass through the ClearPass device's control plane and the control plane of the device. In other words, this process can delay when the device can enter the user identity information in its ClearPass authentication table.

  While this process is taking place, traffic might arrive at the device that is generated by an access request from a user whose authentication and identity information is in transit from ClearPass to the device.

  Rather than allow the device to respond automatically by sending a user query *immediately*, you can set a **delay-query-time** parameter, specified in seconds, that allows the device to wait for a period of time before sending the query.

  After the delay timeout expires, the device sends the query to the CPPM and creates a pending entry in the Routing Engine authentication table. During this period, the traffic matches the default policy and is dropped or allowed, depending on the policy configuration.

> **NOTE:** If there are many query requests in the queue, the device can maintain multiple concurrent connections to ClearPass to increase throughput. However, to ensure that ClearPass is not stressed by these connections, the number of concurrent connections is constrained to no more than 20 (<=20). You cannot change this value.

- A default policy, which is applied to a packet if the device does not find an entry for the user associated with the traffic in its ClearPass authentication table.

  The system default policy is configured to drop packets. You can override this action by configuring a policy that specifies a different action to apply to this traffic.

Table 21 on page 323 shows the effect on the user query function in regard to whether or not Active Directory is enabled.

**Table 21: Relationship Between User Query Function and Active Directory Authentication as Processed by the CLI**

| Active Directory Is Configured | ClearPass User Query Function Is Enabled | CLI Check Result |
|---|---|---|
| No | No | Pass |
| No | Yes | Pass |
| Yes | No | Pass |
| Yes | Yes | Fail |

To avoid the failure condition reflected in the bottom row of the table, you must disable either Active Directory or the user query function. If both are configured, the system displays the following error message:

```
The priority of CP auth source is higher than AD auth source, and the CP user-query
 will shadow all AD features. Therefore, please choose either disabling CP
user-query or not configuring AD.
```

In its response to the user query request, the ClearPass web server returns information for the user's device whose IP address was specified in the request. This response includes a time stamp, which is expressed in UTC (Coordinated Universal Time) as defined by ISO 8601.

Here are some examples:

- 2016-12-30T09:30:10.678123Z

- 2016-12-30T09:30:10Z
- 2016-06-06T00:31:52-07:00

Table 22 on page 324 shows the components that comprise a timestamp format.

**Table 22: Time Stamp Components as Defined by ISO 8601**

| Format Component | Meaning |
|---|---|
| YYYY | two-digit month |
| DD | two-digit day of month |
| hh | two-digits of hour (00 through 23) |
| mm | two-digits of minute |
| ss | two-digits of second |
| s | one or more digits representing a decimal fraction of a second |
| TZD | time zone designator: Z or +hh:mm or -hh:mm |

## Example: Configuring the Integrated ClearPass Authentication and Enforcement User Query Function

**IN THIS SECTION**


- Requirements | 325
- Overview | 326
- Configuration | 329
- Verification | 332


This example covers how to configure the SRX Series device to enable it to query Aruba ClearPass automatically for user authentication and identity information for an individual user when that information is not available.

> **NOTE:** The user query function is supplementary to the Web API method of obtaining user authentication and identity information, and it is optional.

### Requirements

This section defines the software and hardware requirements for the overall topology that includes user query requirements. See Figure 22 on page 328 for the topology. For details on the user query process, see Figure 21 on page 327.

The hardware and software components are:

- Aruba ClearPass (CPPM). The CPPM is configured to use its local authentication source to authenticate users.

  > **NOTE:** It is assumed that the CPPM is configured to provide the SRX Series device with user authentication and identity information, including the username, a list of the names of any groups that the user belongs to, the IP addresses of the devices used, and the device posture token.

- SRX Series device running Junos OS that includes the integrated ClearPass feature.
- A server farm composed of six servers, all in the servers-zone:
  - marketing-server-protected (203.0.113.23 )
  - human-resources-server (203.0.113.25 )
  - accounting-server (203.0.113.72)
  - public-server (203.0.113.91)
  - corporate-server (203.0.113.71)
  - sales-server (203.0.113.81)
- AC 7010 Aruba Cloud Services Controller running ArubaOS.
- Aruba AP wireless access controller running ArubaOS.

  The Aruba AP is connected to the AC7010.

  Wireless users connect to the CPPM through the Aruba AP.
- Juniper Networks EX4300 switch used as the wired 802.1 access device.

Wired users connect to the CPPM using the EX4300 switch.

- Six end-user systems:
  - Three wired network-connected PCs running Microsoft OS
  - Two BYOD devices that access the network through the Aruba AP access device
  - One wireless laptop running Microsoft OS

### Overview

You can configure the user query function to enable the SRX Series device to obtain authenticated user identity information from the CPPM for an individual user when the device's ClearPass authentication table does not contain an entry for that user. The SRX Series device bases the query on the IP address of the user's device that generated the traffic issuing from the access request.

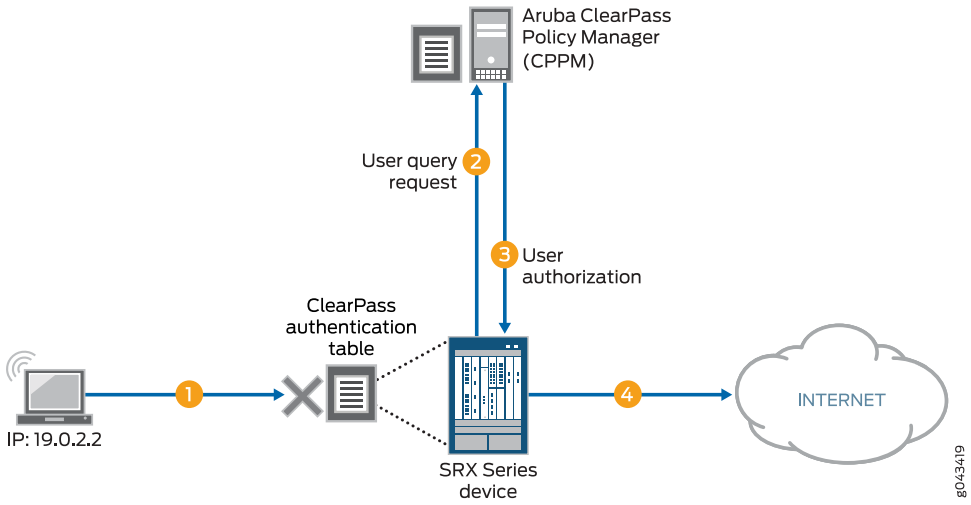
There are a number of reasons why the device might not already have authentication information from the CPPM for a particular user. For example, it can happen that a user has not already been authenticated by the CPPM. This condition could occur if a user joined the network through an access layer that is not on a managed switch or WLAN.

The user query function provides a means for the SRX Series device to obtain user authentication and identity information from the CPPM for a user for whom the CPPM did not post that information to the SRX Series device using the Web API. When the device receives an access request from a user for which there is not an entry in its ClearPass authentication table, it will automatically query the CPPM for it if this function is configured.

Figure 21 on page 327 shows the user query flow process, which encompasses the following steps:

1. A user attempts to access a resource. The SRX Series device receives the traffic requesting access. The device searches for an entry for the user in its ClearPass authentication table, but none is found.
2. The device requests authentication for the user from the CPPM.
3. The CPPM authenticates the user and returns the user authentication and identity information to the device.
4. The device creates an entry for the user in its ClearPass authentication table, and grants the user access to the Internet.

**Figure 21: User Query Function Process**




For details on the parameters that you can use to control when the device issues the query, see "Understanding the Integrated ClearPass Authentication and Enforcement User Query Function" on page 321.

> **NOTE:** You can also manually query the CPPM for authentication information for an individual user when this feature is configured.

The ClearPass endpoint API requires use of OAuth (RFC 6749) to authenticate and authorize access to it. For the device to be able to query the CPPM for individual user authentication and authorization information, it must acquire an access token. For this purpose, the device uses the Client Credentials access token grant type, which is one of the two types that ClearPass supports.

As administrator of the ClearPass Policy Manager (CPPM), you must create an API client on the CPPM with the grant_type set to "client_credentials". You can then configure the device to use that information to obtain an access token. Here is an example of the message format for doing this:

```
curl https://{$Server}/api/oauth – – insecure – – data
"grant_type=client_credentials&client_id=Client2&client_secret=
   m2Tvcklsi9je0kH9UTwuXQwIutKLC2obaDL54/fC2DzC"
```

A successful request from the device to obtain an access token results in a response that is similar to the following example:

```
{
```

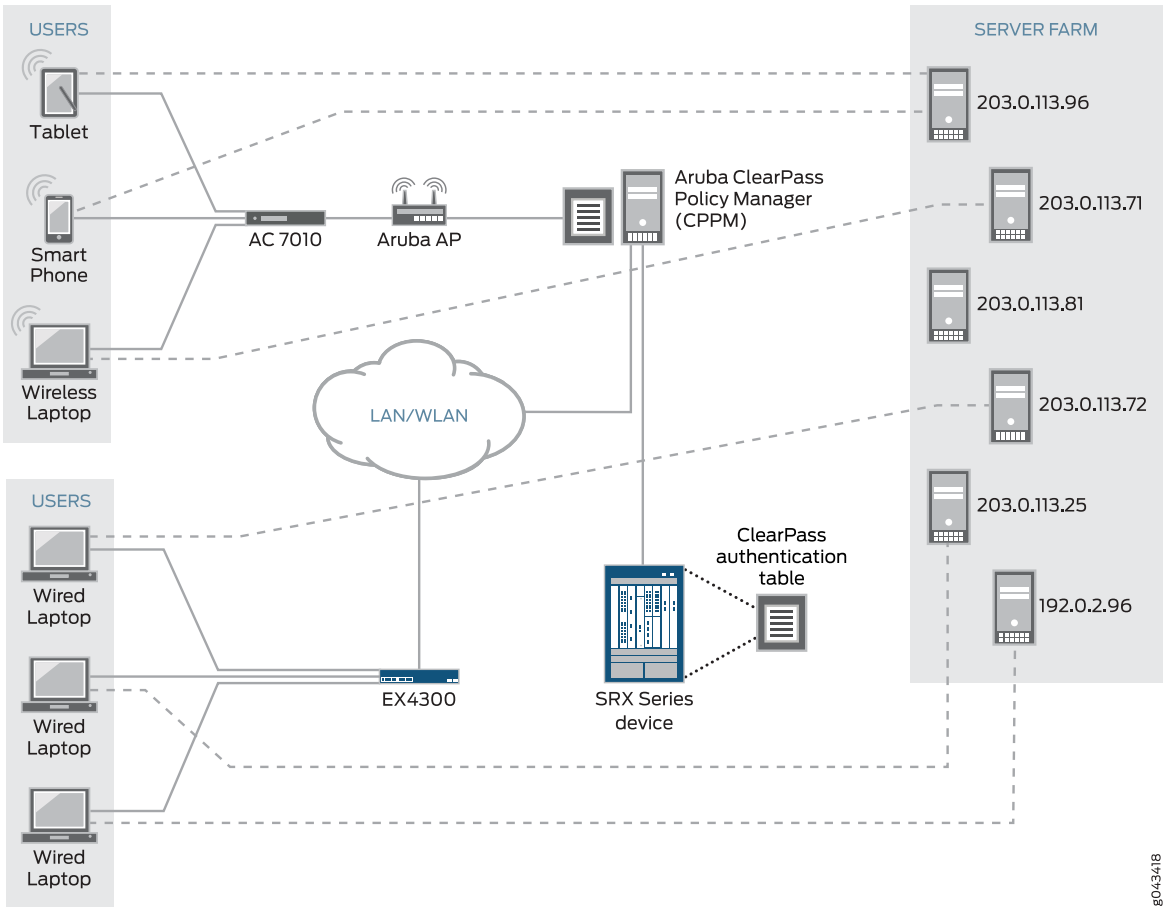
```
“access_token”:”ae79d980adf83ecb8e0eaca6516a50a784e81a4e”,
“expires_in”:2880,
“token_type”:”Bearer”,
“scope”=nu;
}
```

Before the access token expires, the device can obtain a new token using the same message.

***Topology***

Figure 22 on page 328 shows the overall topology for this deployment, which encompasses the user query environment.

**Figure 22: Topology for the Overall Deployment that Includes User Query**



## Configuration

**IN THIS SECTION**


- Configure the User Query Function (Optional) | **329**
- Manually Issuing a Query to the CPPM for Individual User Authentication Information (Optional) | **332**


To enable and configure the user query function, perform these tasks:

### CLI Quick Configuration

To quickly configure this example, copy the following statements, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the statements into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set services user-identification authentication-source aruba-clearpass user-query web-server cp-webserver
    address 192.0.2.199
set services user-identification authentication-source aruba_clearpass user-query ca-certificate
    RADUISServerCertificate.crt
set services user-identification authentication-source aruba-clearpass user-query client-id client-1
set services user-identification authentication-source aruba-clearpass user-query client-secret 7cTr13#
set services user-identification authentication-source aruba-clearpass user-query token-api “api/oauth”
set services user-identification authentication-source aruba-clearpass user-query IP address
    “api/vi/insight/endpoint/ip/$IP$”
```

### *Configure the User Query Function (Optional)*

### Step-by-Step Procedure

Configure the user query function to allow the SRX Series device to connect automatically to the ClearPass client to make requests for authentication information for individual users.

The user query function supplements input from the CPPM sent using the Web API. The Web API daemon does not need to be enabled for the user query function to work. For the user query function, the SRX Series device is the HTTP client. By it sends HTTPS requests to the CPPM on port 443.

To enable the SRX Series device to make individual user queries automatically:

1. Configure Aruba ClearPass as the authentication source for user query requests, and configure the ClearPass webserver name and its IP address. The device requires this information to contact the ClearPass webserver.

Starting with Junos OS Release 15.1X49-D130, you can configure Aruba Clearpass server IP address with IPv6 address, in addition to IPv4 address. Prior to Junos OS Release 15.1X49-D130, IPv4 address was only supported.

> **NOTE:** You must specify aruba-clearpass as the authentication source.

```
[edit services user-identification]
user@host# set authentication-source aruba-clearpass user-query web-server cp-webserver address
  192.0.2.199
```

> **NOTE:** You can configure only one ClearPass webserver.

Optionally, configure the port number and connection method, or accept the following values for these parameters. This example assumes the default values.

- connect-method (default is HTTPS)
- port (by default, the device sends HTTPS requests to the CPPM on port 443

However, if you were to explicitly configure the connection method and port, you would use these statements:

```
set services user-identification authentication-source aruba-clearpass user-query web-server cp-webserver
  connect method <https/http>
set services user-identification authentication-source aruba-clearpass user-query web-server cp-webserver
  port port-number
```

2. (Optional) Configure the ClearPass CA certificate file for the device to use to verify the ClearPass webserver. (The default certificate is assumed if none is configured.)

```
[edit services user-identification]
user@host# set authentication-source aruba_clearpass user-query ca-certificate RADUISServerCertificate.crt
```

The ca-certificate enables the SRX Series device to verify the authenticity of the ClearPass webserver and that it is trusted.

Before you configure the certificate, as administrator of the ClearPass device you must take the following actions:

- Export the ClearPass webserver's certificate from CPPM and import the certificate to the device.

- Configure the ca-certificate as the path, including its CA filename, as located on the SRX Series device. In this example, the following path is used:

  ```
  /var/tmp/RADUISServerCertificate.crt
  ```

3. Configure the client ID and the secret that the SRX Series device requires to obtain an access token required for user queries.

   ```
   [edit services user-identification]
   user@host# set authentication-source aruba-clearpass user-query client-id client-1
   user@host# set authentication-source aruba-clearpass user-query client-secret 7cTr13#
   ```

   The client ID and the client secret are required values. They must be consistent with the client configuration on the CPPM.

   > **TIP:** When you configure the client on the CPPM, copy the client ID and secret to use in the device configuration.

4. Configure the token API that is used in generating the URL for acquiring an access token.

   > **NOTE:** You must specify the token API. It does not have a default value.
   >
   > ```
   > [edit services user-identification]
   > user@host# set authentication-source aruba-clearpass user-query token-api “api/oauth”
   > ```
   >
   > In this example, the token API is **api/oauth**. It is combined with the following information to generate the complete URL for acquiring an access token **https://192.0.2.199/api/oauth**
   >
   > - The connection method is HTTPS.
   > - In this example, the IP address of the ClearPass webserver is 192.0.2.199.

5. Configure the query API to use for querying individual user authentication and identity information.

   ```
   [edit services user-identification]
   user@host# set authentication-source aruba-clearpass user-query query-api ’api/vi/insight/endpoint/ip/$IP$’
   ```

   In this example, the query-api is **api/vi/insight/endpoint/ip/$IP$**. It is combined with the URL **https://192.0.2.199/api/oauth** resulting in **https://192.0.2.199/api/oauth/api/vi/insight/endpoint/ip/$IP$**.

The $IP variable is replaced with the IP address of the end-user's device for the user whose authentication information the SRX Series is requesting.

6. Configure the amount of time in seconds to delay before the device sends the individual user query.

```
[edit services user-identification]
user@host# set authentication-source aruba-clearpass user-query delay-query-time 10
```

***Manually Issuing a Query to the CPPM for Individual User Authentication Information (Optional)***

**Step-by-Step Procedure**

- Configure the following statement to manually request authentication information for the user whose device's IP address is 203.0.113.46.

```
root@device>request service user-identification authentication-source aruba-clearpass user-query address
  203.0.113.46
```

## Verification

**IN THIS SECTION**


- Verifying That the ClearPass Webserver Is Online | 332
- Enabling Trace and Checking the Output | 333
- Determining If the User Query Function Is Executing Normally | 333
- Determining If a Problem Exists by Relying on User Query Counters | 333


Use the following procedures to verify that the user query function is behaving as expected:

***Verifying That the ClearPass Webserver Is Online***

**Purpose**

Ensure that the ClearPass webserver is online, which is the first mean of verifying that the user query request can complete successfully.

**Action**

Enter the **show service user-identification authentication-source authentication-source user-query status** command to verify that ClearPass is online.

```
show service user-identification authentication-source aruba-clearpass
user-query status
```

```
Authentication source: aruba-clearpass
Web server Address: 192.0.2.199
Status: Online
Current connections: 0
```

#### *Enabling Trace and Checking the Output*

**Purpose**

Display in the trace log any error messages generated by the user query function.

**Action**

Set the trace log file name and enable trace using the following commands:

**set system services webapi debug-log trace-log-1**

**set services user-identification authentication-source aruba-clearpass traceoptions flag user-query**

#### *Determining If the User Query Function Is Executing Normally*

**Purpose**

Determine if there is a problem with user query function behavior.

**Action**

Check syslog messages to determine if the user query request failed.

If it failed, the following error message is reported:

```
LOG1: sending user query for IP <ip-address> to ClearPass web server failed.
:reason
```

The reason might be “server unconnected” or “socket error”.

#### *Determining If a Problem Exists by Relying on User Query Counters*

**Purpose**

Display the user query counters to home in on the problem, if one exists, by entering the **show service user-identification authentication-source *authentication-source* user-query counters** command.

**NOTE:** The timestamp returned by ClearPass in response to the user query request can be specified in any of the ISO 8601 formats, including the format that includes a time zone.

**Action**

```
show service user-identification authentication-source aruba-clearpass
user-query counters
```

```
Authentication source: aruba-clearpass

    Web server Address: Address: ip-address
    Access token: token-string
    RE quest sent number: counter
    Routing received number: counter
    Time of last response: timestamp
```

# Enforce Security Policies using ClearPass

**IN THIS SECTION**


- Understanding Enforcement of ClearPass User and Group Authentication | 335
- Example: Enforcing SRX Series Security Policies Using Aruba ClearPass as the Authentication Source | 345


By configuring the security policies, you can control access to the internet for users based on their username and group name.

# Understanding Enforcement of ClearPass User and Group Authentication

IN THIS SECTION


- Understanding How the Device Manages the ClearPass Authentication Table | 335
- User Authentication Entries in the ClearPass Authentication Table | 336
- Communication Between ClearPass and the SRX Series or NFX Series Device | 339
- Understanding Domains and Interested Groups | 342
- When a User Has Already Been Authenticated By Another Source | 345


This topic describes how the SRX Series or NFX Series device enforces user and group authentication when a user attempts to access a resource. It also explains how the device handles information in the ClearPass authentication table user entries when a security policy that references a group in a user entry is removed. Understanding that process will help you troubleshoot issues related to group identity and give you insight into changes in the ClearPass authentication table user entries.

## Understanding How the Device Manages the ClearPass Authentication Table

The integrated ClearPass authentication and enforcement feature enables the SRX Series or NFX Series device and the Aruba ClearPass Policy Manager (CPPM) to collaborate in protecting your company's resources. It enables the device to apply firewall security policies to user traffic and to control user access to protected resources based on user or group identity. To ensure the identity of the user, the device relies on authenticated user information that it receives from the CPPM.

It is useful to understand how the device gets authenticated user identity information from the CPPM, generates entries in its ClearPass authentication table, and manages those entries in relation to security policies and user events. Understanding these processes will help you to quickly identify and resolve related problems.

This topic focuses on:

- How the device obtains user identity information from the CPPM and manages it, and how you can use this information in security policies.
- How security policies that reference a group as the source (source-identity) have bearing on the groups listed in user entries in the ClearPass authentication table. Groups that are referenced by security policies are referred to as *interested groups*.

**WARNING:**

### User Authentication Entries in the ClearPass Authentication Table

In their collaboration, ClearPass acts as the authentication source for the SRX Series or NFX Series device. The CPPM sends to the device identity information about users that it has authenticated. The UserID daemon process in the device receives this information, processes it, and synchronizes it to the Packet Forwarding Engine side in the independent ClearPass authentication table that is generated for this purpose.

As administrator of the device, you can use the authenticated user identity information in security policies to control access to your protected resources and the Internet.

The collection of user identity information that the device obtains from the CPPM and uses to create entries in its global Routing Engine authentication table that is synchronized to its individual ClearPass authentication table is referred to as a mapping, or, more commonly, an IP-user mapping because the username and the related group list are mapped to the IP address of the user's device.

For each user authentication entry in the ClearPass authentication table, a group list identifies the groups that a user belongs to in addition to other information such as the posture token, which indicates state of the device, such as whether it is healthy.

**NOTE:** The integrated user firewall feature for both ClearPass and active directory authentication will manage up to 2048 sessions for each user for whom there is a user identity and authentication entry in the authentication table. There might be additional sessions associated with a user beyond the 2048 supported sessions, but they are not managed by integrated user firewall. When an authentication entry in an authentication table is deleted, integrated user firewall only closes sessions that are associated with that entry. It will not close sessions that it does not manage. That is, sessions that are not associated with the authentication entry are not closed.

You can use a username or a group name in security policies to identity a user and not rely directly on the IP address of the device used, because the IP address of the device is tied to the username and its groups in the ClearPass authentication table entry.

**NOTE:** For each user entry, the number of groups, or roles, in the entry cannot exceed 200. After the capacity is reached, additional roles are discarded and the following syslog message is sent:

```
userid_get_and_check_adauth_num: src_ip ip-address user domain:user dropped.record
  numrecord-number has arrived max num of db
```

The CPPM posts user information to the device in the following format. The device does not use all of this information.

```
<userfw-entries>
  <userfw-entry>
    <source>Aruba ClearPass</source>
    <timestamp>2016-01-29T0310Z</timestamp>
    <operation>logon</operation>
    <IP>192.0.2.123</IP>
    <domain>my-company-domain</domain>
    <user>user1</user>
    <role-list>
      <role>human-resources-grp</role>
      <role>[User Authenticated],/role>
    </role-list>
    <posture>HEALTHY</posture>
    <device_category>Computer</device_category>
  </userfw-entry>
</userfw-entries>
```

Here is the format for a ClearPass authentication table entry for a user, followed by an example entry and a description of its components.

```
IP-address, domain, user, user-group-list
```

In the following example, the user belongs to two groups, the human-resources-grp group and the posture-healthy group. The SRX Series device converts the posture information from the CPPM to a group name. You might configure a security policy that allows all users access to the marketing server if their devices belong to the posture-healthy group (role).

```
192.0.2.11 , my-company-domain, lin, human-resources-grp, posture-healthy
```

- IP address

  This is the IP address of the device used.

- The name of the domain that the user belongs to.

  In this example, the domain name is “my-company-domain.” The default domain name GLOBAL is used if a domain name is not provided.

- The username

  The username is the user’s login name used to connect to the network, which, in this example, is lin.

  This name is constant regardless of the device used.

  When you configure a security policy whose source-identity tuple identifies the source of the traffic by username or group name, not by the IP address of the device used, it is as if the security policy were device independent; it applies to the user’s activity regardless of the device used.

- One or more groups that a user belongs to

  It is here where the concept of *interested groups* and their relationship to security policies comes into play. An interested group is a group that is referenced in a security policy. The concept of interested groups is covered later in this topic.

Note that if a user is connected to the network using multiple devices, there might be more than one IP-user mapping for that user. Each mapping would have its own set of values—that is, domain name and group-list—in conjunction with the username and IP address.

For example, the following three IP address-to-username mappings might exist for the user abe who is connected to the network using three separate devices:

```
203.0.113.5 abe, marketing-grp, posture-healthy
192.0.2.34 abe, marketing-grp, posture-transition
203.0.133.19 abe, marketing-grp, posture-unknown
```

Assume that the SRX Series device receives a logout message for 110.208.132.23, abe. The following partial user authentication entry shows that the user abe is now logged in to the network using only two devices:

```
192.0.2.34 abe, marketing-grp, posture-transition
203.0.133.19 abe, marketing-grp, posture-unknown
```

**WARNING:** If more than 2048 sessions are associated with a single authentication entry in the ClearPass authentication table, the integrated user firewall for ClearPass will not manage the sessions that caused the overflow. Consequently, there will be no user identification information for those sessions reported in the session close log for those sessions.

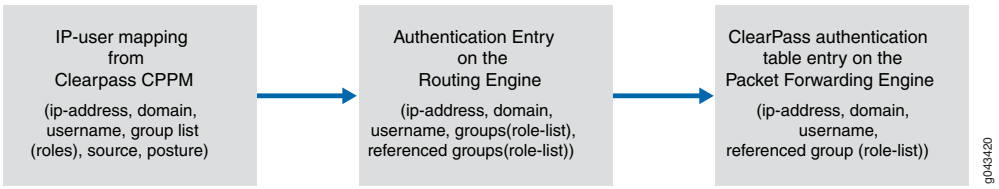
### Communication Between ClearPass and the SRX Series or NFX Series Device

Here is a summary of how the SRX Series or NFX Series device and ClearPass communicate:

- A user joins the company network via a wired or wireless LAN.
- The CPPM authenticates the user.
- The CPPM initiates a secure connection with the device using the integrated Web API.
- The UserID daemon gets the full IP-user mapping from the CPPM. For each authenticated user, the UserID daemon generates an entry in the Routing Engine authentication table.

  The Routing Engine authentication table is common in that it holds authentication entries based on information from other authentication sources in addition to ClearPass. For example, it might also hold entries for users authenticated by Microsoft Active Directory.
- The UserID daemon synchronizes the user authentication information from the Routing Engine authentication table to the ClearPass authentication table on the Packet Forwarding Engine. The ClearPass authentication table is dedicated to holding only ClearPass authentication information. See Figure 23 on page 340.

**Figure 23: User Information from the CPPM to the SRX Series Device Routing Engine Synchronized to the ClearPass Authentication Table**




The device uses the authenticated user identity information in the following process. When a user attempts to access an internal, protected resource or the Internet, the device:

- Checks the traffic generated by the user for a matching security policy. The source traffic must match all of the tuples specified in the security policy. The match includes the source-identity field, which specifies a username or a group name.

  To identify a match, the device compares the username or the group name with the source-identity specification that is configured in a security policy, along with all other security policy values.

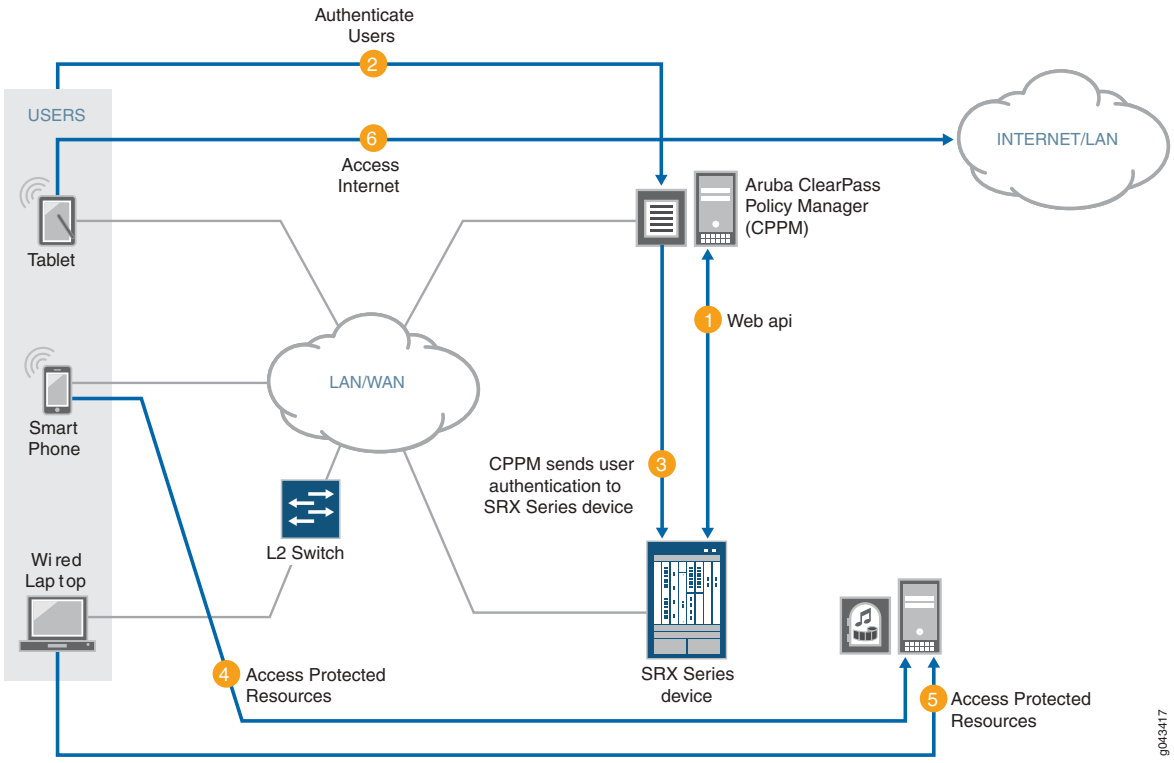
- Checks the ClearPass authentication table for an authentication entry for the user, if a security policy match was found.

  If it does not find an entry in the ClearPass authentication table, the device checks other local authentication tables, in the order that you specified, until a match is found. However, it does not check other local authentication tables if the user query function is configured. See "Understanding the Integrated ClearPass Authentication and Enforcement User Query Function" on page 321.

> **NOTE:** The device can query the CPPM for individual user information, under certain circumstances, when it has not already received that information from the CPPM. This feature is referred to as user query.

Figure 24 on page 341 illustrates the connection and communication between the device and the CPPM. It also shows the paths entailed in authenticating users and allowing them access to the Internet and internal, protected resources.

Figure 24: ClearPass and SRX Series Device Communication and User Authentication Process




As Figure 24 on page 341 depicts, the following activity takes place:

1. The CPPM initiates a secure connection with the SRX Series device using the Web API.
2. Three users join the network and are authenticated by the CPPM.
   - A tablet user joins the network across the corporate WAN.
   - A smartphone user joins the network across the corporate WAN.
   - A wireless laptop user joins the network from a wired laptop connected to a Layer 2 switch that is connected to the corporate LAN.
3. The CPPM sends the user authentication and identity information for the users who are logged in to the network to the device in POST request messages using the Web API.

   When traffic from a user arrives at the device, the device:
   - Identifies a security policy that the traffic matches.
   - Locates an authentication entry for the user in the ClearPass authentication table.
   - Applies the security policy to the traffic after authenticating the user.
4. Traffic from the smartphone user who is requesting access to an internal, protected resource arrives at the device. Because all of the conditions identified in Step 3 are met and the security policy permits it, the device allows the user connection to the protected resource.

5. Traffic from the wired laptop user who is requesting access to a protected resource arrives at the device. Because all of the conditions identified in Step 3 are met and the security policy permits it, the device allows the user connection to the resource.
6. Traffic from the tablet user who is requesting access to the Internet arrives at the device. Because all of the conditions identified in Step 3 are met and the security policy permits it, the device allows the user connection to the Internet.

### Understanding Domains and Interested Groups

How the user identity group information is managed on the device is dominated by two concepts:

- Domain group

  The device follows the usual course in regard to how it handles usernames in domain namespaces. It makes use of the namespace to distinguish names that are the same–such as **admin**—but that are from different sources and are in different domains. Because they belong to different domains, the names are not in conflict.

  Any group that is part of an IP-user mapping will always belong to a domain, whether that domain is a specific domain or the GLOBAL domain. If a domain name is not specified in the IP-user mapping, then the GLOBAL domain is assumed.

  Table 23 on page 342 illustrates how the domain for a group is determined, based on the IP-user mapping information obtained from the CPPM.

  **Table 23: Assigning a Domain to a Group**

| Does the IP-User Mapping Contain a Domain Name? | What Domain Is Applied to the Group? |
|---|---|
| No<br><br>For example:<br><br>IP, , user1, group-list<br><br>The second comma serves as a placeholder for the domain name and the GLOBAL domain is applied. | Groups included in group-list belong to the GLOBAL domain. |
| Yes<br><br>For example:<br><br>IP, domain1, user1, group-list<br><br>**NOTE:** In this example, the IP-user mapping specifies the domain name as domain1. | The domain name, domain1, is included in the IP-user mapping from the CPPM, and it is used. It is retained in the entry for the authenticated user in the ClearPass authentication table on the Packet Forwarding Engine. |

- Interested group

  A group qualifies as an *interested group* if it is referenced by a security policy–that is, if it is specified in a policy's source-identity field. On the Routing Engine authentication table, each user entry contains a group referenced by a policy list that identifies the names of the groups for which a security policy exists. If a group included in a user entry is not currently used in a security policy, it is not included in this list. A group can move in and out of the groups referenced by a policy list.

  - Interested group lists

    An interested group list, or a list of groups referenced by policies, is a subset of overall groups. It is the intersection of the group list in a user authentication entry and the source-identity list for security policies. That is, any group included in a ClearPass authentication table user entry qualifies as an interested group. The Routing Engine synchronizes to the user entry in the ClearPass authentication table on the Packet Forwarding Engine only those groups that are referenced by security policies.

    Here is how it works:

    - The UserID daemon gets the full IP-user role (group) mapping from the CPPM.
    - For each group, the UserID daemon identifies whether it is an interested group by determining if there is a security policy that references it. Any qualifying groups are included in the groups referenced by a policy list on the Routing Engine. The UserID daemon synchronizes to the user entry in the ClearPass authentication table on the Packet Forwarding Engine interested groups along with the rest of the user authentication and identity information.

    The interested groups list for a user entry on the Routing Engine can change, based on the following events:

    - A new security policy is configured that references a group included in the user entry on the Routing Engine but that is not already in the entry's referenced groups list.
    - A currently configured security policy that references a group in its source-identity is deleted.

    Consider the following example:

    - Assume that the CPPM posted the following information for two users to the SRX Series device:

      ```
      192.51.100.1, abe, group1, group2, group3, group4, healthy
      192.0.2.21, john, group1, group5, healthy
      ```

    - After the device maps the posture, defining it as a group, the two user entries in the device Routing Engine authentication table appear as follows:

      ```
      192.51.100.1, abe, group1, group2, group3, group4, posture-healthy
      192.0.2.21, john, group1, group5, posture-healthy
      ```

    - Assume that several security policies include source-identity fields that reference one of the following: group1, group3, posture-healthy.

The intersection of the preceding sets—the original group list and the list of security policies that refer to the groups—results in the following interested groups list:

- For the user john, the groups referenced by policy list includes group1 and posture-healthy.
- For the user abe, the groups referenced by policy list includes group1, group3, and posture-healthy.

Now suppose that the security policy whose source-identity field specified group1 was deleted. The groups referenced by policy lists for the user authentication entries for the two users—john and abe—would be changed, producing the following results:

- For the user john, the list would include only posture-healthy.
- For the user abe, the list would include group3 and posture-healthy.

Table 24 on page 344 shows how a security policy that references a group affects the ClearPass authentication table. It also shows the effect on the ClearPass authentication table when a group is *not* referenced by a security policy, and therefore is not an interested group.

**Table 24: Interested Groups: Effect on the ClearPass Authentication Table**

| Security Policies Configuration and Modification | Resulting Effect on ClearPass Authentication Table Packet Forwarding Engine Entries |
|---|---|
| **Case 1**:<br><br>The SRX Series device gets the IP-user mapping for a user from the CPPM.<br><br>None of the groups in the user mapping are referenced by security policies. | |
| IP-user mapping from the CPPM:<br><br>203.0.113.9, ,user1, g1, g2, g3, g4 | The user authentication entry written to the ClearPass authentication table in the Packet Forwarding Engine for this user does not contain any groups.<br><br>203.0.113.9 , ,user1 |
| **Case 2**:<br><br>The SRX Series device gets the IP-user mapping for a user from the CPPM. It checks the groups list against the security policies list and finds that two of the groups are referenced by security policies. | |
| IP-user mapping on the Routing Engine:<br><br>192.0.2.1, domain1, user2, g1, g2, g3, g4 | The user authentication entry written to the ClearPass authentication table on the Packet Forwarding Engine for this user includes the following groups that are included in the groups referenced by the policy list on the Routing Engine:<br><br>192.0.2.1, domain1, user2, g2, g4 |

### When a User Has Already Been Authenticated By Another Source

It can happen that the device Routing Engine authentication table and the individual Microsoft Active Directory authentication table on the Packet Forwarding Engine, for example, contain an entry for a user who was authenticated by Active Directory. As usual, the CPPM sends the IP-user mapping for the user to the device. The device must resolve the problem because its Routing Engine authentication table is common to both Active Directory and ClearPass.

Here is how the device handles the situation:

- On the Routing Engine authentication table:
  - The device overwrites the Active Directory authentication entry for the user in its common Routing Engine authentication table with the newly generated one from the IP-user mapping for the user from the CPPM.

    There is now no IP address or username conflict.
- On the Packet Forwarding Engine:
  - The device deletes the existing Active Directory authentication entry for the user from the Active Directory authentication table.

    This will delete active sessions associated with the IP address.
  - The device generates a new entry for the CPPM-authenticated user in the Packet Forwarding Engine ClearPass authentication table.

    Traffic associated with the IP-user mapping entry will initiate new sessions based on user authentication in the ClearPass authentication table.

## Example: Enforcing SRX Series Security Policies Using Aruba ClearPass as the Authentication Source

**IN THIS SECTION**


- Requirements | **347**
- Overview | **348**
- Configuration | **351**
- Verification | **363**

This example covers how to configure security to protect your resources and control access to the internet using the SRX Series device integrated ClearPass authentication and enforcement feature, which relies on the Aruba ClearPass Policy Manager as its authentication source. The SRX Series integrated ClearPass feature allows you to configure security policies that control access to company resources and the Internet by identifying users by username, group name, or the name of a role that ties together a group of users and a device type.

Today's network environments are more open to attacks of various kinds because they support *anywhere*, *anytime*, *any device* access, to a greater or lesser degree, and they allow a user to use multiple concurrently network-connected devices. Because it allows you identify the user by username, the integrated ClearPass authentication and enforcement feature narrows the security gap that these capabilities introduce.

For details on how user authentication and identity information is conveyed from the CPPM to the SRX Series device, see the following topics:

- Understanding How ClearPass Initiates a Session and Communicates User Authentication Information Using the Web API on page 306
- Understanding the Integrated ClearPass Authentication and Enforcement User Query Function on page 321

The example covers the following processes:

- How to control access at the user level based on username or group name, not device IP address.

  You can use the source-identity parameter in a security policy to specify the name of a user or the name of a group of users whose authentication is provided by the CPPM. The policy is applied to traffic generated by the users when they attempt to access a protected resource or the Internet regardless of the device used. The access control is tied to the user's name, and not directly to the IP address of the user's device.

  > **NOTE:** You can configure different security policies for a single user that specify different actions, differentiated by the zones and the destination addresses specified or a group that the user belongs to.

- How to display and interpret the contents of the ClearPass authentication table.

  The SRX Series device creates the ClearPass authentication table to contain user authentication and identity information that it receives from the CPPM. The device refers to the table to authenticate a user who requests access to a resource.

  The ClearPass authentication table contents are dynamic. They are modified to reflect user activity in response to various events and also in regard to security policies that reference groups.

  For example, when a user logs out of the network or in to the network, the ClearPass authentication table is modified, as is the case when a user is removed from a group or a referenced security policy that

specifies a group that the user belongs to is deleted. In the latter case, the user entry no longer shows the user as belonging to that group.

In this example, the ClearPass authentication table contents are displayed to depict changes made because of two events. The content for the users is displayed:

- Before and after a specific user logs out of the network
- Before and after a referenced security policy is deleted

  The entry for the user who belonged to the group referenced by the security policy is displayed before and after the policy is deleted.

### Requirements

This section defines the software and hardware requirements for the topology for this example. See Figure 25 on page 351 for the topology design.

The hardware and software components are:

- Aruba ClearPass. The ClearPass Policy Manager (CPPM) is configured to use its local authentication source to authenticate users.

  > **NOTE:** It is assumed that the CPPM is configured to provide the SRX Series device with user authentication and identity information, including the username, a list of the names of any groups that the user belongs to, the IP addresses of the devices used, and the device posture token.

- SRX Series device running Junos OS that includes the integrated ClearPass feature.
- A server farm composed of six servers, all in the servers-zone:
  - marketing-server-protected (203.0.113.23 )
  - human-resources-server (203.0.113.25 )
  - accounting-server (203.0.113.72)
  - public-server (203.0.113.62)
  - corporate-server (203.0.113.71)
  - sales-server (203.0.113.81)
- AC 7010 Aruba Cloud Services Controller running ArubaOS.
- Aruba AP wireless access controller running ArubaOS.

  The Aruba AP is connected to the AC7010.

Wireless users connect to the CPPM through the Aruba AP.

- Juniper Networks EX4300 switch used as the wired 802.1 access device.

  Wired users connect to the CPPM using the EX4300 switch.

- Six end-user systems:
  - Three wired network-connected PCs running Microsoft OS
  - Two BYOD devices that access the network through the Aruba AP access device
  - One wireless laptop running Microsoft OS

### Overview

In its capacity as the authentication source for the integrated ClearPass feature, the CPPM posts to the SRX Series device user authentication and identity information. When it receives this information, the SRX Series UserID daemon processes it and generates entries for the authenticated users in the Routing Engine authentication table and then synchronizes that information to the ClearPass authentication table on the Packet Forwarding Engine side.

The SRX Series device requires the user authentication and identity information to verify that a user is authenticated when the user makes an access request and the traffic generated from the user's device arrives at the SRX Series device. If a security policy exists that specifies in the source-identity parameter the username or the name of a group that the user belongs to, the SRX Series device searches the contents of its ClearPass authentication table for an entry for that user.

If it does not find an entry for the user in its ClearPass authentication table, the SRX Series device can search its other authentication tables, if you have configured a search order that includes them. See "Example: Configuring the SRX Series Integrated ClearPass Feature to Allow the Device to Receive User Authentication Data from ClearPass" on page 310 for information about the authentication table search order.

The integrated ClearPass feature allows you to create identity-aware security policies configured to match traffic issued by users based on their username or the name of a group that they belong to.

> **NOTE:** You configure role mappings on the CPPM, not on the SRX Series device.

For example, a device type role mapping might tie user identities to company-owned computers. You could specify this role as a group in a security policy configured to apply to all users who are mapped to the rule. In this case, the conditions set by CPPM for the rule—use of company-owned computer—would apply to all users mapped to the rule. The SRX Series device does not consider the conditions, but rather accepts the rule from the CPPM.

The following configurations included in this example cover security policies that are applicable based on the type of device used as defined by the CPPM through rule mappings. It is assumed that the CPPM posted to the SRX Series device the following mapped rules that are used as groups in security policies:

- marketing-access-for-pcs-limited-group

  Maps jxchan to the device type PC.

  The policy that specifies marketing-access-for-pcs-limited-group in its source-identity field allows jxchan, and other users who are mapped to it, access to the marketing-server-protected server using their PC, whether it is company owned or not.

- accounting-grp-and-company-device

  Maps users who belong to accounting groups using company devices. The CPPM sends the role accounting-grp-and-company-device to the SRX Series device. The mapping is done on the CPPM by role mapping rules.

  The policy that specifies accounting-grp-and-company-device in its source identity field allows users who are mapped to the rule to access protected resources on the accounting-server. The group accounting-grp is mapped to the rule. Therefore the mapped rule applies to the members of accounting-grp.

  The user viki2 belongs to accounting-grp. If all conditions apply—that is, if viki2 is using a company-owned device and the policy permits access—she is allowed access to the resources on accounting-server. But, recall that the SRX Series device does not analyze the rule. Rather it applies it to all users who are mapped to it by the CPPM.

- guest-device-byod

  Maps the guest group to the device type byod—that is, any user-owned device brought to the network.

  The policy that specifies guest-device-byod in its source identity field denies users who are mapped to the rule access to all servers in the server zone if they are using smartphones or other user-owned devices. The username guest2 is mapped to this rule by the CPPM.

For all cases, if the users are allowed or denied access according to the security policy conditions, you can assume that the following conditions exist:

- The CPPM posted the correct authentication information for the users and groups to the SRX Series device.
- The SRX Series device processed the authenticated user information correctly and generated entries for the users and groups in its ClearPass authentication table.

Starting with Junos OS Release 15.1X49-D130, the SRX Series device supports the use of IPv6 addresses associated with source identities in security policies. If IPv4 or IPv6 entry exists, policies matching that entry are applied to the traffic and access is allowed or denied.

Table 25 on page 350 summarizes the users, their groups, and the zones to which they belong. All users belong to the default GLOBAL domain.

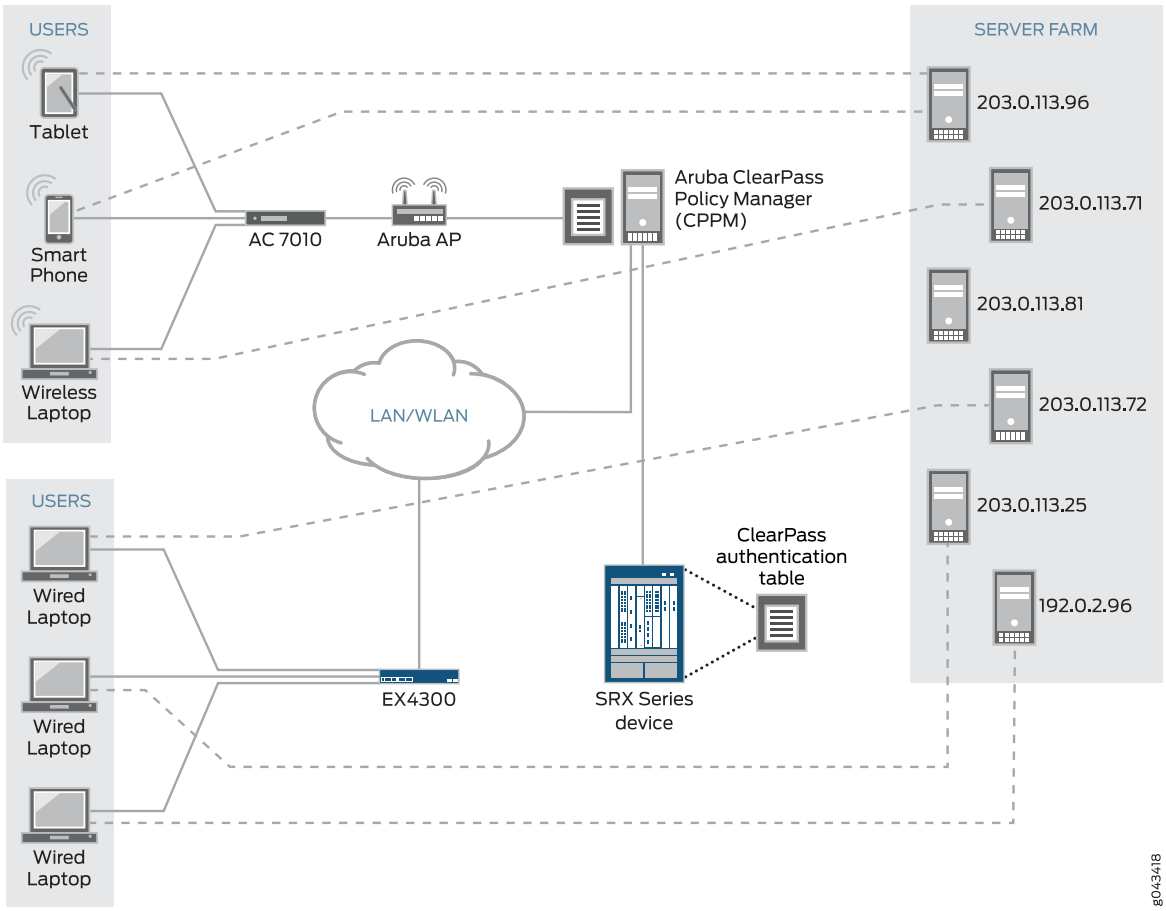
**Table 25: Authenticated User Information for Security Policy Example**

| User | Group | Zone |
|---|---|---|
| Abe (abew1) | • marketing-access-limited-grp | marketing-zone |
| John (jxchan) | • posture-healthy<br>• marketing-access-for-pcs-limited-group<br>• marketing-general<br>• sales-limited<br>• corporate-limited | marketing-zone |
| Lin (lchen1) | • posture-healthy<br>• human-resources-grp<br>• accounting-limited<br>• corporate-limited | human-resources-zone |
| Viki (viki2) | • posture-healthy<br>• accounting-grp<br>• accounting-grp-and-company-device<br>• corporate-limited | accounting-zone |
| guest1 | • posture-healthy<br>• guest | public-zone |
| guest2 | • posture-healthy<br>• guest-device-byod | public-zone |

***Topology***

Figure 25 on page 351 shows the topology for this example.

**Figure 25: Topology for the Integrated ClearPass Authentication Enforcement Through Security Policies Example**




## Configuration

**IN THIS SECTION**


- Configuring Interfaces, Zones, and an Address Book | **354**
- Configuring Identity-Aware Security Policies to Control User Access to Company Resources | **358**
- Results | **361**


This section covers how to configure the SRX Series device to include security policies that match traffic issued by users authenticated by the CPPM.

### CLI Quick Configuration

To quickly configure this example, copy the following statements, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the statements into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3.0 vlan-id 300 family inet address 203.0.113.45/24
set interfaces ge-0/0/3.1 vlan-id 310 family inet address 192.0.2.18/24
set interfaces ge-0/0/3.2 vlan-id 320 family inet address 192.0.2.14/24
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4.0 vlan-id 400 family inet address 192.0.2.16/24
set interfaces ge-0/0/4.1 vlan-id 410 family inet address 192.0.2.19/24
set security zones security-zone marketing-zone interfaces ge-0/0/3.0 host-inbound-traffic system-services
    all
set security zones security-zone marketing-zone interfaces ge-0/0/3.0 host-inbound-traffic protocols all
set security zones security-zone accounting-zone interfaces ge-0/0/3.1 host-inbound-traffic system-services
    all
set security zones security-zone accounting-zone interfaces ge-0/0/3.1 host-inbound-traffic protocols all
set security zones security-zone human-resources-zone interfaces ge-0/0/3.2 host-inbound-traffic
    system-services all
set security zones security-zone human-resources-zone interfaces ge-0/0/3.2 host-inbound-traffic protocols
    all
set security zones security-zone public-zone interfaces ge-0/0/4.0 host-inbound-traffic system-services all
set security zones security-zone public-zone interfaces ge-0/0/4.0 host-inbound-traffic protocols all
set security zones security-zone servers-zone interfaces ge-0/0/4.1 host-inbound-traffic system-services all
set security zones security-zone servers-zone interfaces ge-0/0/4.1 host-inbound-traffic protocols all
set security address-book servers-zone-addresses address marketing-server-protected 203.0.113.23
set security address-book servers-zone-addresses address human-resources-server 203.0.113.25
set security address-book servers-zone-addresses address accounting-server 203.0.113.72
set security address-book servers-zone-addresses address corporate-server 203.0.113.71
set security address-book servers-zone-addresses address public-server 203.0.113.91
set security address-book servers-zone-addresses attach zone servers-zone
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p1 match source-address
    any destination address any
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p1 match application
    any
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p1 match source-identity
    “global\marketing-access-for-pcs-limited-group”
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p1 then permit
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p2 match source-address
    any destination address marketing-zone-protected
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p2 match application
    any
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p2 match source-identity
    “global\abew1”
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p2 then permit
```

```
set security policies from-zone accounting-zone to-zone servers-zone policy acct-cp-device match source-address
    any destination-address accounting-server
set security policies from-zone accounting-zone to-zone servers-zone policy acct-cp-device match application
    any
set security policies from-zone accounting-zone to-zone servers-zone policy acct-cp-device match source-identity
    “global\accounting-grp-and-company-device”
set security policies from-zone accounting-zone to-zone servers-zone policy acct-cp-device then permit
set security policies from-zone human-resources-zone to-zone servers-zone policy human-resources-p1 match
    source-address any destination-address corporate-server
set security policies from-zone human-resources-zone to-zone servers-zone policy human-resources-p1 match
    application any
set security policies from-zone human-resources-zone to-zone servers-zone policy human-resources-p1 match
    source-identity “global\corporate-limited”
set security policies from-zone human-resources-zone to servers-zone policy human-resources-p1 then permit
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p0 match source-address
    any destination-address corporate-server
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p0 match application
    any
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p0 match source-identity
    “global\marketing-access-limited-grp”
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p0 then permit
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p3 match source-address
    any destination-address human-resources-server
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p3 match application
    any
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p3 match source-identity
    “global\sales-limited-group”
set security policies from-zone marketing-zone to-zone servers-zone policy marketing-p3 then permit
set security policies from-zone public-zone to-zone servers-zone policy guest-allow-access match source-address
    any destination address public-server
set security policies from-zone public-zone to-zone servers-zone policy guest-allow-access match application
    any
set security policies from-zone public-zone to-zone servers-zone policy guest-allow-access match source-identity
    “global\guest”
set security policies from-zone public-zone to-zone servers-zone policy guest-allow-access then permit
set security policies from-zone public-zone to-zone servers-zone policy guest-deny-access match source-address
    any destination-address any
set security policies from-zone public-zone to-zone servers-zone policy guest-deny-access match application
    any
set security policies from-zone public-zone to-zone servers-zone policy guest-deny-access match source-identity
    “global\guest-device-byod”
set security policies from-zone public-zone to-zone servers-zone policy guest-deny-access then deny
```

***Configuring Interfaces, Zones, and an Address Book***

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

Configure the following interfaces and assign them to zones:

- ge-0/0/3.0 > marketing-zone
- ge-0/0/3.1 > human-resources-zone
- ge-0/0/3.2> accounting-zone
- ge-0/0/4.0 > public-zone
- ge-0/0/4.1 > servers-zone

Because this example uses logical interfaces, you must configure VLAN tagging.

1. Configure interfaces for the SRX Series device:

```
[edit interfaces]
set ge-0/0/3 vlan-tagging
set ge-0/0/3.0 vlan-id 300 family inet address 203.0.113.45/24
set ge-0/0/3.1 vlan-id 310 family inet address 192.0.2.18/24
set ge-0/0/3.2 vlan-id 320 family inet address 192.0.2.14/24
set ge-0/0/4 vlan-tagging
set ge-0/0/4.0 vlan-id 400 family inet address 192.0.2.16/24
set ge-0/0/4.1 vlan-id 410 family inet address 192.0.2.19/24
```

2. Configure zones.

```
[edit security zones]
user@host#set security-zone marketing-zone interfaces ge-0/0/3.0 host-inbound-traffic system-services
   all
user@host#set security-zone marketing-zone interfaces ge-0/0/3.0 host-inbound-traffic protocols all
user@host#set security-zone accounting-zone interfaces ge-0/0/3.1 host-inbound-traffic system-services
   all
user@host#set security-zone accounting-zone interfaces ge-0/0/3.1 host-inbound-traffic protocols all
user@host#set security-zone human-resources-zone interfaces ge-0/0/3.2 host-inbound-traffic
   system-services all
user@host#set security-zone human-resources-zone interfaces ge-0/0/3.2 host-inbound-traffic protocols
   all
user@host#set security-zone public-zone interfaces ge-0/0/4.0 host-inbound-traffic system-services all
user@host#set security-zone public-zone interfaces ge-0/0/4.0 host-inbound-traffic protocols all
user@host#set security-zone servers-zone interfaces ge-0/0/4.1 host-inbound-traffic system-services all
```

```
user@host#set security-zone servers-zone interfaces ge-0/0/4.1 host-inbound-traffic protocols all
```

3. Configure an address book containing the IP addresses of the servers to use as destination addresses in security policies.

```
[edit security address-book servers-zone-addresses]
user@host# set address marketing-server-protected 203.0.113.23
user@host# set address human-resources-server 203.0.113.25
user@host# set address accounting-server 203.0.113.72
user@host# set address corporate-server 203.0.113.71
user@host# set address public-server 203.0.113.91
```

4. Attach the servers-zone-addresses address book to servers-zone.

```
[edit security address-book]
user@host# set servers-zone-addresses attach zone servers-zone
```

**Results**

From configuration mode, confirm your configuration for interfaces by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
ge-0/0/3 {
    unit 0 {
        vlan-id 300;
        family inet {
            address 203.0.113.45/24;
        }
    }
    unit 1 {
        vlan-id 310;
        family inet {
            address 192.0.2.18/24;
        }
    }
    unit 2 {
        vlan-id 320;
        family inet {
            address 192.0.2.14/24;
        }
```

```
    }
}
ge-0/0/4 {
    vlan-tagging;
    unit 0 {
        vlan-id 400;
        family inet {
            address 192.0.2.16/24;
        }
    }
    unit 1 {
        vlan-id 410;
        family inet {
            address 192.0.2.19/24;
        }
    }
}
```

From configuration mode, confirm your configuration for zones by entering the **show security zones** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
security-zone human-resources-zone {
    interfaces {
        ge-0/0/3.2 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
security-zone accounting-zone {
    interfaces {
        ge-0/0/3.1 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
```

```
                all;
            }
        }
    }
}
security-zone marketing-zone {
    interfaces {
        ge-0/0/3.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
security-zone servers-zone {
    interfaces {
        ge-0/0/4.1 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
security-zone public-zone {
    interfaces {
        ge-0/0/4.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
```

```
        }
      }
    }
  }
```

From configuration mode, confirm your configuration for the address book by entering the **show security address-book** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
servers-zone-addresses {
    address marketing-zone-protected 203.0.113.23 /32;
    address human-resources-server 203.0.113.25 /32;
    address accounting-server 203.0.113.72/32;
    address corporate-server 203.0.113.71/32;
    address public-server 203.0.113.91/32;
    attach {
        zone servers-zone;
    }
}
```

***Configuring Identity-Aware Security Policies to Control User Access to Company Resources***

**Step-by-Step Procedure**

This task entails configuring security policies that apply to a user's access to resources based on username or group name, and not the IP address of the device used.

Note that all users belong to the default GLOBAL domain.

1. Configure a security policy that specifies marketing-access-for-pcs-limited-group as the source-identity. It allows the user jxchan, who belongs to this group, access to any of the servers in the servers-zones when he is using a PC, whether it is a personal device or a company-owned device. The username jxchan is mapped by the CPPM to the rule marketing-access-for-pcs-limited-group.

   ```
   [edit security policies]
   user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p1 match source-address
       any destination address any
   user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p1 match application any
   user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p1 match source-identity
       "global\marketing-access-for-pcs-limited-group"
   user@hoset from-zone marketing-zone to-zone servers-zone policy marketing-p1 then permit
   ```

2. Configure a security policy that allows the user abew1 access to the marketing-zone-protected server (IP address 203.0.113.23 ) in the servers-zone regardless of the device that he uses.

```
[edit security policies]
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p2 match source-address
    any destination address marketing-zone-protected
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p2 match application any
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p2 match source-identity
    “global\abew1”
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p2 then permit
```

3. Configure a security policy that allows the user viki2 access to the accounting-server (IP address 203.0.113.72) in the servers-zone when she is using a company-owned device. The user viki2 belongs to accounting-grp which is mapped to the company-owned-device rule (accounting-grp-and-company-device) by the CPPM.

```
[edit security policies]
user@host# set from-zone accounting-zone to-zone servers-zone policy acct-cp-device match source-address
    any destination-address accounting-server
user@host# set from-zone accounting-zone to-zone servers-zone policy acct-cp-device match application
    any
user@host# set from-zone accounting-zone to-zone servers-zone policy acct-cp-device match source-identity
    “global\accounting-grp-and-company-device”
user@host# set from-zone accounting-zone to-zone servers-zone policy acct-cp-device then permit
```

4. Configure a security policy that allows users who belong to the corporate-limited group limited access to the corporate-server server (IP address 203.0.113.71) in the servers-zone when they are initiating a request from the human-resources zone.

   If the source-address were specified as “any”, the policy would apply to other users who also belong to the corporate-limited group.

```
[edit security policies]
user@host# set from-zone human-resources-zone to-zone servers-zone policy human-resources-p1 match
    source-address any destination-address corporate-server
user@host# set from-zone human-resources-zone to-zone servers-zone policy human-resources-p1 match
    application any
user@host# set from-zone human-resources-zone to-zone servers-zone policy human-resources-p1 match
    source-identity “global\corporate-limited”
user@host# set from-zone human-resources-zone to servers-zone policy human-resources-p1 then permit
```

5. Configure a security policy that allows the user abew1 access to the corporate-server (IP address 203.0.113.71) server in the servers-zone. The user abew1 belongs to marketing-access-limited-grp to which the security policy applies.

```
[edit security policies]
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p0 match source-address
    any destination-address corporate-server
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p0 match application any
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p0 match source-identity
    “global\marketing-access-limited-grp”
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p0 then permit
```

6. Configure a security policy that allows users who belong to the sales-limited-group access to the human-resources-server (IP address 203.0.113.81) server when they initiate a request from the marketing-zone. The user jxchan belongs to sales-limited-group.

```
[edit security policies]
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p3 match source-address
    any destination-address human-resources-server
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p3 match application any
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p3 match source-identity
    “global\sales-limited-group”
user@host# set from-zone marketing-zone to-zone servers-zone policy marketing-p3 then permit
```

7. Configure a security policy that allows users who belong to the guest group access to the public-server (IP address 203.0.113.91) in the servers-zone.

```
[edit security policies]
user@host# set from-zone public-zone to-zone servers-zone policy guest-allow-access match source-address
    any destination address public-server
user@host# set from-zone public-zone to-zone servers-zone policy guest-allow-access match application
    any
user@host# set from-zone public-zone to-zone servers-zone policy guest-allow-access match source-identity
    “global\guest”
user@host# set from-zone public-zone to-zone servers-zone policy guest-allow-access then permit
```

8. Configure a security policy that denies users who belong to the guest-device-byod group access to any servers in the servers-zone when they use their own devices.

```
[edit security policies]
user@host# set from-zone public-zone to-zone servers-zone policy guest-deny-access match source-address
    any destination-address any
user@host# set from-zone public-zone to-zone servers-zone policy guest-deny-access match application
    any
```

```
user@host# user@host# set from-zone public-zone to-zone servers-zone policy guest-deny-access match
  source-identity “global\guest-device-byod”
user@host# set from-zone public-zone to-zone servers-zone policy guest-deny-access then deny
```

***Results***

From configuration mode, confirm your security policies configuration for integrated ClearPass by entering the **show security policies** command.

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
from-zone marketing-zone to-zone servers-zone {
    policy marketing-p1 {
        match {
            source-address any;
            destination-address any;
            application any;
            source-identity "global\marketing-access-for-pcs-limited-group";
        }
        then {
            permit;
        }
    }
    policy marketing-p2 {
        match {
            source-address any;
            destination-address marketing-zone-protected;
            application any;
            source-identity "global\abew1";
        }
        then {
            permit;
        }
    }
    policy marketing-p0 {
        match {
            source-address any;
            destination-address corporate-server;
            application any;
            source-identity "global\marketing-access-limited-grp";
        }
        then {
            permit;
```

```
        }
    }
    policy marketing-p3 {
        match {
            source-address any;
            destination-address human-resources-server;
            application any;
            source-identity "global\sales-limited-group";
        }
        then {
            permit;
        }
    }
}
from-zone accounting-zone to-zone servers-zone {
    policy acct-cp-device {
        match {
            source-address any;
            destination-address accounting-server;
            application any;
            source-identity "global\accounting-grp-and-company-device";
        }
        then {
            permit;
        }
    }
}
from-zone human-resources-zone to-zone servers-zone {
    policy human-resources-p1 {
        match {
            source-address any;
            destination-address corporate-server;
            application any;
            source-identity "global\corporate-limited";
        }
        then {
            permit;
        }
    }
}
from-zone public-zone to-zone servers-zone {
    policy guest-allow-access {
        match {
            source-address any;
```

```
            destination-address public-server;
            application any;
            source-identity “global\guest”;
        }
        then {
            permit;
        }
    }
    policy guest-deny-access {
        match {
            source-address any;
            destination-address any;
            application any;
            source-identity “global\guest-device-byod”;
        }
        then {
            deny;
        }
    }
}
```

## Verification

**IN THIS SECTION**


- Displaying the ClearPass Authentication Table Contents Before and After an Authenticated User Logs Out of the Network | **363**
- Displaying the Authentication Table Contents Before and After a Referenced Security Policy Is Deleted | **364**


This section verifies the ClearPass authentication table contents after certain events occur that cause some of its user authentication entries to be modified. It also shows how to ensure that the ClearPass authentication table has been deleted successfully after you issue the delete command. It includes the following parts:

### *Displaying the ClearPass Authentication Table Contents Before and After an Authenticated User Logs Out of the Network*

#### Purpose

Display the ClearPass authentication table contents when a specific, authenticated user is logged in to the network and after the user logs out.

**Action**

Enter the **show services user-identification authentication-table authentication-source *authentication-source*** command for the ClearPass authentication table, which is referred to as aruba-clearpass. Notice that the ClearPass authentication table includes an entry for the user viki2.

```
show services user-identification authentication-table authentication-source
aruba-clearpass
```

```
Domain: GLOBAL
Total entries: 6
Source IP       Username       groups(Ref by policy)          state
203.0.113.21        viki2          accounting-grp-and-company-dev Valid
203.0.113.89        abew1          marketing-access-limited-grp   Valid
203.0.113.52        jxchan         marketing-access-for-pcs-limit Valid
203.0.113.53        lchen1         corporate-limited              Valid
203.0.113.54        guest1                                        Valid
203.0.113.55        guest2                                        Valid
```

Enter the same command again after viki2 logs out of the network. Notice that the ClearPass authentication table no longer contains an entry for viki2.

```
Domain: GLOBAL
Total entries: 6
Source IP       Username       groups(Ref by policy)          state
203.0.113.89        abew1          marketing-access-limited-grp   Valid
203.0.113.52        jxchan         marketing-access-for-pcs-limit Valid
203.0.113.53        lchen1         corporate-limited              Valid
203.0.113.54        guest1                                        Valid
203.0.113.55        guest2                                        Valid
```

***Displaying the Authentication Table Contents Before and After a Referenced Security Policy Is Deleted***

**Purpose**

Display the ClearPass authentication table contents for a specific user—lchen1—who belongs to a group that is referenced by a security policy. Delete that security policy, then display the entry for that user again.

**Action**

Enter the **show service user-identification authentication-table authentication-source user *user-name*** command to display the ClearPass authentication table entry for a specific user, lchen1. Notice that it includes the group corporate-limited.

**show service user-identification authentication-table authentication-source user lchen1**

```
Domain: GLOBAL
Source IP       Username       groups(Ref by policy)          state
203.0.113.53        lchen1         corporate-limited              Valid
```

The human-resources-p1 security policy source-identity field refers to the group corporate-limited. As shown above in the ClearPassauthentication entry for him, the user lchen1 belongs to that group. Here is the configuration for the human-resources-p1 referenced security policy:

```
from-zone human-resources-zone to-zone servers-zone {
    policy human-resources-p1 {
        match {
            source-address any;
            destination-address corporate-server;
            application any;
            source-identity "global\corporate-limited";
        }
        then {
            permit;
        }
    }
}
```

After you delete the human-resources-p1 security policy, whose source-identity parameter refers to the group called corporate-limited, enter the same command again. Notice that the authentication entry for lchen1 does not contain the corporate-limited group.

**show service user-identification authentication-table authentication-source aruba-clearpass user lchen1**

```
Domain: GLOBAL
Source IP       Username       groups(Ref by policy)          state
203.0.113.53        lchen1                                        Valid
```

Take a different approach in verifying the ClearPass authentication table state after the modification. Display the entire table to verify that the group—corporate-limited—is not included in any of the user

entries. Note that if more than one user belonged to the corporate-limited group, authentication entries for all of the affected users would not show that group name.

From operational mode, enter the **show services user-identification authentication-table authentication-source aruba-clearpass** command.

```
show services user-identification authentication-table authentication-source
aruba-clearpass
```

```
Domain: GLOBAL
Total entries: 6
Source IP       Username       groups(Ref by policy)          state
203.0.113.21        viki2          accounting-grp-and-company-dev Valid
203.0.113.89        abew1          marketing-access-limited-grp   Valid
203.0.113.52        jxchan         marketing-access-for-pcs-limit Valid
203.0.113.53        lchen1                                        Valid
203.0.113.54        guest1                                        Valid
203.0.113.55        guest2                                        Valid
```

**Release History Table**

| Release | Description |
|---|---|
| 15.1X49-D130 | Starting with Junos OS Release 15.1X49-D130, the SRX Series device supports the use of IPv6 addresses associated with source identities in security policies. If IPv4 or IPv6 entry exists, policies matching that entry are applied to the traffic and access is allowed or denied. |

# Filter and Transmit Threat and Attack Logs to ClearPass

**IN THIS SECTION**


- Understanding How the Integrated ClearPass Feature Detects Threats and Attacks and Notifies the CPPM | **367**
- SRX Series Threat and Attack Logs Sent to Aruba ClearPass | **369**
- Example: Configuring Integrated ClearPass to Filter and Rate-limit Threat and Attack Logs | **371**

The SRX Series device transmits the threat and attack logs recorded to the ClearPass Policy Manager (CPPM). You can also configure the threats and attacks related to a specific device and their users. CPPM can use the log data to harden the security.

## Understanding How the Integrated ClearPass Feature Detects Threats and Attacks and Notifies the CPPM

The integrated ClearPass authentication and enforcement feature allows you to integrate your device with the ClearPass Policy Manager (CPPM) to obtain authenticated user identity information. It also allows the device to send attack and threat logs to the CPPM. This topic focuses on sending attack and threat logs to the CPPM.

When the device features detect threat and attack events, the event is recorded in the device event log. The device uses syslog to forward the logs to the CPPM. The CPPM can evaluate the logs and take action based on matching conditions. As administrator of ClearPass, you can use the information from the device and define appropriate actions on the CPPM to harden your security.

Junos OS on the SRX Series device generates over 100 different types of log entries issued by more than 10 of its modules. Among the device features that generate threat and attack logs are SCREENS, IDP, and UTM. To avoid overburdening the SRX Series device and the log server, the integrated ClearPass feature allows you to configure the device to send to the CPPM only attack and threat log entries that were written to the event log in response to activity detected by the SCREENS, IDP, and UTM security features.

You can set the following conditions to control the log transmission:

- A log stream filter to ensure that only threat and attack logs are sent.
- A rate limiter to control the transmission volume. The device log transmission will not exceed the rate-limiting conditions that you set.

For the CPPM to analyze the log information that the sends to it, the content must be formatted in a standard, structured manner. The device log transmission follows the syslog protocol, which has a message format that allows vendor-specific extensions to be provided in a structured way.

Here is an example of an attack log generated by IDP:

```
<14>1 2014-07-24T1358.362+08:00 bjsolar RT_IDP - IDP_ATTACK_LOG_EVENT
[junos@2636.1.1.1.2.86 epoch-time="1421996988" message-type="SIG"
source-address="192.0.2.66" source-port="32796" destination-address="192.0.2.76"
destination-port="21" protocol-name="TCP" service-name="SERVICE_IDP"
application-name="NONE" rule-name="1" rulebase-name="IPS" policy-name="idpengine"
 export-id="4641"repeat-count="0" action="NONE" threat-severity="MEDIUM"
attack-name="FTPROOT" nat-source-address="0.0.0.0" nat-source-port="0"
```

```
nat-destination-address="0.0.0.0" nat-destination-port="0" elapsed-time="0"
inbound-bytes="0" outbound-bytes="0" inbound-packets="0" outbound-packets="0"
source-zone-name="untrust" source-interface-name="ge-0/0/1.0"
destination-zone-name="trust" destination-interface-name="ge-0/0/7.0"
packet-log-id="0" alert="no" username="N/A" roles="N/A" message="-"]
```

Table 26 on page 368 uses the content of this example IDP attack log to identify the parts of an attack log entry. See "SRX Series Threat and Attack Logs Sent to Aruba ClearPass" on page 369 for further details on types of attack and threat logs.

**Table 26: Attack Log Fields Using Example Log**

| Log Entry Component | Meaning | Format | Example |
|---|---|---|---|
| Priority | pri = LOG_USER + severity. Version is always 1 | pri *version* | <14>1 |
| Time and Time Zone | When the log was recorded and in what time zone. | *y-m-dThs.ms+time zone*<br>• y = year<br>• m=month<br>• d = day<br>• T+hours | 2014-07-24T1358.362+08:00 |
| Device/Host Name | Name of the device from which the event log was sent. This value is configured by the user. | string, *hostname* | bjsolar |
| Service Name | SRX Series feature that issued the event log. | string *service* | SERVICE_IDP |
| Application Name | Application that generated the log entry. | string *application-name* | NONE |
| PID | Process ID.<br>The process ID is not meaningful in this context, so *pid* is replaced by "-".<br>The value "-" is a placeholder for process ID. | *pid* | - |

**Table 26: Attack Log Fields Using Example Log *(continued)***

| Log Entry Component | Meaning | Format | Example |
|---|---|---|---|
| Errmsg Tag | Log ID name, error message tag. | string, *log-name and tag* | IDP_ATTACK_LOG_EVENT |
| Errmsg Tag Square Bracket | Log content enclosed in square brackets. | [ ] | - |
| OID | Product ID provided by the chassis daemon (chassisd). | junos@oid | junos@2636.1.1.1.2.86 |
| Epoch Time | The time when the log was generated after the epoch. | *number* | 1421996988 |

## SRX Series Threat and Attack Logs Sent to Aruba ClearPass

The SRX Series integrated ClearPass authentication and enforcement feature collaborates with Aruba ClearPass in protecting a company's resources against potential and actual attacks through use of attack and threat event logs. These logs that are generated by the SRX Series SCREENS, IDP, and UTM components clearly identify the types of attacks and threats that threaten a company's network security.

The SRX Series device filters from the overall log entries the logs that report on threat and attack events, and it forwards these log entries to the ClearPass Policy Manager (CPPM) to be used in assessing and enforcing the company's security policy. The SRX Series device transmits the logs in volumes determined by the rate-limiting conditions that you set.

Table 27 on page 369 identifies the types of threat and attack log entries and the events that they represent.

**Table 27: Threat and Attack Log Entries Generated by SRX Series Components**

| Log Type | Description |
|---|---|
| RT_SCREEN_ICMP | ICMP attack |
| RT_SCREEN_ICMP_LS | |
| RT_SCREEN_IP | IP attack |
| RT_SCREEN_IP_LS | |

**Table 27: Threat and Attack Log Entries Generated by SRX Series Components *(continued)***

| Log Type | Description |
|---|---|
| RT_SCREEN_TCP | TCP attack |
| RT_SCREEN_TCP_LS | |
| RT_SCREEN_TCP_DST_IP | TCP destination IP attack |
| RT_SCREEN_TCP_DST_IP_LS | |
| RT_SCREEN_TCP_SRC_IP | TCP source IP attack |
| RT_SCREEN_TCP_SRC_IP_LS | |
| RT_SCREEN_UDP | UDP attack |
| RT_SCREEN_UDP_LS | |
| AV_VIRUS_DETECTED_MT | Virus infection<br>A virus was detected by the antivirus scanner. |
| AV_VIRUS_DETECTED_MT_LS | |
| ANTISPAM_SPAM_DETECTED_MT | spam<br>The identified e-mail was detected to be spam. |
| ANTISPAM_SPAM_DETECTED_MT_LS | |
| IDP_APPDDOS_APP_ATTACK_EVENT | Application-level distributed denial of service (AppDDoS) attack<br>The AppDDoS attack occurred when the number of client transactions exceeded the user-configured connection, context, and time binding thresholds. |
| IDP_APPDDOS_APP_ATTACK_EVENT_LS | |
| IDP_APPDDOS_APP_STATE_EVENT | AppDDoS attack<br>The AppDDoS state transition occurred when the number of application transactions exceeded the user-configured connection or context thresholds. |
| IDP_APPDDOS_APP_STATE_EVENT_LS | |
| IDP_ATTACK_LOG_EVENT | Attack discovered by IDP<br>IDP generated a log entry for an attack. |
| IDP_ATTACK_LOG_EVENT_LS | |

# Example: Configuring Integrated ClearPass to Filter and Rate-limit Threat and Attack Logs

IN THIS SECTION


- Requirements | 371
- Overview | 372
- Configuration | 373


The SRX Series device can dynamically send to the ClearPass Policy Manager (CPPM) information about threats and attacks identified by its security modules that protect network resources. It detects attack and attack threats that pertain to the activity of specific devices and their users, and it generates corresponding logs. To control this transmission, you must configure the type of logs to be sent and the rate at which they are sent. You can then use this information in setting policy rules on the CPPM to harden your network security.

This example shows how to configure the SRX Series integrated ClearPass authentication and enforcement feature to filter and transmit only threat and attack logs to the CPPM and to control the volume and rate at which the SRX Series device transmits them.

## Requirements

The topology for this example uses the following hardware and software components:

- Aruba CPPM implemented in a virtual machine (VM) on a server. The CPPM is configured to use its local authentication source to authenticate users.
- SRX Series device running Junos OS that includes the integrated ClearPass feature. The SRX Series device is connected to the Juniper Networks EX4300 switch and to the Internet. The SRX Series device communicates with ClearPass over a secure connection.
- Juniper Networks EX4300 switch used as the wired 802.1 access device. The EX4300 Layer 2 switch connects the endpoint users to the network. The SRX Series device is connected to the switch.
- Wired, network-connected PC running Microsoft OS. The system is directly connected to the EX4300 switch.

  Threat and attack logs are written for activity from these devices triggered by events that the security features catch and protect against.

## Overview

The SRX Series integrated ClearPass authentication and enforcement feature participates with Aruba ClearPass in protecting your company's resources against actual and potential attacks. The SRX Series device informs the CPPM about threats to your network resources and attacks against them through logs that it sends. You can then use this information to assess configuration of your security policy on the CPPM. Based on this information, you can harden your security in regard to individual users or devices.

To control the behavior of this feature, you must configure the SRX Series device to filter for attack and threat log entries and set rate-limiting conditions.

You can tune the behavior of this function in the following ways:

- Set a filter to direct the SRX Series device to send only threat and attack logs to the CPPM. This filter allows you to ensure that the SRX Series device and the log server do not need to handle irrelevant logs.
- Establish rate limit conditions to control the volume of logs that are sent.

  You set the rate-limit parameter to control the volume and rate that logs are sent. For example, you can set the rate-limit parameter to 1000 to specify that a maximum of 1000 logs are sent to ClearPass in 1 second. In this case, if there is an attempt to send 1015 logs, the number of logs over the limit—15 logs, in this case—would be dropped. The logs are not queued or buffered.

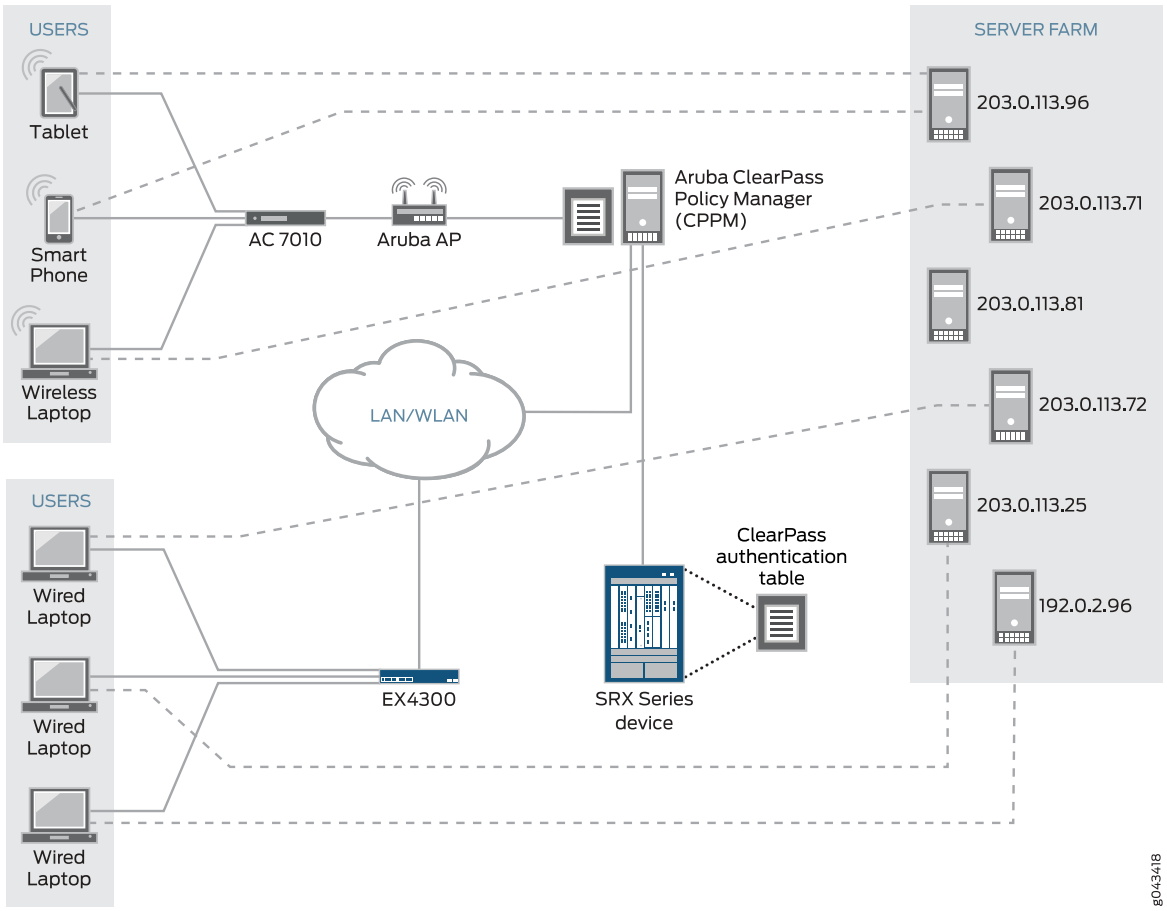
You can configure a maximum of three log streams with each individual log defined by its destination, log format, filter, and rate limit. Log messages are sent to all configured log streams. Each stream is individually rate-limited.

> **NOTE:** To support rate-limiting, log messages are sent out from the device's local SPU at a divided rate. In the configuration process, the Routing Engine assigns a divided rate to each SPU. The divided rate is equal to the configured rate divided by the number of SPUs on the device:
>
> ```
> divided-rate = configured-rate/number-of-SPUs
> ```

### *Topology*

Figure 26 on page 373shows the topology for this example.

Figure 26: Integrated ClearPass Authentication and Enforcement Deployment Topology




## Configuration

**IN THIS SECTION**


- Configuring Integrated ClearPass Authentication and Enforcement to Filter for Threat and Attack Logs Sent to the CPPM | **374**
- Results | **375**


This example covers how to configure a filter to select threat and attack logs to be sent to ClearPass. It also covers how to set a rate limiter to control the volume of logs sent during a given period. It includes these parts:

### CLI Quick Configuration

To quickly configure this example, copy the following statements, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the statements into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security log stream threat-attack-logs host 203.0.113.47
set security log mode stream
set security log source-interface ge-0/0/1.0
set security log stream to_clearpass format sd-syslog
set security log stream to_clearpass filter threat-attack
set security log stream to_clearpass rate-limit 1000
```

***Configuring Integrated ClearPass Authentication and Enforcement to Filter for Threat and Attack Logs Sent to the CPPM***

**Step-by-Step Procedure**

1. Specify a name for the log stream and the IP address of its destination.

```
[edit security]
user@host# set security log stream threat-attack-logs host 203.0.113.47
```

2. Set the log mode to stream.

```
[edit security]
user@host# set log mode stream
```

3. Set the host source interface number.

```
[edit security]
user@host#set log source-interface ge-0/0/1.0
```

4. Set the log stream to use the structured syslog format for sending logs to ClearPass through syslog.

```
[ edit security]
user@host# set log stream to_clearpass format sd-syslog
```

5. Specify the type of events to be logged.

```
[edit security]
user@host# set log stream to_clearpass filter threat-attack
```

> **NOTE:** This configuration is mutually exclusive in relation to the current category set for the filter.

6. Set rate limiting for this stream. The range is from 1 through 65,535.

   This example specifies that up to 1000 logs per second can be sent to ClearPass. When the maximum is reached, any additional logs are dropped.

```
[ edit security]
user@host# set log stream to_clearpass rate-limit 1000
```

***Results***

From configuration mode, confirm your configuration for interfaces by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
mode stream;
source-interface ge-0/0/1.0;
stream threat-attack-logs {
    host {
        203.0.113.47;
    }
}
stream to_clearpass {
    format sd-syslog;
    filter threat-attack;
    rate-limit {
        1000;
    }
}
```

# Configure ClearPass and JIMS at the Same Time

IN THIS SECTION


- Understanding How ClearPass and JIMS Works at the Same Time | 376
- Example: Configure ClearPass and JIMS at the Same Time | 379


You can configure ClearPass and Juniper Identity Management Service (JIMS) at the same time. By configuring the ClearPass and JIMS at the same time, the SRX Series or NFX Series devices can query JIMS for user identification entries, and ClearPass can push these entries to the devices through the Web API.

## Understanding How ClearPass and JIMS Works at the Same Time

The device relies on Juniper Identity Management Service (JIMS) and ClearPass for user identity information. Starting in Junos OS Release 18.2R1, you can configure JIMS, ClearPass, and Web API at the same time in UserFW. Prior to Junos OS Release 18.2R1, you can either configure ClearPass Policy Manager (CPPM) or JIMS. By configuring ClearPass and JIMS at the same time, the device can query JIMS to obtain user identity information from Active Directory and the exchange servers, and ClearPass can push the user authentication and identity information to the device through Web API.

### How ClearPass and JIMS Works at the Same Time?

When a user gets authenticated by CPPM, the CPPM uses a Web API to push user or device information to a device. The device builds up the authentication entry or device information for the user, and the user traffic can pass-through the device based on security policy. When windows Active Directory client log on to domain, device obtains client's user or device information from JIMS via batch query. The authentication table gets updated with entry provided by JIMS. The user traffic can pass-through the device based on security policy.

When both JIMS IP query and ClearPass user query are enabled, device always queries ClearPass first. If CPPM returns with IP-user mapping information, then the information is subsequently added to authentication table. If CPPM does not return the IP-user mapping information or if a device receives a response from CPPM without IP-user mapping, then the device queries JIMS to obtain IP-user or group mapping.

When the IP-user or group mapping is received from both JIMS and CPPM, device considers the latest authentication entries and overwrites the existing authentication entries.

You can set a **delay-query-time** parameter, specified in seconds, that allows the device to wait for a period of time before sending the query. The delay time should be the same value for ClearPass and JIMS. Otherwise, an error message is displayed and the commit check fails.

> **NOTE:** When the IP-user or group mapping is received from both JIMS and CPPM, the device considers the latest authentication entries and overwrites the existing authentication entries.

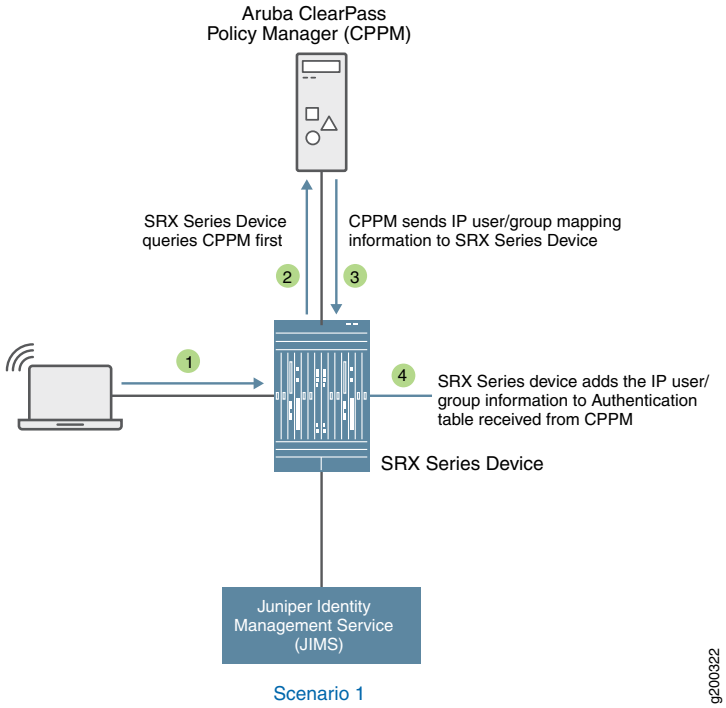
A more detailed explanation with scenarios of how ClearPass and JIMS works is as follows:

### Scenario 1: What an SRX Series Device Does If CPPM Responds with IP-User or Group Mapping Information?

Figure 27 on page 378 shows when an SRX Series device queries CPPM for IP-user or group mapping information and adds to the authentication table.

1. A user attempts to access a resource. When the SRX Series device receives the traffic request, it searches for an entry for the user in its ClearPass authentication table and the local Active Directory authentication table, but the user information is not found.
2. The SRX Series device queries ClearPass for user identity.
3. The ClearPass sends the IP-user or group mapping information to the SRX Series device.
4. The SRX Series device adds the information to the authentication table.

Figure 27: What SRX Series Device Does If CPPM Responds with IP-User or Group Mapping Information?

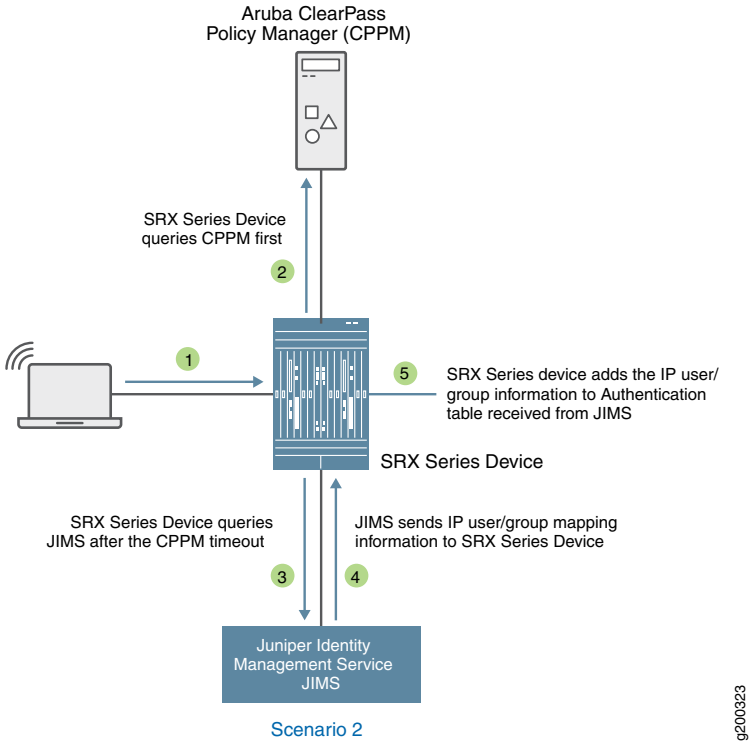



## Scenario 2: What an SRX Series Device Does If CPPM Does Not Respond or CPPM Responds with No IP-User or Group Mapping Information?

Figure 28 on page 379 shows when an SRX Series device queries JIMS if there is no response or no IP-user or group mapping information received from CPPM.

1. A user attempts to access a resource. When the SRX Series device receives the traffic request, it searches for an entry for the user in its ClearPass authentication table and JIMS authentication table, but the user information is not found.
2. The SRX Series device queries ClearPass for user identity.
3. If the SRX Series does not receive a response from ClearPass, the SRX Series device queries JIMS.
4. The JIMS sends IP-user or group mapping information to the SRX Series device.
5. The SRX Series device adds the information received from JIMS to the authentication table.

**Figure 28: What SRX Series Device Does If CPPM Does Not Respond or CPPM Responds with No IP-User or Group Mapping Information?**




## Example: Configure ClearPass and JIMS at the Same Time

**IN THIS SECTION**


- Requirements | 380
- Overview | 380
- Configuration | 380
- Verification | 385


This example shows how to enable Juniper Identity Management Service (JIMS) and ClearPass at the same time for user identity information, and verify how JIMS and ClearPass works at the same time. Also, this example explains which authentication entries are given first preference and how the timeouts behave for JIMS and ClearPass.

### Requirements

This example uses the following hardware and software components:

- An SRX Series device.
- An IP address of the JIMS server.
- ClearPass client IP address.
- Aruba ClearPass Policy Manager (CPPM). The CPPM is configured to use its local authentication source to authenticate users.

> **NOTE:** It is assumed that the CPPM is configured to provide the SRX Series device with user authentication and identity information, including the username, a list of the names of any groups that the user belongs to, the IP addresses of the devices used, and the device posture token.

### Overview

An SRX Series device obtains the user or device identity information from different authentication sources. After the SRX Series device obtains the device identity information, it creates an entry in the device identity authentication table. The SRX Series device relies on JIMS and ClearPass for user identity information. By enabling JIMS and ClearPass at the same time, an SRX Series device queries JIMS to obtain user identity information from Active Directory and the exchange servers, and CPPM pushes the user authentication and identity information to the SRX Series device through Web API.

When both JIMS IP query and ClearPass user query are enabled, SRX Series device always queries ClearPass first. When the IP-user or group mapping is received from both JIMS and CPPM, an SRX Series device considers the latest authentication entries and overwrites the existing authentication entries. You can set a **delay-query-time** parameter, specified in seconds, that allows the SRX Series device to wait for a period of time before sending the query. When JIMS and ClearPass are enabled, the delay time should be the same value for each other. Otherwise, an error message is displayed and the commit check fails.

### Configuration

#### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set services user-identification identity-management connection primary address 192.0.2.0
```

```
set services user-identification identity-management connection primary client-id otest
set services user-identification identity-management connection primary client-secret test
set services user-identification authentication-source aruba-clearpass user-query web-server cp-server
set services user-identification authentication-source aruba-clearpass user-query address 198.51.100.0
set services user-identification authentication-source aruba-clearpass user-query client-id otest
set services user-identification authentication-source aruba-clearpass user-query client-secret test
set services user-identification authentication-source aruba-clearpass user-query token-api oauth_token/oauth
set services user-identification authentication-source aruba-clearpass user-query query-api
   "user_query/v1/ip/$IP$"
set system services webapi user root
set system services webapi user password “$ABC123"
set system services webapi client 203.0.113.0
set system services webapi https port 8443
set system services webapi https default-certificate
set services user-identification authentication-source aruba-clearpass authentication-entry-timeout 30
set services user-identification authentication-source aruba-clearpass invalid-authentication-entry-timeout 30
set services user-identification identity-management authentication-entry-timeout 30
set services user-identification identity-management invalid-authentication-entry-timeout 30
set services user-identification identity-management ip-query query-delay-time 15
set services user-identification authentication-source aruba-clearpass user-query delay-query-time 15
```

**Step-by-Step Procedure**

To configure JIMS and ClearPass at the same time, use the following configurations:

1. Configure the IP address of the primary JIMS server.

   ```
   [edit services]
   user@host# set user-identification identity-management connection primary address 192.0.2.0
   ```

2. Configure the client ID that the SRX Series provides to the JIMS primary server as part of its authentication.

   ```
   [edit services]
   user@host# set user-identification identity-management connection primary client-id otest
   ```

3. Configure the client secret that the SRX Series provides to the JIMS primary server as part of its authentication.

   ```
   [edit services]
   user@host# set user-identification identity-management connection primary client-secret test
   ```

4. Configure Aruba ClearPass as the authentication source for user query requests, and configure the ClearPass webserver name and its IP address. The SRX Series device requires this information to contact the ClearPass webserver.

```
[edit services]
user@host# set user-identification authentication-source aruba-clearpass user-query web-server cp-server
   address 198.51.100.0
```

5. Configure the client ID and the client secret that the SRX Series device requires obtaining an access token required for user queries.

```
[edit services]
user@host# set user-identification authentication-source aruba-clearpass user-query client-id otest
user@host# set user-identification authentication-source aruba-clearpass user-query client-secret test
```

6. Configure the token API that is used in generating the URL for acquiring an access token.

```
[edit services]
user@host# set user-identification authentication-source aruba-clearpass user-query token-api
   oauth_token/oauth
```

7. Configure the query API to use for querying individual user authentication and identity information.

```
[edit services]
user@host# set user-identification authentication-source aruba-clearpass user-query query-api
   "user_query/v1/ip/$IP$"
```

8. Configure the Web API daemon username and password for the account.

```
[edit system services]
user@host# set webapi user user password “$ABC123"
```

9. Configure the Web API client address–that is, the IP address of the ClearPass webserver’s data port.

```
[edit system services]
user@host# set webapi client 203.0.113.0
```

10. Configure the Web API process HTTPS service port.

```
[edit system services]
user@host# set webapi https port 8443
user@host# set webapi https default-certificate
```

11. Configure an authentication entry timeout value for Aruba ClearPass.

```
[edit services]
user@host# set user-identification authentication-source aruba-clearpass invalid-authentication-entry-timeout
    30
```

12. Configure an independent timeout value to be assigned to invalid user authentication entries in the SRX Series authentication table for Aruba ClearPass.

```
[edit services]
user@host# set user-identification identity-management authentication-entry-timeout 30
```

13. Configure an independent timeout value to be assigned to invalid user authentication entries in the SRX Series authentication table for JIMS.

```
[edit services]
user@host# set user-identification identity-management invalid-authentication-entry-timeout 30
```

14. Set a **query-delay-time** parameter, specified in seconds, that allows the SRX Series device to wait for a period of time before sending the query.

```
[edit services]
user@host# set user-identification identity-management ip-query query-delay-time 15
```

15. Set a **query-delay-time** parameter, specified in seconds, that allows the SRX Series device to wait for a period of time before sending the query.

```
[edit services]
user@host# set user-identification authentication-source aruba-clearpass user-query delay-query-time 15
```

***Results***

From configuration mode, confirm your configuration by entering the **show system services webapi**, command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit ]
user@host# show system services webapi
user {
    device;
    password "$ABC123"; ## SECRET-DATA
}
client {
    203.0.113.0;
}
https {
    port 8443;
    default-certificate;
}
```

From configuration mode, confirm your configuration by entering the **show services user-identification authentication-source aruba-clearpass** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit ]
user@host# show services user-identification authentication-source aruba-clearpass
authentication-entry-timeout 30;
invalid-authentication-entry-timeout 30;
user-query {
    web-server {
        cp-server;
        address 10.208.164.31;
    }
    client-id otest;
    client-secret "$ABC123"; ## SECRET-DATA
    token-api oauth_token/oauth;
    query-api "user_query/v1/ip/$IP$";
    delay-query-time 15;
}
```

From configuration mode, confirm your configuration by entering the **show services user-identification identity-management** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit ]
user@host# show services user-identification identity-management
authentication-entry-timeout 30;
invalid-authentication-entry-timeout 30;
      connection {
         primary {
            address 10.208.164.137;
            client-id otest;
            client-secret "$ABC123"; ## SECRET-DATA
         }
      }
      ip-query {
         query-delay-time 15;
      }
```

If you are done configuring the devices, enter **commit** from configuration mode.

## Verification

**IN THIS SECTION**


- Verifying JIMS Authentication Entries | **385**
- Verifying ClearPass Authentication Entries | **386**
- Verifying Device Entries by Domain | **387**
- Verifying ClearPass Webserver Is Online | **387**
- Verifying JIMS Server Is Online | **388**


Confirm that the configuration is working properly.

### *Verifying JIMS Authentication Entries*

**Purpose**

Verify that the device identity authentication table for JIMS is updated.

**Action**

Enter the **show services user-identification authentication-table authentication-source identity-management source-name "JIMS - Active Directory" node 0** command.

```
show services user-identification authentication-table authentication-source
identity-management source-name "JIMS - Active Directory" node 0
```

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad-jims-2008.com
Total entries: 5
Source IP       Username       groups(Ref by policy)          state
192.0.2.2   administrator  dow_group_00001,dow_group_0000 Valid
192.0.2.4   administrator  dow_group_00001,dow_group_0000 Valid
192.0.2.5   administrator  dow_group_00001,dow_group_0000 Valid
192.0.2.7   administrator  dow_group_00001,dow_group_0000 Valid
192.0.2.11  administrator  dow_group_00001,dow_group_0000 Valid
```

**Meaning**

The output displays the authentication entries are updated.

***Verifying ClearPass Authentication Entries***

**Purpose**

Verify that the device identity authentication table for ClearPass is updated.

**Action**

Enter the **show services user-identification authentication-table authentication-source aruba-clearpass node 0** command to verify that entries are updated.

```
show services user-identification authentication-table authentication-source
aruba-clearpass node 0
```

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: juniper.net
Total entries: 1
Source IP                 Username    groups(Ref by policy) state
2001:db8::::63bf:3fff:fdd2 ipv6_user01 ipv6_group1           Valid
```

**Meaning**

The output displays the authentication entries are getting updated for ClearPass.

*Verifying Device Entries by Domain*

**Purpose**

Verify that all authenticated devices belong to the domain.

**Action**

Enter the **show services user-identification device-information table all domain juniper.net node 0** command.

```
show services user-identification device-information table all domain
juniper.net node 0
```

```
node0:
--------------------------------------------------------------------------
Domain: juniper.net
Total entries: 1
Source IP                              Device ID Device-Groups
2001:db8:4136:e378:8000:63bf:3fff:fdd2 dev01 device_group1
```

**Meaning**

The output displays all authenticated devices that belong to the domain.

*Verifying ClearPass Webserver Is Online*

**Purpose**

Verify that the ClearPass webserver is online.

**Action**

Enter the **show services user-identification authentication-source aruba-clearpass user-query status** command.

```
show services user-identification authentication-source aruba-clearpass
user-query status
```

```
node1:
--------------------------------------------------------------------------
Authentication source: aruba-clearpass
    Web server Address: 198.51.100.0
    Status: Online
    Current connections: 0
```

**Meaning**

The output displays the ClearPass webserver is online.

***Verifying JIMS Server Is Online***

**Purpose**

Verify that the JIMS server is online.

**Action**

Enter the **show services user-identification identity-management status** command.

```
show services user-identification identity-management status
```

```
node1:
--------------------------------------------------------------------------
Primary server :
    Address                     : 192.0.2.0
    Port                        : 443
    Connection method           : HTTPS
    Connection status           : Online
Secondary server :
    Address                     : 192.0.2.1
    Port                        : 443
    Connection method           : HTTPS
    Connection status           : Offline
    Last received status message : OK (200)
    Access token                : P1kAlMiG2Kb7FzP5tM1QBI6DSS92c31Apgjk9lV
    Token expire time           : 2018-04-12 06:57:37
```

**Meaning**

The output displays the JIMS server is online.

**Release History Table**

| Release | Description |
|---|---|
| 18.2R1 | Starting in Junos OS Release 18.2R1, you can configure JIMS, ClearPass, and Web API at the same time in UserFW. Prior to Junos OS Release 18.2R1, you can either configure ClearPass Policy Manager (CPPM) or JIMS. |

CHAPTER 7

# Configuration Statements


actions (Services SSL Proxy) | **397**

active-directory-access | **400**

active-directory-authentication-table | **402**

address (Services) | **403**

address (Services User Identification) | **404**

address (Identity Management Advanced Query Primary) | **405**

address (Identity Management Advanced Query Secondary) | **407**

admin-search | **409**

allow-reverse-ecmp | **410**

application (Security Policies) | **411**

application-services (Security Policies) | **413**

assemble | **415**

auth-only-browser | **416**

auth-user-agent | **418**

authentication-entry-timeout (Services User Identification) | **419**

authentication-entry-timeout (Identity Management Advanced Query) | **420**

authentication-source (Services User Identification ClearPass) | **422**

authentication-source (Services User Identification Device Identity) | **424**

batch query | **426**

banner (Access FTP HTTP Telnet Authentication) | **429**

banner (Access Web Authentication) | **430**

base-distinguished-name | **431**

ca-certificate (Services User Identification) | **432**

ca-certificate (Identity Management Advanced Query Primary) | **433**

ca-certificate (Identity Management Advanced Query Secondary) | **435**

ca-profile (Services) | **437**

captive-portal (Services UAC) | **438**

captive-portal (Services UAC Policy) | **439**

certficate (System Services) | **440**

certificate-key (System Services) | **441**

certificate-verification | **442**

client (System Services) | **444**

client-id (Services User Identification) | **445**

client-id (Identity Management Advanced Query Primary) | **446**

client-id (Identity Management Advanced Query Secondary) | **448**

client-group | **450**

client-idle-timeout (Access Profile) | **451**

client-name-filter | **452**

client-secret (Services User Identification) | **453**

client-secret (Identity Management Advanced Query Primary) | **454**

client-secret (Identity Management Advanced Query Secondary) | **456**

client-session-timeout (Access Profile) | **458**

configuration-file | **459**

connection (Identity Management Advanced Query) | **460**

connect-method (Identity Management Advanced Query) | 464

connect-method (Services User Identification) | 466

count | 467

custom-ciphers | 468

debug-level (System Services) | 471

debug-log (System Services) | 472

default-certificate (System Services) | 473

default-profile | 474

delay-query-time (Services User Identification) | 475

distinguished-name (Access) | 476

domain-name (Access Profile) | 477

enable-flow-tracing (Services) | 478

enable-session-cache | 479

end-user-profile | 480

fail | 482

file (Services User Identification) | 483

file (System Logging) | 484

filter (Security) | 486

filter (Identity Management Advanced Query) | 487

firewall-authentication | 491

firewall-authentication (Security) | 493

firewall-authentication (Security Policies) | 495

firewall-authentication (User Identification) | 497

firewall-authentication-forced-timeout | 498

firewall-authentication-service | 500

firewall-user | 501

flag (Services) | 502

from-zone (Security Policies) | 503

ftp (Access) | 507

group-profile (Access) | **508**

http (Access) | **510**

http (Services) | **511**

http (Services User Identification) | **512**

http (System Services) | **513**

https (Services) | **514**

https (Services User Identification) | **515**

https (System Services) | **517**

identity-management | **519**

infranet-controller | **523**

interface (Services) | **524**

interval (Services) | **525**

invalid-authentication-entry-timeout (Services User Identification Active Directory and ClearPass) | **526**

ip-address (Access Profile) | **528**

ip-query (Identity Management Advanced Query) | **529**

ip-user-mapping | **531**

ldap-options | **533**

ldap-server | **534**

level (Services) | **535**

level (Services User Identification) | **536**

lifetime-seconds (Security IKE) | **537**

link (Access) | **538**

local-authentication-table | **539**

log (Services) | **540**

login (Access) | **541**

nas-port-type | **542**

network (Access) | **543**

no-remote-trace (Services User Identification) | **544**

no-user-query (Services User Identification) | **545**

no-tls-certificate-check | **546**

pass-through | **547**

password (Access) | **549**

password (Services) | **550**

password (System Services) | **551**

permit (Security Policies) | **552**

pki-local-certificate (Services) | **554**

policies | **555**

pool (Access) | **566**

port (Access LDAP) | **568**

port (Identity Management Advanced Query) | **569**

port (Services) | **571**

port (System Services) | **572**

preferred-ciphers | **573**

prefix (Access IPv6) | **574**

primary connection (Identity Management Advanced Query) | **575**

priority (Security User Identification) | **578**

push-to-identity-management | **580**

protocol-version | **582**

query-api (Services User Identification) | **583**

query-api (advanced user query) | **585**

radius-options (Access) | **587**

radius-server (Access) | **588**

range (Access) | **589**

rate-limit (Security Log) | **590**

redirect-traffic | **591**

redirect-url | **592**

retry (Access LDAP) | **594**

retry (Access RADIUS) | **595**

revert-interval (Access LDAP) | **596**

revert-interval (Access RADIUS) | **597**

root-ca (Services) | **598**

routing-instance (Access LDAP) | **599**

routing-instance (Access RADIUS) | **600**

search | **601**

search-filter | **602**

secondary connection (Identity Management Advanced Query) | **603**

secret (Access Profile) | **605**

securid-server | **606**

separator | **607**

server-certificate (Services) | **608**

server-certificate-subject | **609**

session-options (Access Profile) | **610**

size (Services) | **611**

source-address (Access LDAP) | **612**

source-address (Access RADIUS) | **613**

source-end-user-profile | **614**

source-identity-log (Security) | **616**

ssl (Services) | **618**

ssl-termination-profile | **621**

success | **622**

system-generated-certificate | **623**

telnet (Access) | **624**

termination (Services) | **625**

test-only-mode | **626**

then (Security Policies) | **627**

timeout (Access LDAP) | **630**

timeout (Access RADIUS) | **631**

timeout (Services) | 632

timeout-action | 633

tls-min-version | 634

tls-peer-name | 635

tls-timeout | 636

tls-type | 637

token-api (Services User Identification) | 638

token-api | 639

to-zone (Security Policies) | 641

traceoptions (Access) | 645

traceoptions (Active Directory Access) | 647

traceoptions (Security Firewall Authentication) | 649

traceoptions (Services SSL) | 651

traceoptions (Services UAC) | 654

traceoptions (Services User Identification) | 656

trusted-ca (Services) | 657

uac-policy (Application Services) | 658

uac-service | 659

unified-access-control (Security) | 660

unified-access-control (Services) | 661

user-group-mapping | 663

user-identification (Services) | 665

webapi (System Services) | 670

webapi-clear-text (Security) | 671

webapi-ssl (Security) | 672

web-authentication | 673

web-authentication (Access) | 674

web-authentication (Interfaces) | 676

web-management (System Services) | 677

web-redirect | **681**

web-redirect-to-https | **682**

web-server (Services) | **683**

whitelist (Services) | **684**

wins-server (Access) | **685**

# actions (Services SSL Proxy)

### Syntax

```
actions {
    crl {
        disable;
        if-not-present (allow | drop);
        ignore-hold-instruction-code;
    }
    disable-session-resumption;
    ignore-server-auth-failure;
    logs {
        all;
        errors;
        info;
        sessions-allowed;
        sessions-dropped;
        sessions-ignored;
        sessions-whitelisted;
        warning;
    }
    renegotiation {
        (allow | allow-secure | drop);
    }
}
```

### Hierarchy Level

```
[edit services ssl proxy profile profile-name]
```

### Release Information

Statement introduced in Junos OS Release 12.1X44-D10. The **crl** statement is supported from Junos OS Release 15.1X49-D30.

### Description

Specify the logging and traffic related actions for a SSL proxy profile.

An SSL proxy profile is required to configure SSL proxy on your SRX Series device. As a part of the proxy profile configuration, you can configure– actions related to certification revocations checks, options to specify if a change in SSL parameters requires renegotiation for a session, option to disable session resumption, option to ignore certificate validation, root CA expiration dates, and other such issues based on your requirements.

**Options**

- **crl**—Specify the certificate revocation actions.
  - **disable**—Disable CRL verification.
  - **if-not-present**—Specify actions for sessions.
    - **allow**—Allow sessions when CRL information is not available.
    - **drop**—Drop sessions when CRL information is not available.
  - **ignore-hold-instruction-code**—Ignore the unconfirmed (on hold) revocation status, and accept a certificate.
- **disable-session-resumption**—Disable session resumption.
- **ignore-server-auth-failure**—Ignore server authentication failure.
- **log**—Specify the logging actions.
  - **all**—Log all events.
  - **errors**—Log all error events.
  - **info**—Log all information events.
  - **sessions-allowed**—Log SSL session allowed events after an error.
  - **sessions-dropped**—Log only SSL session dropped events.
  - **sessions-ignored**—Log session ignored events.
  - **sessions-whitelisted**—Log SSL session whitelisted events.
  - **warning**—Log all warning events.
- **renegotiation**—Specify the renegotiation options.
  - **allow**—Allow secure and nonsecure renegotiation.
  - **allow-secure**—Allow secure negotiation only.
  - **drop**—Drop session on renegotiation request.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

SSL Proxy Overview | 96

Configuring SSL Forward Proxy | 99

Enabling Debugging and Tracing for SSL Proxy | 110

# active-directory-access

**Syntax**

```
active-directory-access {
    domain domain-name {
        user username;
            password password;
        domain-controller domain-controller-name {
            address domain-controller-address;
        }
        ip-user-mapping {
            discovery-method {
                wmi {
                    event-log-scanning-interval seconds;
                    initial-event-log-timespan hours;
                }
            }
        }
        user-group-mapping {
            ldap {
                authentication-algorithm {
                    simple;
                }
                ssl;
                base base;
                user name {
                    password password;
                }
                address ip-address {
                    port port;
                }
            }
        }
    }
}
```

**Hierarchy Level**

```
[edit services user-identification]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X47-D10.

**Description**

Identify the domain and domain controllers where the integrated user firewall feature is implemented; configure the IP address-to-user mapping information and the user-to-group mapping information for accessing the LDAP server.

**Options**

**domain *domain-name***—Required. Name of the domain; the length of the name ranges from 1 through 64 characters. The SRX Series device can have the integrated user firewall feature configured in a maximum of two domains.

**user *username***—Required. Active Directory account name.

**Range:** 1 through 64 characters.

**password *password***—Required. Password of the Active Directory account.

**Range:** 1 through 128 characters.

**domain-controller *domain-controller-name***—Required. Name of the domain controller; the length of the name can range from 1 through 64 characters. A maximum of 10 domain controllers can be configured.

**address *domain-controller-address***—Required. IP address of the domain controller.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

user-identification (Services) | **665**

LDAP Functionality in Integrated User Firewall | **194**

# active-directory-authentication-table

**Syntax**

```
active-directory-authentication-table {
    priority priority;
}
```

**Hierarchy Level**

```
[edit security user-identification authentication-source]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X47-D10.

**Description**

An authentication table is generated by polling Active Directory domain controllers for source identity information about active users. Each entry in the table correlates an authenticated user with an IP address and associated user groups. That information is used for matching in IP-based firewall policies. The user information must be retrieved from the table before policy lookup can proceed and traffic is allowed to pass through the firewall.

Revision History

**Options**

**priority *priority***—Specify the priority of the Active Directory authentication table. The priority determines the sequence for searching among various other authentication tables to retrieve a user role. The priorities of the following tables are considered: local authentication table, firewall authentication table, Active Directory authentication table, and UAC authentication table.

Each authentication table is given a unique priority value. The lower the value, the higher the priority. A table with priority 120 is searched before a table with priority 200. Setting the priority value of a table to 0 is equivalent to disabling the table and eliminating it from the search sequence.

**Range:** A unique value from 0 through 65535.

**Default:** The default priority of the Active Directory authentication table is 125.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

*authentication-source (Security)*

Overview of Integrated User Firewall | 177

*Understanding User Role Firewalls*

*Understanding the User Identification Table*

# address (Services)

**Syntax**

```
address ip-address;
```

**Hierarchy Level**

```
[edit services unified-access-control infranet-controller hostname]
```

**Release Information**

Statement introduced in Junos OS Release 9.4.

**Description**

Specify the IP address of the IC Series device with which the SRX Series devices should communicate.

This statement is required when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Unified Access Control | 151

Acquiring User Role Information from an Active Directory Authentication Server | 151

# address (Services User Identification)

**Syntax**

```
address (ip-address | hostname);
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query web-server]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure for the integrated ClearPass authentication and enforcement feature the address of the ClearPass webserver that the SRX Series device communicates with. The SRX Series device requests user authentication and identity information for an individual user from the ClearPass webserver whose address is configured. If you configure the user query function, the SRX Series device can obtain this information for a specific user when it does not receive it from the ClearPass Policy Manager through Web API POST requests.

**Required Privilege Level**

services—To view this statement in the configuration

services-control—To add this statement to the configuration.

# address (Identity Management Advanced Query Primary)

**Syntax**

```
address ip-address;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection primary],
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the IP address for the primary Juniper Identity Management Service (JIMS) server. The SRX Series device requires the server IP address to connect to the server to obtain an access code that allows it to query the server for user identity information. The IP address is configured as part of a collection of information which includes the SRX Series device's client ID, client secret, and ca-certificate information.

You configure separate sets of information for connection to the primary server and the secondary server. The SRX Series device queries the primary server first. When the primary one fails, it queries the secondary server. You configure the SRX Series device to connect to the secondary server separately.

**NOTE:** This feature supports only IPv4 addresses.

Juniper Identity Management Service uses the credentials grant access token process, which requires use of OAuth2 to authenticate and authorize access to it by the SRX Series device. (See RFC 6749.) Prior to querying the primary server, the SRX Series device must go through a process which, as mentioned previously, entails obtaining an access code.

The Juniper Identity Management Service server must authenticate the SRX Series device before it allows the SRX Series device to query it for user identity information. When the SRX Series device connects to the Juniper Identity Management Service server, it sends the server its identifying client ID, client secret, and ca-certificate. This information must be consistent with the API client configured on the Juniper Identity Management Service primary server to which it authorizes.

**NOTE:** The SRX Series device sends a unique set of identification information to the primary server and the secondary server.

After the SRX Series device is authenticated and issues queries for user identity information, it inserts that information into its authentication table. When a user requests access to a protected resource, the SRX Series device authenticates the user based on the authentication table entry for that user.

The Juniper Identity Management Service provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting. It provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as device identity, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- filter | 487
- port | 569
- primary | 575
- query-api | 585
- secondary | 603
- token-api | 639
- invalid-authentication-entry-timeout | 526

# address (Identity Management Advanced Query Secondary)

**Syntax**

```
address ip-address;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection secondary]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the IP address for the secondary Juniper Identity Management Service (JIMS) server. The SRX Series device requires the server IP address to connect to the server to obtain an access code that allows it to query the server for user identity information. The IP address is configured as part of a collection of information which includes the SRX Series device's client ID, client secret, and ca-certificate information.

The SRX Series device uses the secondary server when the primary one fails. You configure the SRX Series device to connect to the primary server separately.

**NOTE:** This feature supports only IPv4 addresses.

Juniper Identity Management Service uses the credentials grant access token process, which requires use of OAuth2 to authenticate and authorize access to it by the SRX Series device. (See RFC 6749.) Prior to querying the secondary server, the SRX Series device must go through a process which, as mentioned previously, entails obtaining an access code. The Juniper Identity Management Service server must authenticate the SRX Series device before it allows the SRX Series device to query it for user identity information. When the SRX Series device connects to the Juniper Identity Management Service server, it sends the server its identifying client ID, client secret, and ca-certificate. This information must be consistent with the API client configured on the Juniper Identity Management Service secondary server.

**NOTE:** The SRX Series device sends a unique set of identification information to the primary server and the secondary server.

After the SRX Series device is authenticated and issues queries for user identity information, it inserts that information into its authentication table. When a user requests access to a protected resource, the SRX Series device authenticates the user based on the authentication table entry for that user.

The Juniper Identity Management Service provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting. It provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as device identity, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable any other actively used options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication and the ClearPass query and device-id functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275

address | 405

authentication-entry-timeout | 420

batch-query | 426

ca-certificate | 433

client-id | 446

# admin-search

**Syntax**

```
admin-search {
    distinguished-name distinguished-name;
    password password;
}
```

**Hierarchy Level**

```
[edit access ldap-options search],
[edit access profile  profile-name  ldap-options search]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify that a Lightweight Directory Access Protocol (LDAP) administrator search is performed. To perform an administrator search, you must specify administrator credentials, which are used in the bind as part of performing the search.

**Options**

The remaining statements are explained separately.

**Default:** Anonymous search. To perform an administrator search, you must specify administrator credentials, which are used in the bind as part of performing the search.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

# allow-reverse-ecmp

**Syntax**

```
allow-reverse-ecmp
```

**Hierarchy Level**

```
[edit security flow]
```

**Release Information**
Statement introduced in Junos OS Release 17.3.

**Description**
Enable ECMP support for reverse traffic. In this case, Junos OS for SRX Series devices and vSRX instances use a hash algorithm to determine the interface to use for reverse traffic in a flow. This process is similar to asymmetric routing in which a packet traverses from a source to a destination in one path and takes a different path when it returns to the source.

If you do not enable this feature, the software selects a route in the ECMP set to the incoming interface for reverse traffic, which is the default behavior.

**Required Privilege Level**
security—To view this in the configuration.
security-control—To add this to the configuration.

## RELATED DOCUMENTATION

- *Understanding ECMP Flow-Based Forwarding*
- *Understanding ECMP Flow-Based Forwarding for Reverse Traffic on SRX Series Devices and vSRX Instances*

# application (Security Policies)

**Syntax**

```
application {
    [application];
    any;
}
```

**Hierarchy Level**

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* match]

[edit security policies global policy *policy-name* match]

**Release Information**

Statement introduced in Junos OS Release 8.5.

**Description**

Specify the IP or remote procedure call (RPC) application or set of applications to be used as match criteria.

Starting in Junos OS Release 19.1R1, configuring the **application** statement at the **[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* match]** hierarchy level is optional if the **dynamic-application** statement is configured at the same hierarchy level.

**Options**

***application-name-or-set***—Name of the predefined or custom application or application set used as match criteria.

**any**—Any predefined or custom applications or application sets.

**NOTE:** A custom application that does not use a predefined destination port for the application will not be included in the **any** option, and must be named explicitly.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Security Policies Overview*

*Configuring Applications in Unified Policies*

# application-services (Security Policies)

### Syntax

```
application-services {
    advanced-anti-malware-policy advanced-anti-malware-policy;
    application-firewall {
        rule-set rule-set;
    }
    application-traffic-control {
        rule-set rule-set;
    }
    gprs-gtp-profile gprs-gtp-profile;
    gprs-sctp-profile gprs-sctp-profile;
    idp idp;
    (redirect-wx redirect-wx | reverse-redirect-wx reverse-redirect-wx);
    security-intelligence-policy security-intelligence-policy;
    ssl-proxy {
        profile-name profile-name;
    }
    uac-policy {
        captive-portal captive-portal;
    }
    utm-policy utm-policy;
    web-proxy {
        profile-name profile-name;
    }
}
```

### Hierarchy Level

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit]
```

### Release Information

Statement modified in Junos OS Release 11.1. The **web-proxy** option is introduced in Junos OS Release 19.2R1.

### Description

Enable application services within a security policy. You can enable service such as application firewall, IDP, UTM, SSL proxy, and so on by specifying them in a security policy permit action, when the traffic matches the policy rule.

### Options

**advanced-anti-malware-policy**—Specify advanced-anti-malware policy name.

**application-firewall**—Specify the rule sets configured as part of application firewall to be applied to the permitted traffic.

**application-traffic-control**—Specify the rule sets configured as part of AppQoS, application-aware quality of service, to be applied to the permitted traffic.

**gprs-gtp-profile**—Specify GPRS tunneling protocol profile name.

**gprs-sctp-profile**—Specify GPRS stream control protocol profile name.

**idp**—Apply Intrusion detection and prevention (IDP) as application services.

**redirect-wx**—Specify the WX redirection needed for the packets that arrive from the LAN.

**reverse-redirect-wx**—Specify the WX redirection needed for the reverse flow of the packets that arrive from the WAN.

**security-intelligence-policy**—Specify security-intelligence policy name.

**uac-policy** —Enable Unified Access Control (UAC) for the security policy. This statement is required when you are configuring the SRX Series device to act as a Junos OS Enforcer in a UAC deployment.

**captive-portal *captive-portal***—Specify the preconfigured security policy for captive portal on the Junos OS Enforcer to enable the captive portal feature. The captive portal policy is configured as part of the UAC policy. By configuring the captive portal feature, you can redirect traffic destined for protected resources to the IC Series device or to the URL you configure on the Junos OS Enforcer.

**utm-policy *utm-policy***—Specify UTM policy name. The UTM policy configured for antivirus, antispam, content-filtering, traffic-options, and Web-filtering protocols is attached to the security policy to be applied to the permitted traffic.

**web-proxy *profile-name***—Specify secure Web proxy profile name. The secure Web proxy profile is configured with dynamic application and external proxy server details. This profile is attached to the security policy and applied on the permitted traffic.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Application Firewall Overview*

# assemble

### Syntax

```
assemble {
    common-name common-name;
}
```

### Hierarchy Level

```
[edit access ldap-options],
[edit access profile profile-name ldap-options]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify that a user's LDAP distinguished name (DN) is assembled through the use of a common name identifier, the username, and base distinguished name.

### Options

**common-name *common-name***—Common name identifier used as a prefix for the username during the assembly of the user's distinguished name. For example, **uid** specifies " user id," and **cn** specifies "common name."

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

# auth-only-browser

**Syntax**

```
auth-only-browser <auth-user-agent [user-agent] >;
auth-only-browser;
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    pass-through]
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    user-firewall]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D90.

**Description**

Configure firewall authentication to ignore non-browser HTTP/HTTPS traffic. This feature allows you to ensure that unauthenticated users issuing access requests through HTTP/HTTPS browsers are presented with a captive portal interface to allow them to authenticate. By default, firewall authentication responds to all HTTP/HTTPS traffic.

It can happen that non-browser HTTP/HTTPS services running in the background can trigger captive portal authentication, creating a race condition that suppresses presentation of the captive portal interface to the HTTP/HTTPS browser user.

When **auth-only-browser** is configured, non-browser HTTP traffic is dropped to allow for captive portal to be presented to unauthenticated users who request access using a browser.

**Options**

**auth-user-agent *user-agent***—Allow the SRX Series device to use the user-agent strings that you specify to verify that the browser traffic is HTTP/HTTPS traffic. Firewall authentication checks the strings against the User-Agent field in the browser header. You can specify only one value for this parameter. It must not contain spaces and it does not need to be enclosed in parenthesis. For example, **auth-user-agent** might specify Opera1 as one of its values.

You can use the **auth-user-agent** parameter alone for pass-through or user-firewall authentication or in conjunction with **auth-only-browser**.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

auth-user-agent | 418

Understanding SRX Series Assured Captive Portal Support for Unauthenticated Browser Users | 223

# auth-user-agent

**Syntax**

```
auth-user-agent [user-agent];
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    pass-through]
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    pass-through auth-only-browser]
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    user-firewall]]
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    user-firewall auth-only-browser]
```

**Description**

Specify a user-agent value to be used to verify that the user's browser traffic is HTTP/HTTPS traffic. Firewall authentication checks the value against the User-Agent field in the browser header. For example, the **auth-user-agent** parameter might specify Opera1 to be verified against the brower's User-Agent field for a match.

You can use the **auth-user-agent** parameter alone for pass-through or user-firewall authentication or in conjunction with **auth-only-browser**.

The **auth-only-browser** directs firewall authentication to ignore non-browser HTTP/HTTPS traffic to ensure that unauthenticated users using an HTTP/HTTPS browser are authenticated by captive portal before they are granted access to protected resources. It can happen that non-browser HTTP/HTTPS services running in the background can trigger captive portal authentication creating a race condition that suppresses presentation of the captive portal interface to the HTTP/HTTPS browser user.

**Options**

***user-agent***—A string to be matched against values specified in the browser's User-Agent header field that identifies the traffic as HTTP/HTTPS browser traffic. You can specify only one user-agent value for a security policy configuration. The value must not contain spaces. You do not need to enclose the string in parenthesis. The length of a string must be 17 characters or less.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

auth-only-browser | 416

Understanding SRX Series Assured Captive Portal Support for Unauthenticated Browser Users | 223

# authentication-entry-timeout (Services User Identification)

**Syntax**

```
authentication-entry-timeout minutes;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure for the integrated ClearPass authentication and enforcement feature the timeout interval after which idle entries in the ClearPass authentication table expire.

**Options**

***minutes***––Timeout interval. The timout interval begins from when the user authentication entry is added to the ClearPass authentication table. If a value of 0 is specified, the entries will never expire.

**Range:** 10 through 1440 minutes

**Default:** 30 minutes

**Required Privilege Level**

services—To view this statement in the configuration

services-control—To add this statement to the configuration.

# authentication-entry-timeout (Identity Management Advanced Query)

**Syntax**

```
authentication-entry-timeout time-out-in-minutes;
```

**Hierarchy Level**

```
[edit services user-identification identity-management]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the time-out for the user identity authentication entries. You configure this parameter as part of the advanced user identity query feature for SRX Series devices.

The advanced user identity query feature for SRX Series devices relies on the Juniper Identity Management Service (JIMS), a centralized identity collection (CIC) system from which the SRX Series device obtains the user identity information. It provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Options**

***time-out-in-minutes***—The amount of time after which a user identity authentication entry expires.

**Range:** 0 or 10 through 1440 minutes. Specification of 0 indicates no time-out.

**Default:** 60 minutes

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- address | 407
- batch-query | 426
- ca-certificate | 435
- client-id | 446
- connect-method | 464
- filter | 487
- query-api | 585
- secondary | 603
- token-api | 639

# authentication-source (Services User Identification ClearPass)

**Syntax**

```
authentication-source aruba-clearpass {
    authentication-entry-timeout minutes;
    no-user-query;
    traceoptions {
          file {
             filename;
             files number;
             match regular-expression;
             size maximum-file-size;
             (world-readable |no-world-readable);
          }
          flag flag;
          level level ;
          no-remote-trace;
       }
       user-query {
          web-server {
          servername;
          connect-method https|http;
          address  server-address;
          port port -number;
       }
    }
}
```

**Hierarchy Level**

```
[edit services user-identification]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure ClearPass as the authentication source for the integrated ClearPass authentication and enforcement feature.

The ClearPass Policy Manager (CPPM), as the authentication source and client of the SRX Series device HTTP server, initiates a connection to the SRX Series device using the Web API that the SRX Series device

exposes to it. The CPPM sends user authentication and identity information to the SRX Series device across this connection using HTTP or HTTPS POST request messages.

The remaining statements are explained separately. See CLI Explorer.

> **NOTE:** **set authentication-source aruba-clearpass** command can be used to configure the Juniper Identity Management Service as the authentication-source.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# authentication-source (Services User Identification Device Identity)

**Syntax**

```
authentication-source (active-directory | network-access-controller)
```

**Hierarchy Level**

```
[edit services user-identification device-information]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D70.

**Description**
Specify the device identity authentication source. The integrated user firewall device identity authentication feature enables you to control access to resources based on the identity of the device and not that of the user of the device. Supported authentication sources include Active Directory and third-party network access systems.

The SRX Series device obtains the device identity information for authenticated devices from the authentication source. After the SRX Series device obtains the device information, it creates a device identity authentication table to use to store device identity entries.

The SRX Series device searches the device identity authentication table for a device match when traffic issuing from a user's device arrives at the SRX Series device. If it finds a match, the SRX Series device searches for a matching security policy. If it finds a matching security policy, the security policy's action is applied to the traffic.

**Options**
**active-directory**—Specifies Microsoft Active Directory as the authentication source.

The SRX Series device obtains the device identity information for authenticated devices from Active Directory. It reads the Active Directory domain controller event logs to obtain the IP addresses of devices logged into the domain and authenticated by Windows. Then, for each authenticated device, it obtains from the Active Directory LDAP server the names of the groups to which the device belongs, based on the IP addresses of the devices.

**network-access-controller**—Specifies the authentication source as that of a third-party network access controller (NAC) system. If your network environment is configured for a NAC solution and you decide to take this approach, the NAC system sends the device identity information to the SRX Series device. The SRX Series device exposes a RESTful Web services API implementation that enables you to send

the device identity information to the SRX Series device in a formal XML structure. If you take this approach, you must verify that your NAC solution works with the SRX Series device.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Access Control to Network Resources Based on Device Identity Information | 244
- Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature | 247
- Understanding the Device Identity Authentication Table and Its Entries | 252

# batch query

### Syntax

```
batch-query {
    items-per-batch items-per-batch;
    query-interval seconds;
}
```

### Hierarchy Level

```
[edit services user-identification identity-management]
```

### Release Information

Statement introduced in Junos OS Release 15.1X49-D100.

### Description

Configure the SRX Series device to communicate with the Juniper Identity Management Service server to obtain an access token to use to query the server for identity information for an individual user (IP query and user query) or a group of users (batch query). The access token allows the SRX Series device to connect to the Juniper Identity Management Service server to query it for this information.

The batch-query statement allows the SRX Series device to periodically query the Juniper Identity Management Service server automatically for user identity information. When you start the SRX Series device, it automatically sends a batch query request to the Juniper Identity Management Service server to obtain all of the user identity information that it expects. After it receives the user identity information, the SRX Series device periodically issues a query to the Juniper Identity Management Service server requesting that a new report be generated to include any newly available user identity items so as to keep its authentication table entries up-to-date.

You can configure an interval for when the batch query request is to be issued and the maximum number of user identity items to be sent in response to the query in one batch. Only remaining available user identity items are sent if their number is fewer than the configured maximum.

**NOTE:** If you need to refresh the user identities in the authentication table—that is, everything that was received automatically when you started the system and from subsequent batch queries or IP queries—you can clear the authentication table by disabling the user-identification feature configuration. Afterward, you can reconfigure the advanced-query feature to retrieve all available user identities. To accomplish this, you use the following sequence of CLI statements: deactivate services user-identification, commit, activate services user-identification, commit.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the [edit services user-identification] hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

The advanced query feature queries the Juniper Identity Management Service for user identification information that the SRX Series stores in its authentication table and uses to authenticate users. Use of the Juniper Identity Management Service allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

To obtain device information, such as device identity, groups, and the operating system, from the Juniper Identity Management Service server using either the batch-query or ip-query configuration, you must set the device authentication source, as follows.

```
user@host# set services user-identification device-information authentication-source network-access-controller
```

**Options**

**items-per-batch**—The maximum number of user identity items that the SRX Series device will accept in one batch in response to the query.

**Default: 200**

**Range: 100-1000**

**query-interval.**—Interval in seconds after which the SRX Series device will issue a query request for newly generated user identities.

**Default: 5**

**Range: 1-60**

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275

filter | 487

port | 569

primary | 575

query-api | 585

secondary | 603

token-api | 639

# banner (Access FTP HTTP Telnet Authentication)

### Syntax

```
banner {
    fail string;
    login string;
    success string;
}
```

### Hierarchy Level

```
[edit access firewall-authentication pass-through (ftp | http | telnet)]
```

### Release Information

Statement introduced in Junos OS Release 8.5.
HTTPS for Web authentication is supported on SRX5400, SRX5600, and SRX5800 devices and SRX Series Services Gateways from Junos OS Release 12.1X44-D10 and on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways from Junos OS Release 15.1X49-D40.

### Description

Configure the banners that appear to users during the FTP, HTTP, HTTPS, and Telnet firewall authentication process. The banners appear during login, after successful authentication, and after failed authentication.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Firewall Authentication Banner Customization | 41

# banner (Access Web Authentication)

### Syntax

```
banner {
    success string;
}
```

### Hierarchy Level

```
[edit access firewall-authentication web-authentication]
```

### Release Information

Statement introduced in Junos OS Release 8.5.

### Description

Configure the banner that appears to users during the Web authentication process. The banner appears during login, after successful authentication, and after failed authentication.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

# base-distinguished-name

**Syntax**

```
base-distinguished-name base-distinguished-name;
```

**Hierarchy Level**

```
[edit access ldap-options],
[edit access profileprofile-name ldap-options]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Specify the base distinguished name (DN), which can be used in one of the following ways:

- If you are using the **assemble** statement so that the user's distinguished name is being assembled, the base distinguished name is appended to a username to generate the user's distinguished name. The resulting distinguished name is used in the LDAP bind call.
- If you are using the **search** statement so that the user's distinguished name is found by a search, the search is restricted to the subtree of the base distinguished name.

**Options**
*base-distinguished-name*—Series of basic properties that define the user. For example in the base distinguished name **o=juniper, c=us**, where **c** stands for country, and **o** for organization.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

# ca-certificate (Services User Identification)

**Syntax**

```
ca-certificate certificate-file;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query https]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Specify the certificate file that the SRX Series device uses to verify the Clearpass server's certificate for the SSL connection that is used for the user query function. As the ClearPass administrator, you must export the server's certificate from the CPPM and import it to the SRX Series device. Afterward, you must configure the ca-certificate path and the certificate filename on the SRX Series device. Here is an example:

```
'/var/tmp/RADIUSServerCertificate.crt'
```

This configuration is part of the Integrated ClearPass Authentication and Enforcement feature user query function configuration. User query enables the SRX Series device to query the ClearPass Policy Manager (CPPM) for authentication and identity information for an individual user under certain circumstance when it does not receive that information from the CPPM though the Web API POST requests.

**Required Privilege Level**

**services**—To view this statement in the configuration.

**services-control**—To add this statement to the configuration.

# ca-certificate (Identity Management Advanced Query Primary)

**Syntax**

```
ca-certificate ca-certificate;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection primary]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the file name of the Juniper Identity Management Service's ca-certificate for the primary server. The certificate enables the SRX Series device to verify the identity of Juniper Identity Management Service (JIMS) and that it is trusted for the SSL connection.

Before you configure ca-certificate file name, the administrator of the Juniper Identity Management Services server must export the certificate and import it to the SRX Series device. The administrator must configure the complete path and file name of the certificate where it is installed on the SRX Series device, for example, '/var/db/RADIUSServerCertificate.crt'. If the ca-certificate is not configured, the SRX Series device can not verify the Juniper Identity Management Service certificate.

**NOTE:** The SRX Series device supports a self signed + BASE64 encoded X.509 cert only.

The advanced user identity query feature allows you to obtain user identity information from the Juniper Identity Management Service (JIMS) through queries. It allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

If the configuration entails a primary and a secondary Juniper Identity Management Services server, you configure individual certificates for each of them.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- connect-method | 464
- filter | 487
- port | 569
- primary | 575
- query-api | 585
- token-api | 639
- client-id | 446
- invalid-authentication-entry-timeout | 526

# ca-certificate (Identity Management Advanced Query Secondary)

**Syntax**

```
ca-certificate ca-certificate;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection secondary]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
Configure the file name of the Juniper Identity Management Service's ca-certificate for the secondary server that enables the SRX Series device to verify its identity for the SSL connection and that it is trusted.

Before you configure ca-certificate filename, the administrator of the Juniper Identity Management Services system must export the certificate and import it to the SRX Series device. The administrator must configure the complete path and file name of the certificate where it is installed on the SRX Series device, for example, '/var/db/RADIUSServerCertificate.crt'.

If the ca-certificate is not configured, the SRX Series device will not verify the certificate.

The SRX Series device uses the client credentials grant type access token. For this method, Juniper Identity Management Service requires use of OAuth2 to authenticate and authorize access by the SRX Series device. (See RFC 6749).To authenticate itself to the Juniper Identity Management Service, the SRX Series device must acquire an access token. It must authenticate itself before it can query the server for user identity information.

**NOTE:** The SRX Series device supports a self signed + BASE64 encoded X.509 cert only.

To obtain an access token, the SRX Series device must specify the client secret and the client ID. It must also specify the full path and filename of the ca-certificate, as it was installed on the SRX Series device. All of these values must be consistent with the API client configured on the Juniper Identity Management Service.

You specify a separate set of values for the primary server.

If both a primary server and a secondary server are configured, the SRX Series device always attempts to connect to the primary server first.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- filter | 487
- port | 569
- primary | 575
- query-api | 585
- secondary | 603
- token-api | 639
- invalid-authentication-entry-timeout | 526

# ca-profile (Services)

**Syntax**

```
ca-profile ca-profile;
```

**Hierarchy Level**

```
[edit services unified-access-control infranet-controller hostname]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify the certificate authority (CA) of the certificate that the SRX Series device should use in communications with an Infranet Enforcer. The SRX Series device uses the CA to validate the IC Series UAC Appliance server certificate.

Use this statement if you have loaded certificates from multiple certificate authorities (CAs) onto your SRX Series device and you need to configure the device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series appliance .

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

# captive-portal (Services UAC)

### Syntax

```
captive-portal redirect-policy-name{
    redirect-traffic (all | unauthenticated);
    redirect-url redirect-url;
}
```

### Hierarchy Level

```
[edit services unified-access-control]
```

### Release Information

Statement introduced in Junos OS Release 10.2.

### Description

Specify the preconfigured security policy for captive portal on the Junos OS Enforcer to enable the captive portal feature. The captive portal policy is configured as part of the UAC policy.

By configuring the captive portal feature, you can redirect traffic destined for protected resources to the IC Series device or to the URL you configure on the Junos OS Enforcer.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

# captive-portal (Services UAC Policy)

**Syntax**

```
captive-portal captive-portal-policy-name;
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit application-services
    uac-policy]
```

**Release Information**

Statement introduced in Junos OS Release 10.2.

**Description**

Create the captive portal policy in the UAC security policy. You use the captive portal policy to configure the captive portal feature on the Junos OS Enforcer. By configuring the captive portal feature, you can redirect traffic destined for protected resources to the IC Series device or to the URL you configure on the Junos OS Enforcer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Security Policies Overview*

# certficate (System Services)

**Syntax**

```
certificate certificate-filename;
```

**Hierarchy Level**

```
[edit system services webapi https]
```

**Release Information**
Statement introduced in Junos OS release 12.3X48-D30.

**Description**
Configures a custom certificate to be used for the Integrated ClearPass Authentication and Enforcement feature Web API (webapi) configuration when the HTTPS protocol is configured.

When you configure the Web API (webapi) function to use HTTPS, you can use the default certificate, a custom one, or a certificate generated by the PKI local store.

If you configure a custom certificate, you must configure a certificate key with it. Here is an example of how to configure a certificate and certificate key:

```
set system services webapi https certificate /var/tmp/certificate.crt
set system services webapi https certificate-key /var/tmp/certificate.key
```

**NOTE:** The Web API supports only the PEM format for the custom certificate and certificate key.

**Required Privilege Level**
**system**—To view this statement in the configuration.

**system-control**—To add this statement to the configuration.

# certificate-key (System Services)

**Syntax**

```
certificate-key filename;
```

**Hierarchy Level**

```
[edit system services webapi https]
```

**Release Information**

Statement introduced in Junos OS release 12.3X48-D30.

**Description**

Configures the filename of the certificate key to use with the specified custom certificate for the Web API (webapi) HTTPS configuration. A certificate key is required if a custom certificate file is used.

> **NOTE:** The Integrated ClearPass Authentication and Enforcement feature Web API supports only the PEM format for the custom certificate and certificate key.

**Required Privilege Level**

**system**—To view this statement in the configuration.

**system-control**—To add this statement to the configuration.

# certificate-verification

**Syntax**

```
certificate-verification [ optional | required | warning ]
```

**Hierarchy Level**

```
[edit services unified-access-control]
```

**Release Information**

Statement introduced in Junos OS Release 12.1.

**Description**

This option determines whether server certificate verification is required when initiating a connection between an SRX Series device and a Junos Pulse Access Control Service in a UAC configuration. If no CA profile contains the certificate authority (CA) that signed the configured server certificate for the Access Control Service, this option determines whether the commit check should fail, a warning should be displayed, or the connection should be made without any warning.

> **NOTE:** For strict security, this option should be reset to **required**, and the proper CA certificate should be specified in the CA profile.

**Options**

- **optional**—Certificate verification is not required. If the CA certificate is not specified in the **ca-profile** option, the commit check passes and no warning is issued.
- **required**—Certificate verification is required. If the CA certificate is not specified in the **ca-profile** option, an error message is displayed, and the commit check fails. Use this option to ensure strict security.

**Default:** **warning**—Certificate verification is not required. A warning message is displayed during commit check if the CA certificate is not specified in the **ca-profile** option.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

# client (System Services)

**Syntax**

```
client ip-address;
```

**Hierarchy Level**

```
[edit system services webapi]
```

**Release Information**
Statement introduced in Junos OS release 12.3X48-D30.

**Description**
Configures the IP address of the client. For the Integrated ClearPass Authentication and Enforcement feature Web API daemon configuration, the client is the ClearPass Policy Manager (CPPM).

The SRX Series Web API daemon acts as an HTTP(S) server. The CPPM client sends POST request messages containing user authentication and identity information to the Web API daemon. The SRX Series device accepts information only from the configured address of the client.

**Required Privilege Level**
**system**—To view this statement in the configuration.

**system-control**—To add this statement to the configuration.

# client-id (Services User Identification)

**Syntax**

```
client-id client-id;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query]
```

**Release Information**
Statement introduced in Junos OS release 12.3X48-D30.

**Description**
Configures the client ID that the SRX Series device requires to obtain an access token for the Integrated ClearPass Authentication and Enforcement user query function. The client ID must be consistent with the API client configured on the CPPM.

The ClearPass endpoint API requires use of OAuth (RFC 6749) to authenticate and authorize the SRX Series device access. The SRX Series device uses the Client Credentials grant type access token, which is one of the two types that ClearPass supports.

If it is configured, the user query function allows the SRX Series device to query the CPPM for authentication and identity information about individual users when it does not receive this information from the CPPM through the SRX Series Web API daemon (webapi).

**Required Privilege Level**
**services**—To view this statement in the configuration.

**services-control**—To add this statement to the configuration.

# client-id (Identity Management Advanced Query Primary)

**Syntax**

```
client-id client-id;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection primary]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
For the advanced user query function, configure the client ID that the SRX Series provides to the Juniper Identity Management Service (JIMS) primary server as part of its authentication to it. The SRX Series device must authenticate to the server to obtain an access token that allows the SRX Series device to query the server for user identity information. The client ID must be consistent with the API client configured on the Juniper Identity Management Service primary server.

Juniper Identity Management Service requires the SRX Series device to use OAuth2 to authenticate to it before the SRX Series device is allowed to query it for user identity information. OAuth2 requires that the client, in this case the SRX Series device, provide credentials. If the client is authenticated, it is granted an access token.(See RFC 6749)

To obtain an access token, the SRX Series device must specify the client ID and the client secret. Both the client ID and the client secret must be consistent with the API client configured on the Juniper Identity Management Service that is used as the primary server.

You configure this information for a secondary server separately. The SRX Series device always attempts to connect to the primary server first.

The advanced user identity query feature relies on the Juniper Identity Management Service that provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

It provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as device identity, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- connection | 460
- connect-method | 464
- filter | 487
- port | 569
- primary | 575
- invalid-authentication-entry-timeout | 526

# client-id (Identity Management Advanced Query Secondary)

**Syntax**

```
client-id client-id;
```

**Hierarchy Level**

```
[edit services user identification identity-management connection secondary]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure for the secondary Juniper Identity Management Service (JIMS) server connection the client ID that the SRX Series requires to obtain an access token so that it can issue successfully the advanced user query function.

This process is part of the client credentials grant type access token. For the client credentials grant type access token, Juniper Identity Management Service requires use of OAuth2 to authenticate and authorize access by the SRX Series device. (See RFC 6749.) To authenticate itself to the Juniper Identity Management Service, the SRX Series device must acquire an access token. It must authenticate itself before it can query the server for user identity information. To obtain an access token, the SRX Series device must specify the client ID and the client secret. Both the client ID and the client secret must be consistent with the API client configured on the Juniper Identity Management Service secondary server.

**NOTE:** The client ID is part of a set of credentials. You configure a set of credentials for both the primary server and the secondary server. The SRX Series device always attempts to connect to the primary server first.

The advanced user identity query feature relies on the Juniper Identity Management Service that provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

It provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as device identity, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable any other actively used options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication and the ClearPass query and device-id functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- address | 405
- authentication-entry-timeout | 420
- batch-query | 426
- ca-certificate | 435
- connect-method | 464
- ip-query | 529
- invalid-authentication-entry-timeout | 526

# client-group

### Syntax

```
client-group [ group-names ];
```

### Hierarchy Level

```
[edit access profile profile-name client client-name]
[edit access profile profile-name session-options]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify a list of client groups that the client belongs to. If the group list is not defined as part of the client profile, the client group configured at the **[edit access  profile session-options]** hierarchy level is used.

### Options

***group-names*** —Names of one or more groups the client belongs to, separated by spaces—for example **g1, g2, g3**. The total length of the group name string cannot exceed 256 characters.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

# client-idle-timeout (Access Profile)

**Syntax**

```
client-idle-timeout minutes;
```

**Hierarchy Level**

```
[edit access profile profile-name session-options]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify the grace period that begins after an authenticated user terminates all sessions and connections. Authentication is not required if a new connection is initiated during the grace period by the same user.

**Options**

***minutes*** —Number of minutes of idle time that elapse before the session is terminated.

**Range:** 10 through 255 minutes

**Default:** 10 minutes

**Required Privilege Level**

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

# client-name-filter

### Syntax

```
client-name-filter client-name {
    count number;
    domain-name domain-name;
    separator special-character;
}
```

### Hierarchy Level

```
[edit access profile profile-name]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Define client-name-related restrictions. Clients whose names follow these restrictions are authenticated on the server.

### Options

*client-name*—Name of the client.

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# client-secret (Services User Identification)

**Syntax**

```
client-secret client-secret;
```

**Hierarchy Level**

```
[edit ervices user-identification authentication-source aruba-clearpass user-query]
```

**Release Information**
Statement introduced in Junos OS release 12.3X48-D30.

**Description**
Configures the client secret used with the client ID that the SRX Series device requires to obtain an access token for the Integrated ClearPass Authentication and Enforcement user query function. The client secret must be consistent with the client secret configured on the CPPM.

The ClearPass endpoint API requires use of OAuth (RFC 6749) to authenticate and authorize SRX Series device access. The SRX Series device uses the Client Credentials grant type access token, which is one of the two types that ClearPass supports.

If it is configured, the user query function allows the SRX Series device to query the CPPM for authentication and identity information about individual users when it does not receive this information from the CPPM through the SRX Series Web API daemon (webapi).

**Required Privilege Level**
**services**—To view this statement in the configuration.

**services-control**—To add this statement to the configuration.

# client-secret (Identity Management Advanced Query Primary)

**Syntax**

```
client-secret client-secret;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection primary]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
Configure for the advanced user identity query feature the client secret that the SRX Series provides to the Juniper Identity Management Service primary server as part of its authentication to it. The Juniper Identity Management Service uses OAuth2 to authenticate the SRX Series device and grant it an access token. (See RFC 6749). The SRX Series device must authenticate to the server to obtain a token that allows it to query the server for user identity information The client secret must be consistent with the API client configured on the Juniper Identity Management Service (JIMS) primary server.

The Juniper Identity Management Service uses OAuth2 to authenticate the SRX Series device and grant it an access token. (See RFC 6749). The SRX Series device must authenticate to the server before it can query the server for user identity information.

If both a primary server and a secondary server are configured, the SRX Series device always attempts to connect to the primary server first.

The advanced user identity query feature relies on the Juniper Identity Management Service (JIMS) that provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

It provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as device identity, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- filter | 487
- port | 569
- primary | 575
- query-api | 585
- secondary | 603
- token-api | 639
- client-id | 446
- invalid-authentication-entry-timeout | 526

# client-secret (Identity Management Advanced Query Secondary)

**Syntax**

```
client-secret client-secret;
```

**Hierarchy Level**

```
[edit services user-identification (Services) identity-management connection secondary]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
Configure for the secondary Juniper Identity Management Service (JIMS) server connection the client secret that the SRX Series requires to obtain an access token so that it can issue successfully the advanced user query function.

This process is part of the client credentials grant type access token. For the client credentials grant type access token, Juniper Identity Management Service requires use of OAuth2 to authenticate and authorize access by the SRX Series device. (See RFC 6749.) To authenticate itself to the Juniper Identity Management Service, the SRX Series device must acquire an access token. It must authenticate itself before it can query the server for user identity information. To obtain an access token, the SRX Series device must specify the client secret, in addition to the client ID for the secondary server. Both the client secret and the client ID must be consistent with the API client configured on the Juniper Identity Management Service secondary server.

> **NOTE:** The client secret is part of a set of credentials. You configure a set of credentials for both the primary server and the secondary server. The SRX Series device always attempts to connect to the primary server first.

The advanced user identity query feature relies on the Juniper Identity Management Service that provides a global, end-to-end user identity management solution that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

It provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as device identity, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable any other actively used options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication and the ClearPass query and device-id functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- filter | 487
- port | 569
- primary | 575
- query-api | 585
- secondary | 603
- token-api | 639
- client-id | 446
- invalid-authentication-entry-timeout | 526

# client-session-timeout (Access Profile)

### Syntax

```
client-session-timeout minutes;
```

### Hierarchy Level

```
[edit access profile profile-name session-options]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify the amount of time after which user sessions are terminated, regardless of user activity (also known as a forced or hard authentication timeout).

### Options

***minutes*** —Number of minutes after which user sessions are terminated.

**Range:** 1 through 10,000 minutes

**Default:** Off

### Required Privilege Level

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# configuration-file

**Syntax**

```
server-name configuration-file filepath;
```

**Hierarchy Level**

```
[edit access securid-server]
```

**Release Information**

Statement introduced in Release 9.1 of Junos OS.

**Description**

Specify the path of the SecurID server configuration file. The file is copied on the devices in some directory location—for example, **/var/db/securid/sdconf.rec**.

**Options**

- ***server-name***—Name of the SecurID authentication server.
- ***filepath***—Path of the SecurID server configuration file.

**Required Privilege Level**

secret—To view this statement in the configuration.
secret-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# connection (Identity Management Advanced Query)

**Syntax**

```
connection {
    connect-method (http | https);
    port port;
    primary {
        address address;
        ca-certificate ca-certificate;
        client-id client-id;
        client-secret client-secret;
    }
    query-api query-api;
    secondary {
        address address;
        ca-certificate ca-certificate;
        client-id client-id;
        client-secret client-secret;
    }
    token-api token-api;
}
```

**Hierarchy Level**

```
[edit services user-identification identity-management]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure parameters for connecting the SRX Series to the Juniper Identity Management Service server to obtain user identity and device information.

For the SRX Series device to obtain user identity information, you must first establish a connection to the Juniper Identity Management Service server. The parameters to specify for the connection include the protocol, the IP address of the Juniper Identity Management Service server, and the information to authenticate the SRX Series device to the Juniper Identity Management Service server.

If you are using more than one Juniper Identity Management Service server, you must configure each server separately. The SRX Series device always attempts to connect to the primary server first. If the primary server fails, the SRX Series device falls back to the secondary server. The SRX Series device periodically probes the failed primary server and reverts to it when it is available.

**NOTE:** Only configuration of the primary server is mandatory. You are not required to use a secondary server.

The SRX Series advanced user identity query feature queries the Juniper Identity Management Service for user identity information that the SRX Series stores in its authentication table and uses to authenticate users. Use of the Juniper Identity Management Service allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

**WARNING:** Before you use this feature, you must disable any other actively used options under the [edit services user-identification] hierarchy. You cannot commit this configuration if active directory authentication and the ClearPass query and webapi functions are configured and committed.

To obtain device information, such as device identity, groups, and the operating system, from the Juniper Identity Management Service server using either the batch-query or ip-query configuration, you must set the device authentication source, as follows.

```
user@host# set services user-identification device-information authentication-source network-access-controller
```

**Options**

**connect-method**—The protocol to be used for the SRX Series device connection to Juniper Identity Management Service.

**Values:**

- **https**—HTTPS connection
- **http**—HTTP connection

  HTTP is used for debugging purposes only.

**Default:** HTTPS

**port**—The port on the Juniper Identity Management Service server that the SRX Series device uses to connect to it.

**Default: 443**

**Range: 1-65535**

**query-api**—The prefix of the URL path for querying user identities. This value is used to construct the prefix of the path for a batch query, an IP address query, and a user-query, each of which has a unique suffix:

- For IP query, *query-api*/ip/
- For batch query, *query-api*/users/
- For user-query *query-api*/user

**NOTE:** The default value for query-api is query-query/v2.

For example, for a batch query, assume that the query API is configured as **user-query/v2**. To generate the complete URL, the prefix is combined with the connection method, which is **HTTPS**, the IP address of the Juniper Identity Management Service server, expressed as a variable in this example (*JIMS*), the beginning timestamp, **begintime={timestamp}**, and the number of user identity information items to be provided in the record that the Juniper Identity Management Service server returns, **entry_count={count}**.

**'https://*JIMS*/user_query/v2/users/endpoints?begintime={timestamp}&entry_count={count}'**

**token-api**—The path of the URL that the SRX Series device uses to acquire an access token.

The remaining statements are described separately.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275

address | 405

authentication-entry-timeout | 420

batch-query | 426

ca-certificate | 435

client-id | 446

client-secret | 454

client-secret | 456

filter | 487

ip-query | 529

primary | 575

secondary | 603

token-api | 639

# connect-method (Identity Management Advanced Query)

**Syntax**

```
connect-method (http | https);
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the protocol to be used for the SRX Series device connection to Juniper Identity Management Service (JIMS). The SRX Series device connects to the Juniper Identity Management Service to obtain user identity information.

For the SRX Series device to do so, it must first establish a connection to the server. The connect-method parameter is part of a group of parameters that specify the information required for the connection. Other parameters include information about the server, such as its port number and IP address, and security information required by the Juniper Identity Management Service server.

The SRX Series device supports connection to a primary and a secondary Juniper Identity Management Service server. It always attempts to connect to the primary server first. It falls back to the secondary server when its queries to the primary server fail. If the primary server fails, the SRX Series device should be configured to periodically probe the failed primary server and revert to it when it becomes available.

The advanced user query feature allows the SRX Series device to query the Juniper Identity Management Service for identity information.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the [edit services user-identification] hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Options**

**http**—HTTP connection. Use the HTTP protocol for debugging purposes only.

**https**—HTTPS connection. Use the secure HTTPS protocol for requesting user identity information.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- filter | 487
- port | 569
- primary | 575
- query-api | 585
- secondary | 603
- token-api | 639
- invalid-authentication-entry-timeout | 526

# connect-method (Services User Identification)

**Syntax**

```
connect-method (http | https);
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query web-server]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure the application protocol used for the SRX Series device connection to the ClearPass Policy Manager (CPPM) for user query requests. You identify the connection protocol as part of the configuration that identifies the CPPM server. The user query function allows the SRX Series device to request from the CPPM user authentication and identity information for an individual user.

**Options**

**HTTP**—Protocol that the CPPM uses to connect to the SRX Series device.

**HTTPS**—Secure version of the protocol that the CPPM uses to connect to the SRX Series device.

**Default:** HTTPS—The connect-method configuration is optional. If it is not configured, HTTPS is assumed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# count

**Syntax**

```
count number;
```

**Hierarchy Level**

```
[edit access profile profile-name client-name-filter client-name]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Specify the number of characters to be stripped from a client name, from right to left, until the specified number of characters are deleted. The resulting name is sent to the authentication server.

**Options**
***number***—Number of characters to be stripped in a client name.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# custom-ciphers

### Syntax

```
custom-ciphers [ecdhe-rsa-with-3des-ede-cbc-sha | ecdhe-rsa-with-aes-128-cbc-sha |
    ecdhe-rsa-with-aes-128-cbc-sha256 | ecdhe-rsa-with-aes-128-gcm-sha256 | ecdhe-rsa-with-aes-256-cbc-sha |
    ecdhe-rsa-with-aes-256-cbc-sha384 | ecdhe-rsa-with-aes-256-gcm-sha384 | rsa-with-aes-128-cbc-sha256 RSA
    | rsa-with-aes-128-gcm-sha256 RSA | rsa-with-aes-256-cbc-sha256 RSA | rsa-with-aes-256-gcm-sha384 RSA |
    rsa-with-rc4-128-md5 RSA | 128bit rc4 | md5 hash rsa-with-rc4-128-sha RSA | 128bit rc4 |sha hash
    rsa-with-des-cbc-sha RSA | des cbc | sha hash rsa-with-3des-ede-cbc-sha RSA | 3des ede/cbc | sha hash
    rsa-with-aes-128-cbc-sha RSA | 128 bit aes/cbc | sha hash rsa-with-aes-256-cbc-sha RSA | 256 bit aes/cbc |sha
    hash rsa-export-with-rc4-40-md5 RSA-export | 40 bit rc4 | md5 hash rsa-export-with-des40-cbc-sha RSA-export
    | 40 bit des/cbc | sha hash rsa-with-null-md5 RSA | no symmetric cipher | md5 hash rsa-with-null-sha RSA | no
    symmetric cipher | sha hash | ecdhe-ecdsa-with-aes-256-gcm-sha384 | ecdhe-ecdsa-with-aes-256-cbc-sha384
    | ecdhe-ecdsa-with-aes-256-cbc-sha | ecdhe-ecdsa-with-aes-128-gcm-sha256
    |ecdhe-ecdsa-with-aes-128-cbc-sha256 |ecdhe-ecdsa-with-aes-128-cbc-sha
    |ecdhe-ecdsa-with-3des-ede-cbc-sha);];
```

### Hierarchy Level

```
[edit services ssl proxy profile profile-name]
[edit services ssl termination profile profile-name]
[edit services ssl initiation profile profile-name]
```

### Release Information

Statement introduced in Junos OS Release 12.1X44-D10.
This statement is supported in the SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances. Options to support Elliptic Curve Digital Signature Algorithm (ECDSA) added in Junos OS Release 18.3R1.

### Description

Configure custom cipher, which SSH server can use to perform encryption and decryption functions.

Custom ciphers allow you to define your own cipher list. If you do not want to use one of the three categories, you can select ciphers from each of the categories to form a custom cipher set.

To configure custom ciphers, you must set preferred-ciphers to custom. See **preferred-ciphers** for more details.

### Options

**ecdhe-rsa-with-3des-ede-cbc-sha**—ECDHE/RSA, 3 DES EDE/CBC, SHA hash

**ecdhe-rsa-with-aes-128-cbc-sha**—ECDHE/RSA, 128-bit AES/CBC, SHA hash

**ecdhe-rsa-with-aes-128-cbc-sha256**—ECDHE/RSA, 128-bit AES/CBC, SHA256 hash

**ecdhe-rsa-with-aes-128-gcm-sha256**—ECDHE/RSA, 128-bit AES/GCM, SHA256 hash

**ecdhe-rsa-with-aes-256-cbc-sha**—ECDHE/RSA, 256-bit AES/CBC, SHA hash

**ecdhe-rsa-with-aes-256-cbc-sha384**—ECDHE/RSA, 256-bit AES/CBC, SHA384 hash

**ecdhe-rsa-with-aes-256-gcm-sha384**—ECDHE/RSA, 256-bit AES/GCM, SHA384 hash

**rsa-export-with-des40-cbc-sha**—RSA-export, 40-bit DES/CBC, SHA hash

**rsa-export-with-rc4-40-md5**—RSA-export, 40-bit RC4, MD5 hash

**rsa-export1024-with-des-cbc-sha**—RSA 1024-bit export, DES/CBC, SHA hash

**rsa-export1024-with-rc4-56-md5**—RSA 1024-bit export, 56 bit RC4, MD5 hash

**rsa-export1024-with-rc4-56-sha**—RSA 1024-bit export, 56 bit RC4, SHA hash

**rsa-with-3des-ede-cbc-sha**—RSA, 3DES EDE/CBC, SHA hash

**rsa-with-aes-128-cbc-sha**—RSA, 128-bit AES/CBC, SHA hash

**rsa-with-aes-128-cbc-sha256**—RSA, 128-bit AES/CBC, SHA256 hash

**rsa-with-aes-128-gcm-sha256**—RSA, 128-bit AES/GCM, SHA256 hash

**rsa-with-aes-256-cbc-sha**—RSA, 256-bit AES/CBC, SHA hash

**rsa-with-aes-256-cbc-sha256**—RSA, 256-bit AES/CBC, SHA256 hash

**rsa-with-aes-256-gcm-sha384**—RSA, 256-bit AES/GCM, SHA384 hash

**rsa-with-des-cbc-sha**—RSA, DES CBC, SHA hash

**rsa-with-null-md5**—RSA, no symmetric cipher, MD5 hash

**rsa-with-null-sha**—RSA, no symmetric cipher, SHA hash

**rsa-with-rc4-128-md5**—RSA, 128-bit RC4, MD5 hash

**rsa-with-rc4-128-sha**—RSA, 128-bit RC4, SHA hash

**ecdhe-ecdsa-with-aes-256-gcm-sha384**—ECDHE,ECDSA, 256 bit aes/gcm, sha384 hash

**ecdhe-ecdsa-with-aes-256-cbc-sha384**—ECDHE,ECDSA, 256 bit aes/cbc, sha384 hash

**ecdhe-ecdsa-with-aes-256-cbc-sha**—ECDHE,ECDSA, 256 bit aes/cbc, sha hash

**ecdhe-ecdsa-with-aes-128-gcm-sha256**—ECDHE,ECDSA, 128 bit aes/gcm, sha256 hash

**ecdhe-ecdsa-with-aes-128-cbc-sha256**—ECDHE,ECDSA, 128 bit aes/cbc, sha256 hash

**ecdhe-ecdsa-with-aes-128-cbc-sha**—ECDHE,ECDSA, 128 bit aes/cbc, sha hash

**ecdhe-ecdsa-with-3des-ede-cbc-sha**—ECDHE,ECDSA, 3des ede/cbc, sha hash

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- SSL Proxy Overview | **96**
- Configuring SSL Forward Proxy | **99**
- Enabling Debugging and Tracing for SSL Proxy | **110**

# debug-level (System Services)

**Syntax**

```
debug-level level;
```

**Hierarchy Level**

```
[edit system services webapi]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Specify the trace level for the integrated ClearPass authentication and enforcement Web API daemon (webapi).

**Options**
***level***—A flag that specifies the type of logs to be written to the log file for the Web API daemon (webapi).

- **alert**—Matches alert messages.
- **crit**—Matches critical messages.
- **emerg**—Matches emergency messages.
- **error**—Matches error messages.
- **notice**—Matches notification messages.
- **warn**—Matches warning messages.

**Required Privilege Level**
system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# debug-log (System Services)

**Syntax**

```
debug-log filename;
```

**Hierarchy Level**

```
[edit system services webapi]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Specify the name of the log file to which trace messages for the integrated ClearPass authentication and enforcement Web API daemon (webapi) are written.

The debug level flag determines the kind of logs that are written to this file. Possible values are:

**alert**—Matches alert messages.

**crit**—Matches critical messages.

**emerg**—Matches emergency messages.

**error**—Matches error messages.

**notice**—Matches notification messages.

**warn**—Matches warning messages.

**Required Privilege Level**
system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# default-certificate (System Services)

**Syntax**

```
default-certificate;
```

**Hierarchy Level**

```
[edit system services webapi https]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Specify that the default certificate is to be used for the integrated ClearPass authentication and enforcement Web API daemon (webapi) HTTPS configuration. To ensure security, the Junos OS default certificate key size is 2084 bits.

**Required Privilege Level**

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# default-profile

**Syntax**

```
default-profile profile-name;
```

**Hierarchy Level**

```
[edit access firewall-authentication pass-through]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Specify the authentication profile to use if no profile is specified in a policy.

**Options**
***profile-name***—Name of the profile.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# delay-query-time (Services User Identification)

**Syntax**

```
delay-query-time delay-time-in-seconds;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
If the CPPM does not send to the SRX Series device authentication and identity information for a particular user, the SRX Series device can request that information for the user if you configure the user query function.

Delays can occur from when the CPPM initially posts user authentication information to the SRX Series device to when the SRX Series device updates its ClearPass authentication table with that information. In its transit, the user identity information must first pass through the CPPM device's control plane and the control plane of the SRX Series device.

During that period, traffic might arrive at the SRX Series device that is generated by an access request from a user whose authentication and identity information is in transit from the CPPM to the SRX Series device. Rather than allow the SRX Series device to respond automatically by sending a user query request *immediately*, you can set the delay time parameter specifying in seconds how long the SRX Series device should wait before sending the request.

After the delay timeout expires, the SRX Series device sends the query to the CPPM and creates a pending entry for the user in the Routing Engine authentication table. During this period, any arriving traffic matches the default policy whose action on the traffic you can configure.

**Options**

***delay-time-in-seconds***—Amount of time for the SRX Series device to delay before sending queries to the Aruba ClearPass Policy Manager (CPPM) for authentication and identity information for individual users

**Range:** 0 through 60 seconds

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# distinguished-name (Access)

**Syntax**

```
distinguished-name distinguished-name;
```

**Hierarchy Level**

```
[edit access ldap-options search admin-search],
[edit access profile profile-name ldap-options search admin-search]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify the distinguished name of an administrative user. The distinguished name is used in the bind for performing the LDAP search.

**Options**

***distinguished-name***—Set of properties that define the user. For example, **cn**=admin, **ou**=eng, **o**=juniper, **dc**=net.

**Required Privilege Level**

secret—To view this statement in the configuration.
secret-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# domain-name (Access Profile)

**Syntax**

```
domain-name domain-name;
```

**Hierarchy Level**

```
[edit access profile profile-name client-name-filter client-name]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Specify a domain name that must be in a client's name during the authentication process.

**Options**
***domain-name***—Domain name that must be in a client name. The name must not exceed 128 characters.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# enable-flow-tracing (Services)

**Syntax**

```
enable-flow-tracing;
```

**Hierarchy Level**

```
[edit services ssl proxy profile profile-name]
[edit services ssl termination profile profile-name]
[edit services ssl initiation profile profile-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10. This statement is supported on the SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices.

**Description**

Enable flow tracing for the profile.

When you configure **enable-flow-tracing** for SSL profiles, the debug tracing will be enabled on that profile when the flag is set as **selected-profile**.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION


- SSL Proxy Overview | **96**
- Configuring SSL Forward Proxy | **99**
- Enabling Debugging and Tracing for SSL Proxy | **110**

# enable-session-cache

**Syntax**

```
enable-session-cache;
```

**Hierarchy Level**

```
[edit services ssl termination profile profile-name]
[edit services ssl initiation profile profile-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10. This statement is supported on the SRX550M, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices.

**Description**

Enable SSL session cache.

You can enable session caching to cache session information, such as the pre-master secret key and agreed-upon ciphers, for both the client and server.

The cached information is identified by a session ID. In subsequent connections both parties agree to use the session ID to retrieve the information rather than create a new pre-master secret key. Session resumption shortens the handshake process and accelerates SSL transactions there by improves the throughput and maintains an appropriate level of security at the same time.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION


- SSL Proxy Overview | 96
- Configuring SSL Forward Proxy | 99
- Enabling Debugging and Tracing for SSL Proxy | 110

# end-user-profile

**Syntax**

```
end-user-profile profile-name profile-name
domain-name domain-name;
{
attribute device-category {
   string string-value;
}
attribute device-identity {
   string string-value;
}
attribute device-vendor {
   string string-value;
}
attribute device-type {
   string string-value;
}
attribute device-os {
   string string-value;
}
attribute device-os-version {
   string string-value;
   }
}
```

**Hierarchy Level**

```
[edit services user-identification device-information]
```

**Release Information**

Statement introduced in Junos OS 15.1X49-D70.

**Description**

Specify the name of the device identity profile, also referred to as the **end-user-profile**, and either one or more of its attributes or the name of the Active Directory domain to which the device belongs.

The device identity profile is a key component of the SRX Series device identity feature, which enables you to control access to network resources based on the identity of the user's device, not the identity of the user of the device. The device identity profile includes the domain name and a collection of attributes that characterize the device.

**NOTE:** You cannot configure the device identity profile without specifying either the domain that the device belongs to at least one of its attributes.

**Options**

- profile-name *profile-name*—Name of the device identity profile; for example, marketing-west-coast. The profile is specified in the **source-end-user-profile** field of a security policy.
- domain *domain-name*—Name of the domain to which the device belongs; for example, domain1.
- attribute device-identity *string*--Name given to the device, for example, my-device1.
- attribute device-category *string*--Category of the device, for example, laptop.
- attribute device-vendor *string*—Name of the manufacturer of the device, for example, Lenovo.
- attribute device-type *string*--Type of device; for example, ThinkPad.
- attribute device-os *string*-—Operating system running on the device; for example, Windows.
- attribute device-os-version *string*-—Version of the operating system that is running on the device; for example, 10.1.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Understanding Access Control to Network Resources Based on Device Identity Information | 244

Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature | 247

Understanding the Device Identity Authentication Table and Its Entries | 252

# fail

### Syntax

```
fail string;
```

### Hierarchy Level

```
[edit access firewall-authentication pass-through default-profile profile-name (ftp | http | telnet) banner]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify the banner that a client sees if the authentication process fails.

### Options

***string***—Banner text. Maximum length of the message text is 250 characters. Enclose the banner text within spaces or special characters—for example, quotation marks (" ").

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# file (Services User Identification)

**Syntax**

```
file {
    filename
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
}
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass traceoptions]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure the name of the trace log file and its characteristics to which messages for the behavior of the authentication source are logged. For the SRX Series device integrated ClearPass authentication and enforcement feature, the authentication source is the Aruba ClearPass Policy Manager (CPPM).

**Options**

***filename***—Name of the log file.

**files *max-number-of-files***—Specifies the maximum number of trace files.

Range: 2 through 1000

**match *regular-expression***—Specifies a regular expression that determines which lines are logged.

**no-world-readable**—Denies users the ability to read the log file.

**size *max-file-size***—Specifies the trace file maximum file size.

Range: 10,240 through 1,073,741,824.

**world-readable**—Allows users to read the log file.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# file (System Logging)

### Syntax

```
file name {
    allow-duplicates;
    archive name password password routing-instance routing-instance <(binary-data | no-binary-data)> <files files>
        <size bytes> <start-time start-time> <transfer-interval minutes> <(world-readable | no-world-readable)>;
    contents (any | authorization | change-log | conflict-log | daemon | dfc | external | firewall | ftp |
        interactive-commands | kernel | local0 | lpr | mail | news | ntp | pfe | privileged | security | syslog | user | uucp) {
    }
    explicit-priority;
    match match;
    match-strings [ match-strings ... ];
    structured-data (brief | detail);
}
```

### Hierarchy Level

```
[edit system syslog]
```

### Release Information

Statement introduced before Junos OS Release 12.1X47 for SRX Series.

### Description

Specify the file in which to log data.

### Options

- *filename*—Specify the name of the file in which to log data.
- *allow-duplicates*—Do not suppress the repeated messages.
- *any*—Specify all facilities information.
  - *alert*—Specify the conditions that should be corrected immediately.
  - *critical*—Specify the critical conditions.
  - *emergency*—Specify the conditions that cause security functions to stop.
  - *error*—Specify the general error conditions.
  - *info*—Specify the information about normal security operations.
  - *none*—Do not specify any messages.
  - *notice*—Specify the conditions that should be handled specifically.

    - *warning*—Specify the general warning conditions.
- *archive*—Specify the archive file information.
    - *archive-sites*—Specify a list of destination URLs for the archived log files.
        - *url*—Specify the primary and failover URLs to receive archive files.
    - *binary-data*—Mark file such that it contains binary data.
    - *no-binary-data*—Do not mark the file such that it contains binary data.
    - *files*—Specify the number of files to be archived. Range: 1 through 1000 files.
    - *size*—Specify the size of files to be archived. Range: 65,536 through 1,073,741,824 bytes.
    - *world-readable*—Allow any user to read the log file.
    - *no-world-readable*—Do not allow any user to read the log file.
    - *start-time*—Specify the start time for file transmission. Enter the start time in the yyyy-mm-dd.hh:mm format.
    - *transfer-interval*—Specify the frequency at which to transfer the files to archive sites.
- *authorization*—Specify the authorization system.
- *change-log*—Specify the configuration change log.
- *conflict-log*—Specify the configuration conflict log.
- *daemon*—Specify the various system processes.
- *dfc*—Specify the dynamic flow capture.
- *explicit-priority*—Include the priority and facility in messages.
- *external*—Specify the local external applications.
- *firewall*—Specify the firewall filtering system.
- *ftp*—Specify the FTP process.
- *interactive-commands*—Specify the commands executed by the UI.
- *kernel*—Specify the kernel information.
- *match*—Specify the regular expression for lines to be logged.
- *ntp*—Specify the NTP process.
- *pfe*—Specify the Packet Forwarding Engine.
- *security*—Specify the security-related information.

- *structured-data*—Log the messages in structured log format.
  - *brief*—Omit English language text from the end of the logged message.
- *user*—Specify the user processes.
  - *info*—Specify the informational messages.

**Required Privilege Level**
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

# filter (Security)

**Syntax**

```
filter threat-attack
```

**Hierarchy Level**

```
[edit security log stream stream-name]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Configure the log stream filter to transmit only threat and attack logs to the ClearPass Policy Manager (CPPM). The integrated ClearPass authentication and enforcement feature sends to the CPPM threat and attack logs detected by the SRX Series device security modules. You can use these reports to inform your approach to hardening the CPPM security policy. Setting the log stream filter to threat-attack ensures that the SRX Series device and the log server are not overburdened by irrelevant logs.

**NOTE:** Unlike for other features that use a filter for log streams, threat-attack is the only log stream filter supported for integrated ClearPass. Therefore, it is not shown here as an option.

**Required Privilege Level**
security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

# filter (Identity Management Advanced Query)

**Syntax**

```
filter {
    domain name;
    exclude-ip {
        address-book book-name;
        address-set address-set;
    }
    include-ip {
        address-book book-name;
        address-set address-set;
    }
}
```

**Hierarchy Level**

```
[edit services user-identification identity-management]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

The advanced user identity query feature enables the SRX Series device to communicate with the Juniper Identity Management Service (JIMS) server to obtain user identity information for an individual user (ip-query) or a group of users (batch query).

Optionally, you can configure filters to convey to the JIMS server at a more granular level the users for whom you want information, based on their IP addresses. The filter statement gives you the flexibility to specify a range of IP addresses to be excluded from the record that the JIMS server sends in response or a range of IP addresses to be included in it. You can also constrain the query target to users in one or more specific active directory domains. Only IPv4 addresses are supported.

You can configure a filter that includes all three specifications: **include-ip**, **exclude-ip**, and **domain**.

> **NOTE:** Filters are contextual. That is, you can use a different filter configuration for different requests. If you change the filter configuration, the new filter applies to subsequent user identity requests exclusively. It has no bearing on prior query requests

Use of the JIMS allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Options**

**include-ip** —address-book *book-name* address-set *address-set-name*. Optionally, configure a filter that directs the SRX Series device to issue a query to the JIMS server to include in its response record user identity information for users based on IP addresses in certain address-ranges.

The following are the two behaviors when an **include-ip** is configured:

- Batch query—An SRX Series device sends a request to JIMS with the include list of IP addresses.
- IP query—If the IP address to be queried is included, then the SRX Series device queries JIMS only for those IP addresses that need to be included and does not query for other IP addresses; based on the IP query, JIMS does not trigger the PC probe for the IP addresses that are not included in the IP query.

A filter can include up to twenty IP address ranges. Therefore, an address set that contains more than twenty ranges will cause the filter configuration to fail. To specify the ranges, specify the name of a predefined address set which includes them and which is included in an existing address book.

**NOTE:** The filter for IP addresses does not support nested address sets in an address book. If an address book contains nested address sets, it is ignored.

Here is an include-ip address configuration:

```
user@host# set security address-book mybook address addr1 range-address 198.51.100.0 to 198.51.120.0
user@host# set security address-book mybook address-set myset address addr1
user@host# set service user-identification identity-management filter include-ip address-book mybook
    address-set myset
```

**exclude-ip**—address-book *book-name* address-set *address-set-name*. Optionally, configure a filter that directs the SRX Series device to issue a query to the JIMS server to exclude from its response record user identity information for users based on the specified address-ranges.

The following are the two behaviors when an **exclude-ip** is configured:

- Batch query—An SRX Series device sends a request to JIMS with the exclude list of IP addresses.
- IP query—If the IP address to be queried is excluded, then no request is sent from an SRX Series device to JIMS.

To specify the ranges, specify the name of a predefined address set which includes them and which is included in an existing address book. The address set must not include more than twenty IP addresses, otherwise the **exclude-ip** filter will fail. Here is an **exclude-ip** address configuration similar to that of the **include-ip** filter:

```
user@host# set security address-book mybook address addr1 range-address 198.51.100.0/24 to
    198.51.120.0/24
user@host# set security address-book mybook address-set myset address addr1
user@host# set service user-identification identity-management filter exclude-ip address-book mybook
    address-set myset
```

**NOTE:** Starting in Junos OS Release 18.3R1, you can include or exclude IPv6 addresses for filtering the IP addresses, in addition to IPv4 addresses.

**domain**—One or more active directory domains of interest to the SRX Series device. You can specify up to twenty domain names for the filter.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- address | 405
- authentication-entry-timeout | 420
- batch-query | 426
- ca-certificate | 433
- client-id | 446
- port | 569
- invalid-authentication-entry-timeout | 526
- primary | 575
- query-api | 585
- secondary | 603
- token-api | 639

# firewall-authentication

**Syntax**

```
firewall-authentication {
    pass-through {
        default-profile profile-name;
        ftp {
            banner {
                fail string;
                login string;
                success string;
            }
        }
        http {
            banner {
                fail string;
                login string;
                success string;
            }
        telnet {
            banner {
                fail string;
                login string;
                success string;
            }
    }
    traceoptions {
        file {
            filename;
            files number;
            flag flag;
            match regular-expression;
            no-remote-trace;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }
    }
    web-authentication {
        banner {
            success string;
        }
        default-profile profile-name;
    }
```

```
}
```

**Hierarchy Level**

```
[edit access]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.

**Description**

Configure default firewall authentication settings used by firewall authentication policies that restrict and permit access of firewall users to protected resources behind a firewall.

**Options**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Dynamic VPN Overview*

Firewall User Authentication Overview | 35

# firewall-authentication (Security)

**Syntax**

```
firewall-authentication {
    traceoptions (Security Firewall Authentication) {
        flag (all | authentication | proxy) {
        }
    }
}
```

**Hierarchy Level**

```
[edit security]
```

**Release Information**
Statement introduced in Junos OS Release 8.5.

**Description**
Defines the type of firewall authentication available for a logical system. Also specifies the data plane firewall authentication tracing options.

**Options**

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple flag statements.
  - **all**—Enable all tracing operations.
  - **authentication**—Trace data-plane firewall authentication events.
  - **proxy**—Trace data-plane firewall authentication proxy events.
- **detail**—Display moderate amount of data.
- **extensive**—Display extensive amount of data.
- **terse**—Display minimum amount of data.

**Required Privilege Level**
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

# firewall-authentication (Security Policies)

**Syntax**

```
firewall-authentication {
    pass-through {
        access-profile profile-name;
        client-match user-or-group-name;
        ssl-termination-profile profile-name;
        web-redirect;
        web-redirect-to-https;
        auth-only-browser
            auth-user-agent
    }
    push-to-identity-management
    user-firewall {
        access-profile profile-name;
        domain domain-name
        ssl-termination-profile profile-name;
        web-redirect;
        web-redirect-to-https;
        auth-only-browser
    }
    web-authentication {
        client-match user-or-group-name;
    }
}
```

**Hierarchy Level**

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* then permit]

**Release Information**

Statement introduced in Junos OS Release 8.5.

Support added for the **user-firewall** option in Junos OS Release 12.1X45-D10.

Support for the **ssl-termination-profile** and **web-redirect-to-https** options added on SRX5600 and SRX5800 Services Gateways starting from Junos OS Release 12.1X44-D10, on SRX5400 devices starting from 12.1X46-D10, and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, support for the **web-redirect** and **web-redirect-to-https** options under **user-firewall** added on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Services Gateways.

Starting with Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1, support for the **auth-only-browser** option was added under **pass-through** and **user-firewall** and the **auth-user-agent** option was added under **pass-through auth-only-browser** on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Services Gateways. Starting with Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1, support for the **auth-only-browser** option was added under **pass-through** and **user-firewall** and the **auth-user-agent** option was added under **pass-through auth-only-browser** on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Services Gateways. Starting with Junos OS Release 15.1X49-D100 and Junos OS Release 17.3R1, support was added for **push-to-identity-management**.

**Description**

Configure firewall authentication methods.

**Options**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

- *Understanding User Role Firewalls*

# firewall-authentication (User Identification)

**Syntax**

```
firewall-authentication priority priority;
```

**Hierarchy Level**

```
[edit security user-identification authentication-source]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X45-D10. Support for **disable** option dropped in Junos OS Release 12.1X47-D10.

**Description**

Enables the firewall authentication table as an authentication source. The priority of this table among other authentication tables establishes the search sequence used to identify user and role values.

**Options**

***priority***—A unique value between 0 and 65535 that determines the sequence for searching multiple tables to retrieve a user role. Each table is given a unique priority value. The lower the value, the higher the priority. A table with priority 120 is searched before a table with priority 200.

**Default:** 150

Setting the priority value of the firewall authentication table to 0 is equivalent to disabling the table and eliminating it from the search sequence.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

- *authentication-source (Security)*
- *Understanding User Role Firewalls*

# firewall-authentication-forced-timeout

**Syntax**

```
firewall-authentication-forced-timeout minutes;
```

**Hierarchy Level**

```
[edit services user-identification active-directory-access]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
Configure the firewall authentication forced timeout setting to apply to entries for user who authenticate through captive portal.

When a user authenticates through captive portal, an authentication table entry is generated for that user based on the information that the SRX Series device obtains from the firewall authentication module. At that point, the default traffic-based authentication timeout logic is applied to the entry. This statement gives you control over how long non-domain users who authenticate through captive portal remain authenticated.

When the firewall authentication forced timeout value is configured, it is used in conjunction with the traffic-based timeout logic.

Here is how timeout settings affect active directory authentication entries for users authenticated through captive portal.

- The firewall authentication forced timeout is set for 3 hours.

  Traffic continues to be received and generated by a device associated with an authentication entry for a user. After 3 hours the authentication entry expires, although at that time there are sessions anchored in Packet Forwarding Engine for the authentication entry.

- If set, the firewall authentication forced timeout has no effect.

  An authentication entry does not have sessions anchored to it. It expires after the time set for the authentication entry timeout, for example, 30 minutes.

- The firewall authentication forced timeout configuration is deleted.

  Firewall authentication forced timeout has no effect on new authentication entries. Firewall authentication forced timeout remains enforced for existing authentication entries to which it applied before it was deleted. That is, for those authentication entries, the original forced timeout setting remains in effect.

- The firewall authentication forced timeout configuration setting is changed.

The new timeout setting is applied to new incoming authentication entries. Existing entries keep the original, former setting.

- The firewall authentication forced timeout is set to 0, disabling it.

  If the firewall authentication forced timeout is set to a new value, that value is assigned to all incoming authentication entries. There is no firewall authentication forced timeout setting for existing authentication entries.

- The firewall authentication forced timeout value is not configured.
  - The SRX Series device generates an authentication entry for a user. The default traffic-based timeout logic is applied to the authentication entry.
  - The active directory timeout value is configured for 50 minutes. A traffic-based timeout of 50 minutes is applied to an authentication entry.
  - The active directory timeout is not configured. The default traffic-based timeout of 30 minutes is applied to an authentication entry.

**Options**

***minutes***—The maximum duration for which the non-domain users who authenticate through captive portal remain authenticated.

**Default:** 30 minutes

**Range:** 10 through 1440 minutes

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding the Forced Timeout Setting Assigned to Active Directory Authentication Entries for Users Authenticated Through Captive Portal | **226**

invalid-authentication-entry-timeout | **526**

Understanding the Invalid Authentication Table Entry Timeout Setting | **190**

# firewall-authentication-service

**Syntax**

```
firewall-authentication-service (enable | disable);
```

**Hierarchy Level**

```
[edit system processes]
```

**Release Information**
Statement introduced in Junos OS Release 8.5.

**Description**
Enable or disable the firewall authentication service process.

**Options**
- **enable**—Start the firewall authentication service process.
- **disable**—Stop the firewall authentication service process.

**Required Privilege Level**
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# firewall-user

### Syntax

```
firewall-user {
    password password;
}
```

### Hierarchy Level

```
[edit access profile profile-name client client-name]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify a client as a firewall user and the associated password (encrypted).

### Options

**password *password***—Password used by the firewall user during local authentication.

**Range:** 1 through 128 characters

### Required Privilege Level

secret—To view this statement in the configuration.
secret-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# flag (Services)

### Syntax

```
flag (all | cli-configuration | initiation | proxy | selected-profile | termination);
```

### Hierarchy Level

```
[edit services ssl traceoptions]
```

### Release Information

Statement introduced in Junos OS Release 12.1X44-D10. This statement is supported on the SRX1500, SRX5400, SRX5600, and SRX5800 devices and vSRX.

### Description

Specify the tracing flag parameters.

### Options

- *all*—Trace all the parameters.
- *cli-configuration*—Trace CLI configuration events.
- *initiation*—Trace initiation service events.
- *proxy*—Trace proxy service events.
- *selected-profile*—Trace events for profiles with **enable-flow-tracing** set.
- *termination*—Trace termination service events.

### Required Privilege Level

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

# from-zone (Security Policies)

**Syntax**

```
from-zone zone-name to-zone zone-name {
    policy policy-name {
        description description;
        match {
            application {
                [junos-defaults | application];
                any;
            }
            dynamic-application {
                [dynamic-application-name |dynamic-application-group-name];
                any;
                none;
            }
            destination-address {
                [address];
                any;
                any-ipv4;
                any-ipv6;
            }
            source-address {
                [address];
                any;
                any-ipv4;
                any-ipv6;
            }
            source-identity {
                [role-name];
                any;
                authenticated-user;
                unauthenticated-user;
                unknown-user;
            }
            source-end-user-profile {
                profile-name;
            }
        }
        scheduler-name scheduler-name;
        then {
            count {
                alarm {
```

```
        per-minute-threshold number;
        per-second-threshold number;
    }
}
deny;
log {
    session-close;
    session-init;
}
```

```
permit {
    application-services {
        application-firewall {
            rule-set rule-set-name;
        }
        application-traffic-control {
            rule-set rule-set-name;
        }
        gprs-gtp-profile profile-name;
        gprs-sctp-profile profile-name;
        idp;
        redirect-wx | reverse-redirect-wx;
        ssl-proxy {
            profile-name profile-name;
        }
        uac-policy {
            captive-portal captive-portal;
        }
        utm-policy policy-name;
    }
    destination-address {
        drop-translated;
        drop-untranslated;
    }
    firewall-authentication {
        pass-through {
            access-profile profile-name;
            client-match user-or-group-name;
            ssl-termination-profile profile-name;
            web-redirect;
            web-redirect-to-https;
        }
        user-firewall {
            access-profile profile-name;
            domain domain-name
            ssl-termination-profile profile-name;
        }
        web-authentication {
            client-match user-or-group-name;
        }
    }
    services-offload;
    tcp-options {
        initial-tcp-mss mss-value;
```

```
            reverse-tcp-mss mss-value;
            sequence-check-required;
            sequence-check-required;
            syn-check-required;
          }
          tunnel {
            ipsec-group-vpn group-vpn;
            ipsec-vpn vpn-name;
            pair-policy pair-policy;
          }
        }
        deny | reject;
        deny | reject [profile name];
      }
    }
}
```

**Hierarchy Level**

```
[edit security policies]
```

**Release Information**

Statement introduced in Junos OS Release 8.5. Support for the **services-offload** option added in Junos OS Release 11.4. Support for the **source-identity** option added in Junos OS Release 12.1. Support for the **description** option added in Junos OS Release 12.1. Support for the **ssl-termination-profile** and **web-redirect-to-https** options added in Junos OS Release 12.1X44-D10. Support for the **user-firewall** option added in Junos OS Release 12.1X45-D10. Support for the **initial-tcp-mss** and **reverse-tcp-mss** options added in Junos OS Release 12.3X48-D20. Support for the **dynamic-application** and **deny** options added in Junos OS Release 18.2R1.

**Description**

Specify a source zone and destination zone to be associated with the security policy.

**Options**

- **from-zone *zone-name***—Name of the source zone.
- **to-zone *zone-name***—Name of the destination zone.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Security Policies Overview*
- *Understanding Security Policy Rules*
- *Understanding Security Policy Elements*
- *Unified Policies Configuration Overview*

# ftp (Access)

**Syntax**

```
ftp {
    banner {
        fail string;
        login string;
        success string;
    }
}
```

**Hierarchy Level**

```
[edit access firewall-authentication pass-through]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.

**Description**

Configure banners for the FTP login prompt, successful authentication, and failed authentication.

**Options**

The remaining statements are explained separately.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

# group-profile (Access)

**Syntax**

```
group-profile profile-name {
    ppp {
        cell-overhead;
        encapsulated-overhead;
        framed-pool address-pool-name;
        idle-timeout seconds;
        interface-id interface-identifier;
        keepalive seconds;
        primary-dns IP address;
        primary-wins IP address;
        secondary-dns IP address;
        secondary-dns IP address;
    }
}
```

**Hierarchy Level**

```
[edit access]
```

**Release Information**

Statement introduced in Release 10.4 of Junos OS.

**Description**

Configure a group profile to define Point-to-Point Protocol (PPP) attributes. Any client referencing the configured group profile inherits all the group profile attributes.

**Options**

- **ppp**—Configure Point-to-Point Protocol (PPP) attributes.
- **cell-overhead**—Configure the session to use Asynchronous Transfer Mode (ATM)-aware egress shaping.
- **framed-pool *pool-name***—Configure a framed-pool.
- **idle-timeout**—Configure the idle timeout for a user.
- **interface-id**—Configure the interface identifier.
- **keep-alive**—Configure the keepalive interval for an L2TP tunnel.
- **primary-dns**—Specify the primary-dns IP address.
- **secondary-dns**—Specify the secondary-dns IP address.

- **primary-wins**—Specify the primary-wins IP address.
- **secondary-wins**—Specify the secondary-wins IP address.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# http (Access)

### Syntax

```
http {
    banner {
        fail string;
        login string;
        success string;
    }
}
```

### Hierarchy Level

```
[edit access firewall-authentication pass-through]
```

### Release Information

Statement introduced in Junos OS Release 8.5.

### Description

Configure banners for the HTTP login prompt, successful authentication, and failed authentication.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Obtaining Username and Role Information Through Firewall Authentication | 172

# http (Services)

**Syntax**

```
http {
    interfaces [ interface-names ];
    port port;
}
```

**Hierarchy Level**

```
[edit system services web-management]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.

**Description**

Unencrypted HTTP connection setting.

**Options**

**interfaces [ *interface-names*]**—Name of one or more interfaces on which to allow the HTTP service.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION


- Example: Configuring HTTPS Traffic to Trigger Pass-Through Authentication | 70
- Example: Configuring HTTPS Traffic to Trigger Web Authentication | 89
- Firewall User Authentication Overview | 35

# http (Services User Identification)

**Syntax**

```
http port port-number;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source name user-query web-server name connect-method]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure HTTP as the connection protocol to use for the SRX Series integrated ClearPass authentication and enforcement feature's connection to the ClearPass Policy Manager (CPPM) webserver for individual user authentication queries. You identify the connection protocol as part of the configuration that identifies the CPPM webserver (mutually exclusive with HTTPS).

If the SRX Series devices does not find an authentication entry for a user in its local ClearPass authentication table, it can query the Aruba ClearPass webserver for this information.

**NOTE:** This configuration assumes that aruba-clearpass is specified as the authentication source.

**Options**

***port-number***—Port numbr to use for the HTTP connection protocol.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# http (System Services)

### Syntax

```
http port  port-number;
```

### Hierarchy Level

```
[edit system services webapi]
```

### Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

### Description

Specify HTTP as the communication protocol for the Web API function of the SRX Series integrated ClearPass authentication and enforcement feature.

The SRX Series device exposes to the ClearPass Policy Manager (CPPM) the Web API for it to use to initiate a connection and then use that connection to send to the SRX Series device user authentication and identity information.

This statement also specifies the port number to use for the HTTP connection. The port number is optional.

> **NOTE:** If you deploy HTTP along with a Web management application, you must ensure that they run on different service ports.

### Options

***port-number***—Port for HTTP to use for the Web API function.

**Default:** 8080

### Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# https (Services)

### Syntax

```
https {
    interfaces [ interface-names ];
    local-certificate local-certificate-name;
    pki-local-certificate pki-local-certificate-name;
    port port;
    system-generated-certificate name;
}
```

### Hierarchy Level

```
[edit system services web-management]
```

### Release Information

Statement introduced on the SRX5400, SRX5600, and SRX5800 devices starting from Junos OS Release 12.1X44-D10 and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.

### Description

Encrypted HTTPS connections.

### Options

***interface-names***—Name of one or more interfaces on which to allow the HTTPS service.

***local-certificate-name*** —Name of the X.509 certificate for a Secure Sockets Layer (SSL) connection. An SSL connection is configured at the **[edit security certificates local]** hierarchy.

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Example: Configuring HTTPS Traffic to Trigger Pass-Through Authentication | 70
- Example: Configuring HTTPS Traffic to Trigger Web Authentication | 89
- Firewall User Authentication Overview | 35

# https (Services User Identification)

**Syntax**

```
https (
    certificate  local-certificate;
    certificate-key  local-certificate-key;
    default-certificate;
    pki-local-certificate certificate-name;
    port port-number;
)
```

**Hierarchy Level**

[edit services user-identification authentication-source *name* user-query web-server *name* connect-method]

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure HTTPS as the connection protocol used for the SRX Series connection to the ClearPass Policy Manager (CPPM) for user query requests. You identify the connection protocol as part of the configuration that identifies the CPPM webserver.

The integrated ClearPass authentication and enforcement user query function allows the SRX Series device to request from the CPPM user authentication and identity information for an individual when the SRX Series ClearPass authentication table does not contain that information.

**NOTE:** This configuration assumes that aruba-clearpass is specified as the authentication source.

**Options**

**https**—Use the encrypted HTTPS protocol. (Mutually exclusive with HTTP.)

**Default:** HTTPS

**default-certificate**—Use the default HTTPS certificate.

For security reasons, the HTTPS default-certificate key size 2048.

***filename***—Custom certificate file.

The Web API supports only the Privacy-Enhanced Mail (PEM) format for the custom certificate and certificate key configuration.

***local-certificate-key***—Web API daemon service certificate key. This parameter is required if a custom service certificate file is configured.

***pki-certificate***—Use the local X.509 PKI certificate.

***port-number***—HTTPS service port.

**Range:** 1 through 65535.

**Default:** 8443

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# https (System Services)

**Syntax**

```
https (
    certificate  local-certificate;
    certificate-key  local-certificate-key;
    default-certificate;
    pki-local-certificate certificate-name;
    port port-number;
)
```

**Hierarchy Level**

```
[edit system services webapi connect-method]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Specify HTTPS as the communication protocol for the Web API function of the SRX Series integrated ClearPass authentication and enforcement feature. When you configure HTTPS, you specify the service certificate and certificate key. You can also specify the port to be used.

The Web API daemon, acting as an HTTPS server, allows the ClearPass Policy Manager (CPPM), acting as the client, to send POST request messages to it. The CPPM, which is the authentication source for this feature, sends to the SRX Series device user authentication and identity information.

> **NOTE:** If you deploy HTTPS with a Web management application, ensure that they run on different service ports.

**Options**

**https**—Specifies use of the encrypted HTTPS protocol. (Mutually exclusive with HTTP.)

**default-certificate**—Configures the Web API daemon (webapi) to use the default HTTPS certificate.

**Default:** key size, 2048

***filename***—Configures the Web API daemon to use the specified, custom certificate file.

For certificate and certificate key configuration, the Web API function supports only the Privacy-Enhanced Mail (PEM) format.

***local-certificate-key***—Configures the Web API daemon service certificate key. This parameter is required if a custom service certificate file is configured.

***certificate-name***—Configures the Web API daemon to use the local X.509 PKI certificate.

***port-number***—Configures the HTTPS service port.

**Range:** For port number, 1 through 65,535.

**Default:** For port, 8443.

**Required Privilege Level**

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# identity-management

**Syntax**

```
identity-management {
    authentication-entry-timeout minutes;
    batch-query {
        items-per-batch items-per-batch;
        query-interval seconds;
    }
    connection {
        connect-method (http | https);
        port port;
        primary {
            address address;
            ca-certificate ca-certificate;
            client-id client-id;
            client-secret client-secret;
        }
        query-api query-api;
        secondary {
            address address;
            ca-certificate ca-certificate;
            client-id client-id;
            client-secret client-secret;
        }
        token-api token-api;
    }
    filter {
        domain name;
        exclude-ip {
            address-book book-name;
            address-set address-set;
        }
        include-ip {
            address-book book-name;
            address-set address-set;
        }
    }
    invalid-authentication-entry-timeout minutes;
    ip-query {
        no-ip-query;
        query-delay-time seconds;
    }
```

```
    traceoptions {
        file <filename> <files files> <match match> <size size> <(world-readable | no-world-readable)>;
        flag name;
        level (all | error | info | notice | verbose | warning);
        no-remote-trace;
    }
}
```

### Hierarchy Level

[edit services **user-identification (Services)**]

### Release Information

Statement introduced in Junos OS Release 15.1X49-D100.

### Description

Configure identity management to collect identity information. The SRX Series or NFX Series device relies on JIMS to obtain user identity information much in the same way that it does LDAP. By configuring the **identity-management**, the device can query JIMS for identity information, populate identity management authentication table with the information that is obtained from JIMS and use the populated identity management authentication table to authenticate a user or a device requesting access to a protected resource.

**Options**

**authentication-entry-timeout**—Authentication entry timeout number (0, 10-1440) (minutes)

**Default:** 60

**invalid-authentication-entry-timeout**—Invalid authentication entry timeout number (0, 10-1440) (minutes)

**Default:** 30

**file**—Configure the trace file options.

- *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. By default, the name of the file is the name of the process being traced.
- **files** *number*—Maximum number of trace files. When a trace file named ***trace-file*** its maximum size, it is renamed to ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

  If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and a filename.

  Range: 2 through 1000 files

  Default: 10 files
- **match *regular-expression***—Refine the output to include lines that contain the regular expression.
- **size *maximum-file-size***—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the **trace-file** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

  If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option and a filename.

  Syntax: ***x* K** to specify KB, ***x* m** to specify MB, or ***x* g** to specify GB

  Range: 10 KB through 1 GB

  Default: 128 KB
- **world-readable** | **no-world-readable**—By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

**flag**——Trace operation to perform.

**level**—-Level of debugging output.

- **all** —Match all levels.
- **error** —Match error conditions.

- **info** —Match informational messages.
- **notice** —Match conditions that should be handled specially.
- **verbose** —Match verbose messages.
- **warning** —Match warning messages.

**no-remote-trace**——Set remote tracing as disabled.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**
system

RELATED DOCUMENTATION

Configure Juniper Identity Management Service to Obtain User Identity Information | 275

# infranet-controller

### Syntax

```
infranet-contoller host-name {
    address ip-address;
    ca-profile [ca-profile];
    interface interface-name;
    password password;
    port port-number;
    server-certificate-subject subject;
}
```

### Hierarchy Level

```
[edit services unified-access-control ]
```

### Release Information

Statement introduced in Junos OS Release 9.4.

### Description

To configure an Infranet Controller, specify the hostname of the IC Series device with which the SRX Series device should communicate. Possible values for this statement range from 1 to 31 characters.

This statement is required when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

One or more IC Series devices can be configured as Infranet Controllers on the SRX Series device. There is no maximum number of IC Series devices that can be configured. However, only one IC Series device can be active at any time. The others are failover devices. A round-robin algorithm determines which of the configured IC Series devices is the active Infranet Controller. If the active Infranet Controller becomes inoperative, the algorithm is reapplied to the remaining IC Series devices that are configured to establish the new active Infranet Controller.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# interface (Services)

**Syntax**

```
interface interface-name;
```

**Hierarchy Level**

```
[edit services unified-access-control infranet-controller hostname]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify the SRX Series interface through which the IC Series device should connect.

This statement is required when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

port (Services) | 571
password (Services) | 550

# interval (Services)

**Syntax**

```
interval seconds;
```

**Hierarchy Level**

```
[edit services unified-access-control ]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify the value in seconds that the SRX Series device should expect to receive a heartbeat signal from the IC Series device (default 30). This configuration statement is used in conjunction with the **timeout** statement to test active communications with the IC Series device. The value of the **interval** statement must be smaller than the value of **timeout** statement.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

timeout (Services) | 632
timeout-action | 633

# invalid-authentication-entry-timeout (Services User Identification Active Directory and ClearPass)

**Syntax**

```
invalid-authentication-entry-timeout timeout-value-in-minutes;
```

**Hierarchy Level**

```
[edit services user-identification active-directory-access]
[edit services user-identification authentication-source aruba-clearpass]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
Configure an independent timeout value to be assigned to invalid user authentication entries in the SRX Series authentication table for either Windows active directory or Aruba ClearPass. The invalid authentication entry timeout setting is different from the general authentication entry timeout setting. It allows you to protect invalid user authentication entries in an authentication table from expiring before the user can be validated.

User authentication entries in an authentication table contain a time-out value after which the entry expires, or is no longer valid. An invalid authentication entry is created with a NULL and INVALID state for a user's IP address and stored in the access directory authentication table when there is no identity information for that user. Prior to implementation of this feature, the current time-out value that applies to all user entries was applied to the invalid entry also.

Separate authentication tables exist for the two authentication sources and you configure separate settings for them, as illustrated in the following examples.

Use the following command to configure the invalid authentication entry timeout for entries in the Windows active directory authentication table. In this example, invalid authentication entries in the active directory authentication table will expire 40 minutes after they were created.

```
user@host# set services user-identification active-directory-access invalid-authentication-entry-timeout 40
```

Use the following command to configure the invalid authentication entry timeout for entries in the SRX Series ClearPass authentication table. In this example, invalid authentication entries in the SRX Series ClearPass authentication table will expire 22 minutes after they were created.

```
user@host# set services user-identification authentication-source aruba-clearpass
  invalid-authentication-entry-timeout 22
```

The following rules govern how the invalid authentication entry timeout setting is used:

- When you initially configure the invalid authentication entry timeout value, it is applied to any invalid authentication entries that are created *after* it was configured.

  However, all existing invalid authentication entries retain the default timeout of 30 minutes.

- If you do not configure the invalid authentication entry timeout function, then the default timeout of 30 minutes is applied to all invalid authentication entries.

- If you configure the invalid authentication entry timeout value but later you delete it, the default timeout of 30 minutes is applied to any invalid authentication entries created *after* the deletion.

  However, any invalid authentication entries to which the invalid entry timeout value was applied *before* the deletion retain that setting.

- If you change the setting for the invalid authentication entry timeout value, the new value is applied to all invalid authentication entries that were created *after* the value was changed. However, all existing invalid authentication entries retain the former invalid authentication entry timeout setting, if it applied to them. Those to which the default value of 30 minutes applies retain that setting.

- When the state of an invalid authentication entry changes to Pending or Valid, the invalid authentication entry timeout setting is no longer applicable to it. Therefore, the timeout value assigned to that entry is changed to the value that is set for the general authentication entry timeout.

**Options**

***timeout-value-in-minutes***—Expiration time in minutes to be applied to invalid authentication entries in the SRX Series authentication table for either Windows active directory or Aruba ClearPass authentication sources.

**Range:** 0 through 1440 minutes.

**Default:** 30 minutes

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding the Invalid Authentication Table Entry Timeout Setting | 190

Understanding the Forced Timeout Setting Assigned to Active Directory Authentication Entries for Users Authenticated Through Captive Portal | 226

firewall-authentication-forced-timeout | 498

# ip-address (Access Profile)

**Syntax**

```
ip-address address
```

**Hierarchy Level**

```
[edit access profile name  client name  xauth]
```

**Release Information**

Statement introduced in Release 10.4 of Junos OS.

**Description**

Specify the IP address for the client.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# ip-query (Identity Management Advanced Query)

**Syntax**

```
ip-query {
    no-ip-query;
    query-delay-time seconds;
}
```

**Hierarchy Level**

```
[edit services user-identification identity-management]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the parameters to be used for the IP query function. When this feature is enabled, the SRX Series device queries the Juniper Identity Management Service (JIMS) server for user identity information based on the IP address of a user's device.

For example, if information for a user is missing from a flow, the SRX Series device can issue a query request specifying the IP address of the user's device. Also, If the SRX Series device does not have identity information for a specific user, it can engage captive portal to authenticate the user. After it authenticates the user, the SRX Series device can issue a query request to the Juniper Identity Management Service, specifying the user ID and the IP address of the user's device to obtain additional information, such as the names of the groups that the user belongs to.

If there are many IP query requests in the queue, the SRX Series device can maintain multiple concurrent HTTP/HTTPS connections with the Juniper Identity Management Service to increase throughput. However, the number of concurrent connections are kept at a reasonable level, which is twenty or less, so as not to impose pressure on the Juniper Identity Management Service.

> **NOTE:** IP query is one of three query methods: IP query, batch query, and user query. All three types of queries can occur concurrently. They are not mutually exclusive.

The advanced user identity query feature, to which this configuration statement belongs, relies on the Juniper Identity Management Service that allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting. The feature allows the SRX Series device to query the Juniper Identity Management Service to pull user identity information.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

To obtain device information, such as device identity, groups, and the operating system, from the Juniper Identity Management Service server using either the batch-query or ip-query configuration, you must set the device authentication source, as follows.

```
user@host# set services user-identification device-information authentication-source network-access-controller
```

**Options**

**no-ip-query**—Disable IP query. IP query is enabled by default.

**query-delay-time**—Time after which the SRX Series device sends the query. Rather than allow the SRX Series device to respond automatically by sending a user query *immediately*, you can set a **query-delay-time** parameter, specified in seconds, that allows the SRX Series device to wait for a period of time before sending the query.

**Default:** 15

**Range:** 0-60 seconds

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- port | 569
- primary | 575
- query-api | 585
- secondary | 603
- invalid-authentication-entry-timeout | 526

# ip-user-mapping

**Syntax**

```
ip-user-mapping {
    discovery-method {
        wmi {
            event-log-scanning-interval seconds;
            initial-event-log-timespan hours;
        }
    }
}
```

**Hierarchy Level**

```
[edit services user-identification active-directory domain]
```

**Release Information**
Statement introduced in Junos OS Release 12.1X47-D10.

**Description**
Control how the SRX Series device accesses a domain controller in order to monitor and scan security event logs on the domain controller. By parsing the event log, the SRX Series gets IP address-to-user mappings. This process is part of the integrated user firewall feature. The **ip-user-mapping** statement is optional because WMI is the default discovery method and its properties have default values.

The other available method the SRX Series uses to retrieve address-to-user mapping information is manual (on-demand) probing of a domain PC.

**Options**
**discovery-method**—Method of discover IP address-to-user mappings.

**wmi**—Windows Management Instrumentation (WMI) is the discovery method used to access the domain controller.

**event-log-scanning-interval *seconds***—Optional. Interval at which the SRX Series scans the event log on the domain controller.

**Range:** 5 through 60 seconds

**Default:** 10 seconds

**initial-event-log-timespan *hours***—Optional. Time of the earliest event log on the domain controller that the SRX Series will initially scan. This argument applies to the initial deployment only.

After WMIC and the user identification start working, the SRX Series scans only the latest event log.

**Range:** 1 through 168 hours

**Default:** 1 hour

**Required Privilege Level**

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- active-directory-access | **400**
- clear services user-identification active-directory-access | **711**
- request services user-identification active-directory-access ip-user-probe | **720**
- user-identification (Services) | **665**
- show services user-identification active-directory-access statistics | **795**
- traceoptions (Active Directory Access) | **647**

# ldap-options

### Syntax

```
ldap-options {
    assemble {
        common-name common-name;
    }
    base-distinguished-name base-distinguished-name;
    revert-interval seconds;
    search {
        admin-search {
            distinguished-name distinguished-name;
            password password;
        }
        search-filter filter-name;
    }
}
```

### Hierarchy Level

```
[edit access],
[edit access profile profile-name]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Configure LDAP authentication options.

### Options

The remaining options are explained separately.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# ldap-server

**Syntax**

```
ldap-server server-address {
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    source-address source-address;
    timeout seconds;
    no-tls-certificate-check;
    tls-min-version (v1.1 | v1.2);
    tls-peer-name;
    tls-timeout;
    tls-type {
        start-tls;
    }
}
```

**Hierarchy Level**

```
[edit access]
[edit access profile profile-name]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify that the device uses a Lightweight Directory Access Protocol (LDAP) server for authentication.

**Options**

***server-address***—Address of the LDAP authentication server.

The remaining statements are explained separately.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Obtaining Username and Role Information Through Firewall Authentication | 172

LDAP Functionality in Integrated User Firewall | 194

# level (Services)

**Syntax**

```
level [brief | detail | extensive | verbose];
```

**Hierarchy Level**

```
[edit services ssl traceoptions]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10.

**Description**

Specify the level of debugging the output. This statement is supported on the SRX550M, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX.

**Options**

- *brief*—Specify brief debugging output.
- *detail*—Specify detailed debugging output.
- *extensive*—Specify extensive debugging output.
- *verbose*—Specify verbose debugging output.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

# level (Services User Identification)

**Syntax**

```
level (brief | detail | extensive | verbose);
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass traceoptions]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Configure the level of messages o be written to the trace log file about authentication source behavior.

For the integrated ClearPass authentication enforcement feature, the authentication source is Aruba ClearPass.

**Options**
**all**—Matches all levels.

**error**—Matches error conditions.

**info**—Matches informational messages.

**notice**—Matches conditions that require special handling.

**verbose**—Matches verbose messages.

**warning**—Matches warning messages.

**Required Privilege Level**
services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# lifetime-seconds (Security IKE)

### Syntax

```
lifetime-seconds seconds;
```

### Hierarchy Level

```
[edit security ike proposal proposal-name]
```

### Release Information

Statement introduced in Junos OS Release 8.5. Default value modified in Junos OS Release 10.2.

### Description

Specify the lifetime (in seconds) of an IKE security association (SA). When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated.

### Options

***seconds***—Lifetime of the IKE SA.

**Range:** 180 through 86,400 seconds

**Default:** 28,800 seconds

### Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*IPsec VPN Overview*

*Junos OS User Authentication Methods*

# link (Access)

**Syntax**

```
link pool-name;
```

**Hierarchy Level**

```
[edit access address-assignment pool]
```

**Release Information**

Statement introduced in Release 10.4 of Junos OS.

**Description**

Configure the name of the secondary address-assignment pool that is linked to a primary address-assignment pool. The secondary pool provides a backup pool for local address assignment.

**Options**

*pool-name*—Name of the address assignment pool.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Obtaining Username and Role Information Through Firewall Authentication | 172

# local-authentication-table

**Syntax**

```
local-authentication-table priority priority;
```

**Hierarchy Level**

```
[edit security user-identification authentication-source]
```

**Release Information**
Statement introduced in Junos OS Release 12.1. Support for **disable** option dropped in Junos OS Release 12.1X47-D10.

**Description**
An authentication table created on the SRX Series device using the **request security user-identification local-authentication-table add** command.

**Options**
**priority *priority***—A unique value between 0 and 65535 that determines the sequence for searching multiple tables to retrieve a user role. Each table is given a unique priority value. The lower the value, the higher the priority. A table with priority 120 is searched before a table with priority 200. The default priority value of the local authentication table is 100.

Setting the priority value of the local authentication table to 0 is equivalent to disabling the table and eliminating it from the search sequence.

**Required Privilege Level**
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Understanding User Role Firewalls*

*Understanding the User Identification Table*

# log (Services)

**Syntax**

```
log {
    all;
    errors;
    info;
    sessions-allowed;
    sessions-dropped;
    sessions-ignored;
    sessions-whitelisted;
    warning;
}
```

**Hierarchy Level**

[edit services ssl proxy profile *profile-name* actions]

**Release Information**
Statement introduced in Junos OS Release 12.1X44-D10.

**Description**
Specify the logging actions. When configuring SSL proxy, you can choose to set the option to receive some or all of the logs.

SSL proxy logs contain the logical system name, SSL proxy whitelists, policy information, SSL proxy information, and other information that helps you troubleshoot when there is an error.

You can configure logging of all or specific events, such as error, warning, and information events. You can also configure logging of sessions that are whitelisted, dropped, ignored, or allowed after an error occurs.

**Options**
- **all**—Log all events.
- **errors**—Log all error events.
- **info**—Log all information events.
- **sessions-allowed**—Log SSL session allowed events after an error.
- **sessions-dropped**—Log only SSL session dropped events.
- **sessions-ignored**—Log session ignored events.
- **sessions-whitelisted**—Log SSL session whitelisted events.

- **warning**—Log all warning events.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

# login (Access)

**Syntax**

```
login string;
```

**Hierarchy Level**

```
[edit access firewall-authentication pass-through default-profile profile-name (ftp | http | telnet) banner]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Specify the login banner for users using FTP, HTTP, and Telnet during the authentication process.

**Options**
***string***—Banner text. Maximum length of the message text is 250 characters. Enclose the banner text within spaces or special characters—for example quotation marks (" ").

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35
Obtaining Username and Role Information Through Firewall Authentication | 172

# nas-port-type

**Syntax**

```
nas-port-type {
    ethernet (ethernet);
}
```

**Hierarchy Level**

```
[edit access profile name radius options]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D110 for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

**Description**

RADIUS is an authentication method for validating users trying to access the device using Telnet. Using the **nas-port-type** configuration statement, you can define the type of physical port to authenticate the user.

**Options**

**ethernet**— Translation mechanism for changing the Ethernet value.

**Values:**

- ethernet—Configure the NAS port type as Ethernet

**Required Privilege Level**

access

# network (Access)

**Syntax**

```
network
```

**Hierarchy Level**

```
[edit access address-assignment pool <name> family (inet | inet6)]
```

**Release Information**
Statement introduced in Release 10.4 of Junos OS.

**Description**

Specify the IPv4 network address for the pool. This attribute is mandatory. For an IPv6 pool, you will set the IPv6 network prefix.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# no-remote-trace (Services User Identification)

**Syntax**

```
no-remote-trace;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass traceoptions]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Disable remote tracing.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# no-user-query (Services User Identification)

**Syntax**

```
no-user-query;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Disable the integrated ClearPass authentication and enforcement user query function, if it is configured. You can use the no-user-query statement to turn off the user query function without having to delete the configuration.

The user query function allows the SRX Series device to query the ClearPass webserver for authentication and identity information for an individual user whose information was not posted to the SRX Series device by ClearPass.

**Required Privilege Level**
**services**—To view this statement in the configuration.

**services-control**—To add this statement to the configuration.

# no-tls-certificate-check

**Syntax**

```
no-tls-certificate-check;
```

**Hierarchy Level**

```
[edit access profile profile-name ldap-server ip-address]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D70.

**Description**

Specify validation of the server certificate not required. SRX Series devices support an additional check on the Lightweight Directory Access Protocol (LDAP) server's certificate during the Transport Layer Security (TLS) handshake for LDAP authentication. If the validation of the server certificate is not required, you can use this option to ignore the validation and accept the certificate without checking. By default, this option is disabled.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Example: Configuring Integrated User Firewall on SRX Series | 197

# pass-through

### Syntax

```
pass-through {
    default-profile profile-name;
    ftp {
        banner {
            fail string;
            login string;
            success string;
        }
    }
    http {
        banner {
            fail string;
            login string;
            success string;
        }
    telnet {
        banner {
            fail string;
            login string;
            success string;
        }
}
```

### Hierarchy Level

```
[edit access firewall-authentication]
```

### Release Information

Statement introduced in Junos OS Release 8.5.
HTTPS for pass-through authentication is supported on SRX5400, SRX5600, and SRX5800 devices starting from Junos OS Release 12.1X44-D10 and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.

### Description

Configure pass-through , when a host or user from one zone needs to access a protected resource in another zone. A user must use an FTP, Telnet, or HTTP client to access the IP address of the protected resource and get authenticated by the firewall. The device uses FTP, Telnet, and HTTP to collect username and password information. Subsequent traffic from the user or host is allowed or denied based on the result of this authentication. After the user is authenticated, the firewall proxies the connection.

**Options**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Obtaining Username and Role Information Through Firewall Authentication | 172

# password (Access)

**Syntax**

```
password password;
```

**Hierarchy Level**

```
[edit access ldap-options search admin-search],
[edit access profile profile-name ldap-options search admin-search]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Configure the plain-text password for the administrative user. This password is used in the bind for performing the LDAP search.

**Options**

***password***—Administrative user password.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# password (Services)

**Syntax**

```
password password;
```

**Hierarchy Level**

```
[edit services unified-access-control infranet-controller hostname]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify the password that the SRX Series device should send to the IC Series device to establish communications. The SRX Series device sends the password in its first message to the IC Series device.

This statement is required when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

ca-profile (Services) | 437
server-certificate-subject | 609

# password (System Services)

**Syntax**

```
password password;
```

**Hierarchy Level**

```
[edit system services webapi user]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Specify the password for the integrated ClearPass authentication and enforcement feature Web API daemon (webapi) user.

Range: 1 through 128 characters.

The Web API daemon, acting as an HTTP server, exposes to the Aruba ClearPass Policy Manager (CPPM) an API that allows the CPPM, acting as a client, to send POST request messages to it. The CPPM, which serves as the authentication source, initiates the session to the SRX Series device and sends it user authentication and identity information.

**Required Privilege Level**
system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# permit (Security Policies)

**Syntax**

```
permit {
    application-services {
        application-firewall {
            rule-set rule-set-name;
        }
        application-traffic-control {
            rule-set rule-set-name;
        }
        gprs-gtp-profile profile-name;
        gprs-sctp-profile profile-name;
        idp;
        redirect-wx | reverse-redirect-wx;
        ssl-proxy {
            profile-name profile-name;
        }
        uac-policy {
            captive-portal captive-portal;
        }
        utm-policy policy-name;
    }
    destination-address {
        drop-translated;
        drop-untranslated;
    }
    firewall-authentication {
        pass-through {
            access-profile profile-name;
            client-match user-or-group-name;
            ssl-termination-profile profile-name;
            web-redirect;
            web-redirect-to-https;
        }
        user-firewall {
            access-profile profile-name;
            domain domain-name
            ssl-termination-profile profile-name;
        }
        web-authentication {
            client-match user-or-group-name;
        }
```

```
    }
    services-offload;
    tcp-options {
        sequence-check-required;
        syn-check-required;
    }
    tunnel {
        ipsec-group-vpn group-vpn;
        ipsec-vpn vpn-name;
        pair-policy pair-policy;
    }
}
```

### Hierarchy Level

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* then]

### Release Information

Statement introduced in Junos OS Release 8.5. Support for the **tcp-options** added in Junos OS Release 10.4. Support for the **services-offload** option added in Junos OS Release 11.4. Support for the **ssl-termination-profile** and **web-redirect-to-https** options added in Junos OS Release 12.1X44-D10. Support for the **user-firewall** option added in Junos OS Release 12.1X45-D10.

### Description

Specify the policy action to perform when packets match the defined criteria.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

# pki-local-certificate (Services)

**Syntax**

```
pki-local-certicate pki-certificate;
```

**Hierarchy Level**

```
[edit system services webapi https]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Configure the Web API daemon to use the local X.509 PKI certificate for HTTPS when HTTPS is specified as the communication protocol. The SRX Series integrated ClearPass authentication and enforcement feature exposes the Web API to the ClearPass Policy Mananger (CPPM) to allow the CPPM to initiate a connection to the SRX Series device. For this feature, ClearPass acts as the authentication source. The CPPM uses the HTTPS connection to send user authentication and identity information to the SRX Series device.

**Required Privilege Level**
**services**—To view this statement in the configuration.

**services-control**—To add this statement to the configuration.

# policies

### Syntax

```
policies {
    default-policy (deny-all | permit-all);
    from-zone from-zone-name {
        to-zone;
        policy name {
            description description;
            match (Security Policies Global) {
                source-address (Security Policies);
                destination-address (Security Policies);
                application (Security Policies);
                source-identity;
                source-end-user-profile <source-end-user-profile-name>;
                dynamic-application (Security Policies);
                url-category;
                from-zone (Security Policies Global);
                to-zone (Security Policies Global);
                source-l3vpn-vrf-group [ source-l3vpn-vrf-group ... ];
                destination-l3vpn-vrf-group [ destination-l3vpn-vrf-group ... ];
                destination-address-excluded;
                source-address-excluded;
            }
            scheduler-name scheduler-name;
            then {
                deny;
                permit {
                    application-services {
                        (redirect-wx | reverse-redirect-wx);
                        advanced-anti-malware-policy advanced-anti-malware-policy;
                        application-traffic-control {
                            rule-set rule-set;
                        }
                        gprs-gtp-profile gprs-gtp-profile;
                        gprs-sctp-profile gprs-sctp-profile;
                        icap-redirect icap-redirect;
                        idp;
                        idp-policy idp-policy;
                        security-intelligence-policy security-intelligence-policy;
                        ssl-proxy {
                            profile-name profile-name;
                        }
```

```
    uac-policy {
        captive-portal captive-portal;
    }
    utm-policy utm-policy;
    web-proxy {
        profile-name profile-name;
    }
}
destination-address {
    (drop-translated | drop-untranslated);
}
firewall-authentication {
    pass-through {
        access-profile access-profile;
        auth-only-browser;
        auth-user-agent name;
        client-match [ client-match ... ];
        ssl-termination-profile ssl-termination-profile;
        web-redirect;
        web-redirect-to-https;
    }
    user-firewall {
        access-profile access-profile;
        auth-only-browser;
        auth-user-agent name;
        domain domain;
        ssl-termination-profile ssl-termination-profile;
        web-redirect;
        web-redirect-to-https;
    }
    web-authentication {
        client-match [ client-match ... ];
    }
    push-to-identity-management;
}
services-offload;
tcp-options {
    initial-tcp-mss initial-tcp-mss;
    reverse-tcp-mss reverse-tcp-mss;
    sequence-check-required;
    syn-check-required;
    window-scale;
}
```

```
            tunnel {
                ipsec-vpn ipsec-vpn;
                pair-policy pair-policy;
            }
        }
        reject {
            profile profile;
            ssl-proxy {
                profile-name profile-name;
            }
        }
        count {
        }
        log {
            session-close;
            session-init;
        }
    }
  }
}
```

```
global {
    policy name {
        description description;
        match (Security Policies Global) {
            source-address (Security Policies);
            destination-address (Security Policies);
            application (Security Policies);
            source-identity;
            source-end-user-profile <source-end-user-profile-name>;
            dynamic-application (Security Policies);
            url-category;
            from-zone (Security Policies Global);
            to-zone (Security Policies Global);
            source-l3vpn-vrf-group [ source-l3vpn-vrf-group ... ];
            destination-l3vpn-vrf-group [ destination-l3vpn-vrf-group ... ];
            destination-address-excluded;
            source-address-excluded;
        }
        scheduler-name scheduler-name;
        then {
            deny;
            permit {
                application-services {
                    (redirect-wx | reverse-redirect-wx);
                    advanced-anti-malware-policy advanced-anti-malware-policy;
                    application-traffic-control {
                        rule-set rule-set;
                    }
                    gprs-gtp-profile gprs-gtp-profile;
                    gprs-sctp-profile gprs-sctp-profile;
                    icap-redirect icap-redirect;
                    idp;
                    idp-policy idp-policy;
                    security-intelligence-policy security-intelligence-policy;
                    ssl-proxy {
                        profile-name profile-name;
                    }
                    uac-policy {
                        captive-portal captive-portal;
                    }
                    utm-policy utm-policy;
                    web-proxy {
                        profile-name profile-name;
                    }
```

```
    }
    destination-address {
        (drop-translated | drop-untranslated);
    }
    firewall-authentication {
        pass-through {
            access-profile access-profile;
            auth-only-browser;
            auth-user-agent name;
            client-match [ client-match ... ];
            ssl-termination-profile ssl-termination-profile;
            web-redirect;
            web-redirect-to-https;
        }
        user-firewall {
            access-profile access-profile;
            auth-only-browser;
            auth-user-agent name;
            domain domain;
            ssl-termination-profile ssl-termination-profile;
            web-redirect;
            web-redirect-to-https;
        }
        web-authentication {
            client-match [ client-match ... ];
        }
        push-to-identity-management;
    }
    services-offload;
    tcp-options {
        initial-tcp-mss initial-tcp-mss;
        reverse-tcp-mss reverse-tcp-mss;
        sequence-check-required;
        syn-check-required;
        window-scale;
    }
    tunnel {
        ipsec-vpn ipsec-vpn;
        pair-policy pair-policy;
    }
}
```

```
            reject {
                profile profile;
                ssl-proxy {
                    profile-name profile-name;
                }
            }
            count {
            }
            log {
                session-close;
                session-init;
            }
        }
    }
}
policy-rematch <extensive>;
policy-stats {
    system-wide (disable | enable);
}
pre-id-default-policy {
    then {
        log {
            session-close;
            session-init;
        }
        session-timeout {
            icmp seconds;
            icmp6 seconds;
            ospf seconds;
            others seconds;
            tcp seconds;
            udp seconds;
        }
    }
}
```

```
stateful-firewall-rule name {
    match-direction (input | input-output | output);
    policy name {
        description description;
        match (Security Policies Global) {
            source-address (Security Policies);
            destination-address (Security Policies);
            application (Security Policies);
            source-identity;
            source-end-user-profile <source-end-user-profile-name>;
            dynamic-application (Security Policies);
            url-category;
            from-zone (Security Policies Global);
            to-zone (Security Policies Global);
            source-l3vpn-vrf-group [ source-l3vpn-vrf-group ... ];
            destination-l3vpn-vrf-group [ destination-l3vpn-vrf-group ... ];
            destination-address-excluded;
            source-address-excluded;
        }
        scheduler-name scheduler-name;
        then {
            deny;
            permit {
                application-services {
                    (redirect-wx | reverse-redirect-wx);
                    advanced-anti-malware-policy advanced-anti-malware-policy;
                    application-traffic-control {
                        rule-set rule-set;
                    }
                    gprs-gtp-profile gprs-gtp-profile;
                    gprs-sctp-profile gprs-sctp-profile;
                    icap-redirect icap-redirect;
                    idp;
                    idp-policy idp-policy;
                    security-intelligence-policy security-intelligence-policy;
                    ssl-proxy {
                        profile-name profile-name;
                    }
                    uac-policy {
                        captive-portal captive-portal;
                    }
                    utm-policy utm-policy;
                    web-proxy {
                        profile-name profile-name;
```

```
        }
    }
    destination-address {
        (drop-translated | drop-untranslated);
    }
    firewall-authentication {
        pass-through {
            access-profile access-profile;
            auth-only-browser;
            auth-user-agent name;
            client-match [ client-match ... ];
            ssl-termination-profile ssl-termination-profile;
            web-redirect;
            web-redirect-to-https;
        }
        user-firewall {
            access-profile access-profile;
            auth-only-browser;
            auth-user-agent name;
            domain domain;
            ssl-termination-profile ssl-termination-profile;
            web-redirect;
            web-redirect-to-https;
        }
        web-authentication {
            client-match [ client-match ... ];
        }
        push-to-identity-management;
    }
    services-offload;
    tcp-options {
        initial-tcp-mss initial-tcp-mss;
        reverse-tcp-mss reverse-tcp-mss;
        sequence-check-required;
        syn-check-required;
        window-scale;
    }
    tunnel {
        ipsec-vpn ipsec-vpn;
        pair-policy pair-policy;
    }
}
```

```
            reject {
                profile profile;
                ssl-proxy {
                    profile-name profile-name;
                }
            }
            count {
            }
            log {
                session-close;
                session-init;
            }
        }
    }
}
stateful-firewall-rule-set name {
    stateful-firewall-rule name;
}
traceoptions (Security Policies) {
    file <filename> <files files> <match match> <size size> <(world-readable | no-world-readable)>;
    flag name;
    no-remote-trace;
}
unified-policy {
    max-lookups max-lookups;
}
}
```

### Hierarchy Level

```
[edit security]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.
Support for the **services-offload** option added in Junos OS Release 11.4.
Support for the **source-identitiy** option added in Junos OS Release 12.1.
Support for the **description** option added in Junos OS Release 12.1.
Support for the **ssl-termination-profile** and **web-redirect-to-https** options are added starting from Junos OS Release 12.1X44-D10 and Junos OS Release 15.1X49-D40.
Support for the **user-firewall** option added in Junos OS Release 12.1X45-D10.
Support for the **domain** option, and for the **from-zone** and **to-zone** global policy match options, added in Junos OS Release 12.1X47-D10.
Support for the **initial-tcp-mss** and **reverse-tcp-mss** options added in Junos OS Release 12.3X48-D20.
Support for the **extensive** option for **policy-rematch** added in Junos OS Release 15.1X49-D20.
Starting in Junos OS Release 18.2R1, an IDP policy is available within unified security policy. The IDP policy access is simplified and made available under the unified policy as one of the policy. When an IDP policy is available within a unified security policy, configuring source or destination address, source and destination-except, from and to zone, or application is not required, because the match happens in the security policy itself.
Starting in Junos OS Release 18.3R1, when an SRX Series device is configured with a unified policies, you can configure multiple IDP policies and set one of those policies as the default IDP policy. If multiple IDP policies are configured for a session and when policy conflict occurs, the device applies the default IDP policy for that session and thus resolves any policy conflicts.

> **NOTE:** If you have configured two or more IDP policies in a unified security policy, then you must configure the default IDP policy.

**Description**

Configure a network security policies with IPv6 addresses only if flow support for IPv6 traffic is enabled on the device.

**Options**

**default-policy**—Configure a default action when no user-defined policy match.

**Values:**

- deny-all—Deny all traffic if no policy match
- permit-all—Permit all traffic if no policy match

**policy-rematch**—Re-evaluate the policy when changed.

**Values:**

- extensive—Perform policy extensive rematch

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Security Policies Overview*

# pool (Access)

**Syntax**

```
pool pool-name {
    family {
        inet {
            dhcp-attributes {
                boot-file boot file name;
                boot-server boot server name;
                domain-name domain name;
                grace-period seconds;
                maximum-lease-time (seconds | infinite);
                name-server ipv4-address;
                name-server address;
                netbios-node-type (b-node | h-node | m-node | p-node);
                option dhcp option-identifier-code;
                option-match {
                    option-82 {
                        circuit-id match-value;
                        remote-id match-value;
                    }
                }
                router IPv4 address;
                server-identifier IP address;
                tftp-server server name;
                wins-server IPv4 address;
            }
            host hostname;
            network network address;
            range range-name {
                high upper-limit;
                low lower-limit;
            }
            xauth-attributes {
                primary-dns IP address;
                primary-wins IP address;
                secondary-dns IP address;
                secondary-wins IP address;
            }
        }
        inet6 {
            dhcp-attributes {
                dns-server IPv6-address;
```

```
            grace-period seconds;
            maximum-lease-time seconds;
            option dhcp-option-identifier-code;
            sip-server-address IPv6-address;
            sip-server-domain-name domain-name;
        }
        prefix IPv6-network-prefix;
        range range-name {
            high upper-limit;
            low lower-limit;
            prefix-length delegated-prefix-length;
        }
    link pool-name;
}
```

**Hierarchy Level**

```
[edit access address-assignment]
```

**Release Information**

Statement introduced in Release 10.4 of Junos OS.

**Description**

Configure the name of an address assignment pool. The remaining statements are explained separately.

**Options**

***pool-name***—Name assigned to the address-assignment pool.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Obtaining Username and Role Information Through Firewall Authentication | 172

# port (Access LDAP)

**Syntax**

```
port port-number;
```

**Hierarchy Level**

```
[edit access ldap-server server-address],
[edit access profile profile-name ldap-server server-address]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Configure the port number on which to contact the LDAP server.

**Options**

***port-number***—Port number on which to contact the LDAP server.

**Default:** 389

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# port (Identity Management Advanced Query)

**Syntax**

```
port port;
```

**Hierarchy Level**

```
[edit services user-identification services identity-management connection]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure the port on the Juniper Identity Management Service server that the SRX Series device uses to connect to the server.

The SRX Series device by default always attempts to connect to the primary server. It falls back to the secondary server when its queries to the primary server fail. If the primary server fails, the SRX Series device should be configured to periodically probe the failed primary server and revert to it when it is available.

The advanced user identity query feature relies on the Juniper Identity Management Service to provide a global, end-to-end user identity management solution which allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting. Juniper Identity Management Service provides a centralized identity collection (CIC) system from which the SRX Series device obtains user identity information. It also includes device endpoint context, also referred to as *device identity*, and machine identity (machine ID) information for the user.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the [edit services user-identification] hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Options**

**port**—The port number on the Juniper Identity Management Service server that the SRX Series device connects to.

**Default:** 443

**Range:** 1-65535

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | **275**

connection | **460**

connect-method | **464**

filter | **487**

ip-query | **529**

primary | **575**

# port (Services)

**Syntax**

```
port port-number;
```

**Hierarchy Level**

```
[edit services unified-access-control infranet-controller hostname]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify the port on the IC Series device through which the SRX Series device should establish connections (default 11123). Possible values for this statement range from 1 through 65,535.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

interface (Services) | 524
password (Services) | 550

# port (System Services)

**Syntax**

```
port port-number;
```

**Hierarchy Level**

```
[edit system services webapi http]
[edit system services webapi https]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Specify the SRX Series device TCP port to use for incoming HTTP or HTTPS connection requests initiated by the ClearPass Policy Manager (CPPM). The SRX Series device integrated ClearPass authentication and enforcement feature exposes its Web API (webapi) to the CPPM. The CPPM uses the Web API to establish a connection to the SRX Series device and send user authentication and identity information to it.

**Options**

***port-number***—For HTTP connection protocol.

**Range:** 1 through 65535.

**Default:** 8080

**port *port-number***—For HTTPS connection protocol.

**Range:** 1 through 65535.

**Default:** 8443

**Required Privilege Level**

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

# preferred-ciphers

### Syntax

```
preferred-ciphers (custom | medium | strong | weak);
```

### Hierarchy Level

```
[edit services ssl proxy profile profile-name ]
[edit services ssl termination profile profile-name ]
[edit services ssl initiation profile profile-name]
```

### Release Information

Statement introduced in Junos OS Release 12.1X44-D10.

### Description

Select preferred ciphers. Preferred ciphers allow you to define an SSL cipher that can be used with acceptable key strength. Ciphers are divided in three categories depending on their key strength: strong, medium, or weak.

Custom ciphers allow you to define your own cipher list. If you do not want to use one of the three categories, you can select ciphers from each of the categories to form a custom cipher set. To configure custom ciphers, you must set **preferred-ciphers** to **custom**.

### Options

- **custom**—Configure custom cipher suite and order of preference.
- **medium**—Use ciphers with key strength of 128 bits or greater.
- **strong**—Use ciphers with key strength of 168 bits or greater.
- **weak**—Use ciphers with key strength of 40 bits or greater.

### Required Privilege Level

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

SSL Proxy Overview | 96

# prefix (Access IPv6)

**Syntax**

```
prefix IPv6-network prefix;
```

**Hierarchy Level**

```
[edit access address-assignment pool pool-name family inet6]
```

**Release Information**
Statement introduced in Release 10.4 of Junos OS.

**Description**
Specify the IPv6 prefix for the IPv6 address-assignment pool. This statement is mandatory for IPv6 address-assignment pools.

**Options**
*IPv6-network-prefix*—IPv6 prefix.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# primary connection (Identity Management Advanced Query)

**Syntax**

```
primary {
    address ipv4-address-or-ipv6-address;
    ca-certificate ca-certificate;
    client-id client-id;
    client-secret client-secret;
}
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.
IPv6 address support introduced in Junos OS Release 18.3R1.

**Description**

Configure parameters that the SRX Series device uses to connect to the Juniper Identity Management Service (JIMS) primary server and authenticate to it to obtain an access token. JIMS requires that the SRX Series device use OAuth2 to authenticate to it before the SRX Series device is allowed to query the JIMS server for user identity information. The SRX Series device must provide the JIMS server with credentials, including a client ID and a client secret. If the client is authenticated--in this case the SRX Series device—it is granted an access token. (See RFC 6749.) Both the client ID and the client secret must be consistent with the API client configured on the JIMS primary server.

In addition to configuring the client ID and the client secret, you configure the filename of the JIMS's ca-certificate. The certificate enables the SRX Series device to verify the identity of JIMS and that it is trusted for the SSL connection.

If the deployment configuration consists of more than one JIMS server, a primary and secondary relationship is established. The SRX Series device always attempts to connect to the primary server. When one or more queries to the primary server fails, the system falls back to the secondary server.

**WARNING:** Before you use this feature, you must disable any other actively used options under the [edit services user-identification] hierarchy. You cannot commit this configuration if active directory authentication and the ClearPass query and webapi functions are configured and committed.

**Options**

**address**—IP address of the primary server.

**ca-certificate**—Filename of the JIMS primary server's ca-certificate. Before you configure the ca-certificate filename, the administrator of the JIMS server must export the certificate to the SRX Series device. The administrator must configure the complete path and filename of the certificate on the SRX Series device, for example, '/var/db/RADIUSServerCertificate.crt'. If the ca-certificate is not configured, the SRX Series device cannot verify the certificate.

**NOTE:** The SRX Series device supports a self signed + BASE64 encoded X.509 certificate only.

**client-id**—Client ID that the SRX Series provides to the Juniper Identity Management Service primary server as part of its authentication to it. The SRX Series device must authenticate to the server to obtain an access token that allows the SRX Series device to query the server for user identity information The client ID must be consistent with the API client configured on the JIMS primary server.

**client-secret**—Client secret that the SRX Series provides to the Juniper Identity Management Service primary server as part of its authentication to it. The client secret must be consistent with the API client configured on the JIMS primary server.

**Required Privilege Level**

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- connect-method | 464
- filter | 487
- ip-query | 529
- port | 569
- query-api | 585
- authentication-entry-timeout | 420
- batch-query | 426

# priority (Security User Identification)

**Syntax**

```
authentication-source {
    active-directory priority priority;
    aruba-clearpass priority priority;
    firewall-authentication priority priority;
    local-authentication-table priority priority;
    unified-access-control priority priority;
}
```

**Hierarchy Level**

```
[edit security user-identification]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Set the lookup priority to identify the order in which the SRX Series device checks its configured authentication tables for user authentication information. Authentication tables are searched in order based on their priority setting in which lowest value takes precedence.

For the integrated ClearPass authentication and enforcement feature, the SRX Series device must be configured to search the ClearPass authentication table first.

> **NOTE:** Note that both the authentication source, Aruba ClearPass, and the SRX Series ClearPass authentication table are both referred to as aruba-clearpass in the CLI and its output.

You need to set this value only if the local authentication table, whose default value is 100, also resides on the Packet Forwarding Engine. In that case, you must configure a higher priority value, such as 120, for the local authentication table.

**Options**

***priority***—Aruba-clearpass authentication table search priority.

**Range:** 1 through 65535.

**Default:** 110.

Default values for other authentication tables:

- Local authentication table: 100
- Active Directory (AD) table: 125
- UAC authentication table: 150
- Firewall authentication table: 200

**Required Privilege Level**

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

# push-to-identity-management

**Syntax**

```
push-to-identity-management;
```

**Hierarchy Level**

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* then permit firewall-authentication]

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Configure firewall authentication to push authentication entries with a successful authentication state to the Juniper Identity Management Service server. You use this statement in conjunction with the query-api/user statement, which sets the path of the URL for querying user identities.

When the SRX Series device does not have authentication information for a user based on the user's IP address, it can force the user to authenticate through captive portal to obtain the user ID information and authenticate the user. If a security policy that specifies firewall authentication is configured with the **push-to-identity-management** statement, the user information is pushed to the Juniper Identity Management Service server.

After you push the entry to the Juniper Identity Management Service server, you can use the batch query function to obtain authentication information for that user from the Juniper Identity Management Service server, including the groups that the user belongs to.

**NOTE:** The SRX Series device does not update the authentication-entry time-out state to Juniper Identity Management Service.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275

query-api | 585
filter | 487
port | 569
primary | 575
secondary | 603
token-api | 639

# protocol-version

**Syntax**

```
protocol-version (all | tls1 | tls11 | tls12);
```

**Hierarchy Level**

```
[edit services ssl termination profile profile-name]
[edit services ssl initiation profile profile-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10. The **tls11** and **tls12** options are introduced in 15.1X49-D30.

**Description**

Specify the accepted SSL protocol version.

You can specify the SSL/TLS protocol version the SRX Series device uses to negotiate in SSL connections.

**Options**

- **all**—Accept all versions of TLS.
- **TLS version 1.0**—Accept TLS version 1.0. It provides secure communication over networks by providing privacy and data integrity between communicating applications
- **TLS version 1.1**—Accept TLS version 1.1. This enhanced version of TLS provides protection against cipher-block chaining (CBC) attacks.
- **TLS version 1.2** —Accept TLS version 1.2. This enhanced version of TLS provides improved flexibility for negotiation of cryptographic algorithms.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35
SSL Proxy Overview | 96

# query-api (Services User Identification)

**Syntax**

```
query-api query-api
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Configure **query-api** to specify the path of the URL that the SRX Series device uses to query the ClearPass Policy Manager (CPPM) webserver for authentication and identity information for an individual user. For the SRX Series device to be able to make a request, you must have configured it to obtain an access token. See **token-api (Services User Identification)**.

The integrated ClearPass authentication and enforcement user query function supplements the Web API function (webapi) by allowing the SRX Series device to obtain from the CPPM authentication information for an individual user whose information does not already exist in the SRX Series ClearPass authentication table.

Consider the following **query-api** example:

```
api/v1/insight/endpoint/ip/$IP$
```

The SRX Series device generates the complete URL for the user query request by combining the query-api string with the connection method (HTTPS) and the CPPM webserver IP address ({$server}).

```
https://{$server}/api/v1/insight/endpoint/ip/$IP$
```

In this example, the SRX Series device replaces the variables with the following values resulting in a specific URL request for the individual user:

```
https://203.0.113.76/api/v1/insight/endpoint/ip/192.0.2.98
```

Under normal circumstances, the ClearPass webserver sends user authentication information to the SRX Series device in POST request messages and the SRX Series device writes that information to its ClearPass

authentication table. When the SRX Series device receives an access request from a user, it searches its ClearPass authentication table for an entry for that user.

It can happen that the SRX Series device might not have received authentication for a user from the CPPM because the user has not yet been authenticated by the CPPM. For example, the user might have joined the network through an access layer not on a managed switch or WLAN. When the CPPM receives the user query from the SRX Series device, it authenticates the user and returns the authentication information to the device.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# query-api (advanced user query)

**Syntax**

```
query-api query-api;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
Configure the prefix of the URL path for querying user identities. This value is used to construct the prefix of the path for queries for individual users, as well as for ip-query and batch-query requests, each of which has a unique suffix:

- For user-query *query-api*/user
- For IP query, *query-api*/ip/
- For batch query, *query-api*/users/

For example, for individual user queries, you can obtain user information, such as the groups that the user belongs to, from the Juniper Identity Management Service server. When the SRX Series device does not have authentication information for a user based on the user's IP address, it can force the user to authenticate through captive portal to obtain the user ID information and authenticate the user.

If a security policy that specifies firewall authentication is configured with the **push-to-identity-management** statement, the user information is pushed to the JIMS server.

After you push the user identity entry to the JIMS server, you can obtain the user identity information, including information such as groups that the user belongs to and information about the user's device. This information is returned to you in the next batch query response.

To generate the complete URL for a user query, the prefix **user_query/v2** is combined with the connection method, which is HTTPS, the IP address of the Juniper Identity Management Service server, the IP address of the user's device, and the domain name.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- push-to-identity-management | 580
- filter | 487
- ip-query | 529
- port | 569
- primary | 575
- token-api | 639

# radius-options (Access)

### Syntax

```
radius-options {
    revert-interval seconds;
}
```

### Hierarchy Level

```
[edit access];
[edit access profile profile-name]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Configure RADIUS options.

### Options

The remaining statement is explained separately.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# radius-server (Access)

### Syntax

```
radius-server server-address {
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    secret password;
    source-address source-address;
    timeout seconds;
}
```

### Hierarchy Level

```
[edit access],
[edit access profile profile-name]
```

### Release Information

Statement modified in Junos OS Release 8.5.

### Description

Configure RADIUS for Layer 2 Tunneling Protocol (L2TP) or Point-to-Point Protocol (PPP) authentication.

To configure multiple RADIUS servers, include multiple **radius-server** statements. The servers are tried in order and in a round-robin fashion until a valid response is received from one of the servers or until all the configured retry limits are reached.

### Options

***server-address***—Address of the RADIUS authentication server.

The remaining statements are explained separately.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# range (Access)

### Syntax

```
range range-name {
    high upper-limit;
    low lower-limit;
    prefix-length delegated-prefix-length;
}
```

### Hierarchy Level

```
[edit access address-assignment pool pool-name family inet6]
[edit access address-assignment pool pool-name family inet]
```

### Release Information

Statement introduced in Release 10.4 of Junos OS.

### Description

Configure an IP name range used within an address-assignment pool. For IPv4, you do not create a prefix-length.

### Options

- *range-name*—Name of the range.
- **high** *upper-limit*—Upper limit of IPv6 address range.
- **low** *lower-limit*—Lower limit of IPv6 address range.
- **prefix-length** *delegated-prefix-length*—IPv6 delegated prefix length.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# rate-limit (Security Log)

**Syntax**

```
rate-limit rate-limit;
```

**Hierarchy Level**

```
[edit security log stream stream-name]
[edit logical-systems name security log stream stream-name]
[edit tenants tenant-name security log stream stream-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.
The [edit **logical-systems** *name* security log stream] hierarchy level introduced in Junos OS Release 18.2R1.
The [edit **tenants** *tenant-name* security log stream] hierarchy level introduced in Junos OS Release 18.3R1.

**Description**

The Integrated Authentication and Enforcement feature sends threat and attack logs generated by the SRX Series device security modules to the ClearPass Policy Manager (CPPM) to use in its security policy assessment.

The logs are sent in stream mode. To avoid overburdening the SRX Series device and the log server, you can control the rate at which these logs are sent. By setting a rate-limit value, you can constrain the number of logs that are sent in 1 second. After the limit is reached, no more logs are sent.

Range: 1 through 65,535.

**NOTE:** For devices with multicore systems that use SPUs, each SPU is programmed with the configured-rate, which results in an aggregate-rate proportional to the number of SPUs.

```
SPU configured-rate = aggregate-rate/number-of-SPUs
```

Rate limiting on SRX5400, SRX5600, and SRX5800 devices is generally not as accurate as it is on SRX100, SRX210, SRX220, SRX240, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, and SRX4200 devices, because the generation of logs is not entirely balanced between SPUs.

**Required Privilege Level**

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

# redirect-traffic

**Syntax**

```
redirect-traffic (all | unauthenticated);
```

**Hierarchy Level**

```
[edit services unified-access-control captive-portal policy]
```

**Release Information**

Statement introduced in Junos OS Release 10.2.

**Description**

Specify to redirect traffic destined for protected sources to the IC Series device. You can choose to redirect all traffic or only unauthenticated traffic.

**Options**

- **all**—Redirect all traffic destined for the protected sources to the IC Series device. Specify this option if you want to redirect all traffic (IPsec or source IP) to the currently connected IC Series device or to an IP address or domain name that you specify in a redirect URL.
- **unauthenticated**—Redirect unauthenticated traffic destined for the protected sources to the IC Series device. Select this option if your deployment uses source IP only or a combination of source IP and IPsec. The Junos OS Enforcer redirects clear-text traffic from unauthenticated users to the currently connected IC Series device or to an IP address or domain name that you specify in a redirect URL.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# redirect-url

**Syntax**

```
redirect-url url;
```

**Hierarchy Level**

```
[edit services unified-access-control captive-portal policy]
```

**Release Information**

Statement introduced in Junos OS Release 10.2.

**Description**

Specify to redirect traffic destined for protected sources to a specified URL.

You can configure the following options in the redirect URL string:

- ***%dest-url%***—Specifies the protected resource which the user is trying to access.
- ***%enforcer-id%***—Specifies the ID assigned to the Junos OS Enforcer by the IC Series device.
- ***%policy-id%***—Specifies the encrypted policy ID for the security policy that redirected the traffic.
- ***%dest-ip%***—Specifies the IP address or hostname of the protected resource that the user is trying to access.
- ***%ic-ip%***—Specifies the IP address or hostname of the IC Series device to which the Junos OS Enforcer is currently connected.

If you do not specify the redirect URL, the Junos OS Enforcer uses the following default configuration:

```
https://%ic-ip%/?target = %dest-url%  &enforcer = %enforcer-id%  &policy = %policy-id%
  &dest-ip = %dest-ip%
```

**NOTE:** The maximum size of a redirect payload is 1450 bytes. The size of the redirect URL is restricted to 1407 bytes (excluding a few HTTP headers). If a user accesses a destination URL that is larger than 1407 bytes, the Infranet Controller authenticates the payload, calculates the exact length of the redirect URL, and trims the destination URL so that it can fit into the redirect URL. The destination URL can be fewer than 1407 bytes based on what else is present in the redirect URL (for example, policy ID). The destination URL in the default redirect URL is trimmed so that the redirect packet payload size is limited to 1450 bytes. If the length of the payload is larger than 1450 bytes, the excess length is trimmed and the user is directed to the destination URL that has been resized to 1450 bytes.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# retry (Access LDAP)

**Syntax**

```
retry attempts;
```

**Hierarchy Level**

```
[edit access ldap-server server-address],
[edit access profile profile-name ldap-server server-address]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify the number of retries that a device can attempt to contact an LDAP server.

**Options**

***attempts***—Number of retries that the device is allowed to attempt to contact an LDAP server.

**Range:** 1 through 10

**Default:** 3

**Required Privilege Level**

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# retry (Access RADIUS)

**Syntax**

```
retry attempts;
```

**Hierarchy Level**

```
[edit access radius-server server-address],
[edit access profile profile-name radius-server server-address]
```

**Release Information**

Statement modified in Release 8.5 of Junos OS.

**Description**

Specify the number of retries that a device can attempt to contact a RADIUS authentication server.

**Options**

***attempts***—Number of retries that the device is allowed to attempt to contact a RADIUS server.

**Range:** 1 through 10

**Default:** 3

**Required Privilege Level**

secret—To view this statement in the configuration.

secret-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# revert-interval (Access LDAP)

**Syntax**

```
revert-interval seconds;
```

**Hierarchy Level**

```
[edit access ldap-options],
[edit access profile profile-name ldap-options]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify the amount of time that elapses before the primary server is contacted if a backup server is being used.

**Options**

***seconds***—Number of seconds that elapse before the primary server is contacted.

**Range:** 60 through 4,294,967,295 seconds

**Default:** 600 seconds

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# revert-interval (Access RADIUS)

**Syntax**

```
revert-interval seconds;
```

**Hierarchy Level**

```
[edit access radius-options]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify the amount of time that elapses before the primary server is contacted if a backup server is being used.

**Options**

***seconds***—Number of seconds that elapse before the primary server is contacted.

**Range:** 60 through 4,294,967,295 seconds

**Default:** 600 seconds

**Required Privilege Level**

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# root-ca (Services)

### Syntax

```
root-ca root-certificate;
```

### Hierarchy Level

```
[edit services ssl proxy profile profile-name]
[edit services ssl termination profile profile-name]
```

### Release Information

Statement introduced in Junos OS Release 12.1X44-D10.

### Description

Root certificate for interdicting server certificates in proxy mode. This statement is supported on the SRX1500, SRX5400, SRX5600, and SRX5800 devices and vSRX.

### Options

*root-ca-name*—Specify root certificate for interdicting server certificates in proxy mode.

### Required Privilege Level

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

Firewall User Authentication Overview | 35

# routing-instance (Access LDAP)

### Syntax

```
routing-instance routing-instance-name;
```

### Hierarchy Level

```
[edit access ldap-server server-address],
[edit access profile profile-name ldap-server server-address]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Configure the routing instance used to send LDAP packets to the LDAP server. A routing instance is a collection of routing tables, the interfaces contained in the routing tables, and the routing protocol parameters that control the information in the routing tables.

### Options

***routing-instance-name***—Name of the routing instance.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# routing-instance (Access RADIUS)

### Syntax

```
routing-instance routing-instance-name;
```

### Hierarchy Level

```
[edit access radius-server server-address],
[edit access profile profile-name radius-server server-address]
```

### Release Information

Statement modified in Release 8.5 of Junos OS.

### Description

Configure the routing instance used to send RADIUS packets to the RADIUS server. A routing instance is a collection of routing tables, the interfaces contained in the routing tables, and the routing protocol parameters that control the information in the routing tables.

### Options

***routing-instance-name*** —Name of the routing instance.

### Required Privilege Level

secret—To view this statement in the configuration.
secret-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# search

### Syntax

```
search {
    admin-search {
        distinguished-name distinguished-name;
        password password;
    }
    search-filter filter-name;
}
```

### Hierarchy Level

```
[edit access ldap-options],
[edit access profile profile-name ldap-options]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify that a search is used to get a user's LDAP distinguished name (DN). The search is performed based on the search filter and the part typed in by the user during authentication.

### Options

The remaining statements are explained separately.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# search-filter

**Syntax**

```
search-filter filter-name;
```

**Hierarchy Level**

```
[edit access ldap-options search],
[edit access profile profile-name ldap-options search]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Specify that a search filter is used to find the user's LDAP distinguished name (DN). For example, a filter of **cn** specifies that the search matches a user whose common name is the username.

**Options**

***filter-name***—Name of the filter used to find the user's distinguished name.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# secondary connection (Identity Management Advanced Query)

**Syntax**

```
secondary {
    address ipv4-address-or-ipv6-address;
    ca-certificate ca-certificate;
    client-id client-id;
    client-secret client-secret;
}
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.
IPv6 address support introduced in Junos OS Release 18.3R1.

**Description**

Configure parameters that the SRX Series device uses to connect to the Juniper Identity Management Service (JIMS) secondary server and authenticate to it in order to obtain an access token. JIMS requires that the SRX Series device use OAuth2 to authenticate to it before the SRX Series device is allowed to query the JIMS server for user identity information. The SRX Series device must provide the JIMS server with credentials, including a client ID and a client secret. If the client is authenticated--in this case the SRX Series device—it is granted an access token. (See RFC 6749.) Both the client ID and the client secret must be consistent with the API client configured on the JIMS Service primary server.

In addition to configuring the client ID and the client secret, you configure a ca-certificate for the secondary server, if one exists. You configure the file name of the JIMS's ca-certificate. The certificate enables the SRX Series device to verify the identity of JIMS and that it is trusted for the SSL connection.

The SRX Series device always attempts to connect to the primary server first. When one or more queries to the primary server fails, the system falls back to the secondary server.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Options**

**address**—IP address of the secondary server.

**ca-certificate**—File name of the ca-certificate for the secondary server. Before you configure the ca-certificate file name, the administrator of the JIMS server must export the certificate and import it to the SRX Series device. The administrator must configure the complete path and file name of the certificate on the SRX Series device, for example, '/var/db/RADIUSServerCertificate.crt'. If the ca-certificate is not configured, the SRX Series device can not verify the JIMS certificate.

**NOTE:** The SRX Series device supports a self signed + BASE64 encoded X.509 certificate only.

**client-id**—Client ID that the SRX Series provides to the JIMS Service secondary server as part of its authentication to it. The SRX Series device must authenticate to the server to obtain an access token that allows the SRX Series device to query the server for user identity information The client ID must be consistent with the API client configured on the JIMS primary server.

**client-secret**—Client secret that the SRX Series provides to the JIMS secondary server as part of its authentication to it. The client secret must be consistent with the API client configured on the JIMS secondary server.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275
- connect-method | 464
- filter | 487
- ip-query | 529
- port | 569
- query-api | 585
- authentication-entry-timeout | 420
- batch-query | 426

# secret (Access Profile)

### Syntax

```
secret password;
```

### Hierarchy Level

```
[edit access profile profile-name radius-server server-address]
```

### Release Information

Statement modified in Release 8.5 of Junos OS.

### Description

Specify the RADIUS secret password, which is shared between the router and the RADIUS server. The device uses this secret to encrypt the user's password that is sent to the RADIUS server.

### Options

***password***—RADIUS secret. Maximum length is 256 characters.

### Required Privilege Level

secret—To view this statement in the configuration.
secret-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# securid-server

### Syntax

```
securid-server {
    server-name configuration-file filepath;
}
```

### Hierarchy Level

```
[edit access]
```

### Release Information

Statement introduced in Release 9.1 of Junos OS.

### Description

Configure SecurID server for SecurID authentication type.

### Options

The remaining statement is explained separately.

**NOTE:** You can configure only one SecurID server. SecurID challenges are not yet supported.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# separator

**Syntax**

```
separator special-character;
```

**Hierarchy Level**

```
[edit access profile profile-name client-name-filter client-name]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Specify a character to identify where stripping of characters occurs in a client name. Stripping removes characters to the right of each instance of the specified character, plus the character itself. The stripping begins with the rightmost separator character.

Use the **separator** statement with the **count** statement to determine which characters in a client name are stripped. If the specified number of separator characters (count) exceeds the actual number of separator characters in the client name, stripping stops at the last available separator character.

**Options**
***special-character***—Character used to identify where to start the stripping of characters in a client name.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# server-certificate (Services)

**Syntax**

```
server-certificate server-certificate;
```

**Hierarchy Level**

```
[edit services ssl termination profile profile-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10. This statement is supported on the SRX1500, SRX5400, SRX5600, and SRX5800 devices and vSRX.

**Description**

Specify the local certificate identifier.

**Options**

**server-certificate**—Specify the name of the local certificate identifier.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

# server-certificate-subject

**Syntax**

```
server-certificate-subject subject;
```

**Hierarchy Level**

```
[edit services unified-access-control infranet-controller hostname]
```

**Release Information**

Statement introduced in Junos OS Release 9.4.

**Description**

Optionally specify the full subject name of the certificate that the SRX Series device should use to validate the IC Series device's server certificate.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

ca-profile (Services) | 437
password (Services) | 550

# session-options (Access Profile)

### Syntax

```
session-options {
    client-group [group-names];
    client-idle-timeout minutes;
    client-session-timeout minutes;
}
```

### Hierarchy Level

```
[edit access profile profile-name]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Define options that control a user's session after successful authentication.

### Options

The remaining statements are explained separately.

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# size (Services)

**Syntax**

```
size size;
```

**Hierarchy Level**

```
[edit services ssl traceoptions file file-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10.

**Description**

Specify the maximum trace file size. This statement is supported on the SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX.

**Options**

**size**—Specify the maximum trace file size.

**Range**: 10,240 to 1,073,741,824.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99
Firewall User Authentication Overview | 35

# source-address (Access LDAP)

**Syntax**

```
source-address source-address;
```

**Hierarchy Level**

```
[edit access ldap-server server-address],
[edit access profile profile-name ldap-server server-address]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Configure a source address for each configured LDAP server. Each LDAP request sent to a LDAP server uses the specified source address.

**Options**

***source-address***—Valid IP address configured on one of the device interfaces.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# source-address (Access RADIUS)

**Syntax**

```
source-address source-address;
```

**Hierarchy Level**

```
[edit access radius-server server-address],
[edit access profile profile-name radius-server server-address]
```

**Release Information**

Statement modified in Junos OS Release 8.5.

**Description**

Configure a source address for each configured RADIUS server. Each RADIUS request sent to a RADIUS server uses the specified source address.

**Options**

*source-address—Valid IP address configured on one of the device interfaces.*

**Required Privilege Level**

secret—To view this statement in the configuration.
secret-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# source-end-user-profile

**Syntax**

```
source-end-user-profile device-identity-profile-name;
```

**Hierarchy Level**

```
[edit security policies from-zone from-zone to-zone to-zone policy policy-name match]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D70.

**Description**

The **source-end-user-profile** field in a security policy enables you to specify a device identity profile that identifies the traffic source based on the device from which the traffic issued. The security policy action is applied to traffic issuing from a device if the device matches the attributes specified in the profile and it matches the rest of the security policy parameters.

The device identity profile feature provides a solution for cases in which you cannot or do not want to use the user identity to control access to network resources. The device identity feature allows you to use the identity of a device and its attributes to control access to network resources instead of the identity of the user of that device.

You might want to control network access based on the device identity for various reasons. For example, you might allow your users to use their own devices (BYOD) to access network resources and you do not want to use captive portal authentication. Also, some companies might have older switches that do not support 802.1, or they might not have a Network Access Control (NAC) system.

**Options**

**device-identity-profile-name**—Device identity profile that specifies characteristics that can apply to one or more devices.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Access Control to Network Resources Based on Device Identity Information | 244

Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature | 247

Understanding the Device Identity Authentication Table and Its Entries | 252

# source-identity-log (Security)

**Syntax**

```
source-identity-log
```

**Hierarchy Level**

```
[edit security zones security-zone zone-name]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D60.

**Description**
Specify the source-identity-log parameter as part of the configuration for a zone to enable it to trigger user identity logging when that zone is used as the source zone (from-zone) in a security policy. If a zone is configured for zone-based user identity logging and it is used as the source zone in a security policy, the system logs the user identity of any user who belongs to that zone and whose traffic matches the security policy's terms.

A zone configured for zone-based user identity logging is reusable. That is, you can use it as the source zone in any security policy.

For zone-based user identity logging to occur, you must have configured the session initialization (session-init) and the session termination (session-close) events as actions for the security policy.

Zone-based user identity logging allows you to broaden the scope of users whose identities are recorded in the session log. The source-identity security policy tuple writes the user or group name to log, but it restricts application of the security policy to the specified user or user group.

**Required Privilege Level**
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- Understanding How to Include User Identity Information in the Session Log File Based on the Source Zone | 235
- Example: Configuring Integrated User Firewall to Write User Identity to the Session Log Based On the Source Zone | 237
- Overview of Integrated User Firewall | 177

Example: Configuring Integrated User Firewall on SRX Series | 197

# ssl (Services)

**Syntax**

```
ssl {
    initiation {
        profile profile-name {
            actions {
                ignore-server-auth-failure;
            }
            client-certificate;
            custom-ciphers [cipher];
            enable-flow-tracing;
            enable-session-cache;
            preferred-ciphers (custom | medium | strong | weak);
            protocol-version (all | tls1 | tls11 | tls12);
            trusted-ca (all | [ca-profile] );
        }
    }
    proxy {
        global-config {
            session-cache-timeout seconds;
        }
        profile profile-name {
            actions {
                crl {
                    disable;
                    if-not-present (allow | drop);
                    ignore-hold-instruction-code;
                }
                disable-session-resumption;
                ignore-server-auth-failure;
                log {
                    all;
                    errors;
                    info;
                    sessions-allowed;
                    sessions-dropped;
                    sessions-ignored;
                    sessions-whitelisted;
                    warning;
                }
                renegotiation {
                    (allow | allow-secure | drop);
```

```
            }
        }
        custom-ciphers [cipher];
        enable-flow-tracing;
        preferred-ciphers (custom | medium | strong | weak);
        root-ca root-certificate;
        trusted-ca (all | [ca-profile] );
        whitelist [global-address-book-addresses];
    }
  }
  termination {
    profile profile-name {
        custom-ciphers [cipher];
        enable-flow-tracing;
        enable-session-cache;
        preferred-ciphers (custom | medium | strong | weak);
        protocol-version (all | tls1 | tls11 | tls12);
        server-certificate certificate-identifier;
    }
  }
  traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        (no-world-readable | world-readable);
        size maximum-file-size;
    }
    flag flag;
    level [brief | detail | extensive | verbose];
    no-remote-trace;
  }
}
```

**Hierarchy Level**

```
[edit services]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10. The **crl** statement is supported from 15.1X49-D30. The **protocol-version** statement is updated to include **tls11** and **tls12** from Junos OS Release 15.1X49-D30.

**Description**

Specify the configuration for Secure Socket Layer (SSL) support service. This statement is supported on the SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX.

**Options**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

Firewall User Authentication Overview | 35

# ssl-termination-profile

**Syntax**

```
ssl-termination-profile profile-name;
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    pass-through]
```

**Release Information**
Statement introduced in Junos OS Release 12.1X44-D10.

**Description**
Specify the SSL termination profile used for SSL offloading.

**Options**
***profile-name***—Specify the name of the SSL termination profile used to the SSL offload.

**Required Privilege Level**
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Security Policies Overview*

# success

### Syntax

```
success string;
```

### Hierarchy Level

```
[edit access firewall-authentication pass-through default-profile name (ftp | http | telnet) banner],
[edit access firewall-authentication web-authentication]
```

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Specify the banner (message) that users see when trying to connect using FTP, HTTP, or Telnet after successful authentication.

### Options

***string***—Banner text. Maximum length of the message text is 250 characters. Enclose the banner text within spaces or special characters—for example, quotation marks (” “).

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# system-generated-certificate

**Syntax**

```
system-generated-certificate;
```

**Hierarchy Level**

```
[edit system services web-management https]
```

**Release Information**
Command introduced in Junos OS Release 11.1 for EX Series switches.

**Description**
Configure the automatically generated self-signed certificate for enabling HTTPS services.

**Required Privilege Level**
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

- *Enabling HTTPS and XNM-SSL Services on Switches Using Self-Signed Certificates (CLI Procedure)*

# telnet (Access)

**Syntax**

```
telnet {
    banner {
        fail string;
        login string;
        success string;
    }
}
```

**Hierarchy Level**

```
[edit access firewall-authentication pass-through]
```

**Release Information**
Statement introduced in Release 8.5 of Junos OS.

**Description**
Configure banners for Telnet login prompt, successful authentication, and failed authentication.

**Options**
The remaining statements are explained separately.

**Required Privilege Level**
access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# termination (Services)

### Syntax

```
termination {
    profile profile-name {
        custom-ciphers [cipher];
        enable-flow-tracing;
        enable-session-cache;
        preferred-ciphers (custom | medium | strong | weak);
        protocol-version (all | tls1 | tls11 | tls12);
        server-certificate certificate-identifier;
    }
}
```

### Hierarchy Level

```
[edit services ssl]
```

### Release Information

Statement introduced in Junos OS Release 12.1X44-D10. The **protocol-version** statement is updated to include **tls11** and **tls12** from Junos OS Release 15.1X49-D30.

### Description

Specify the configuration for Secure Socket Layer (SSL) termination support service.

Following types of SSL profiles are supported on SRX Series to secure connections based on the role of the SRX Series device:

- SSL initiation: The SRX Series device, acting as an SSL proxy client, initiates and maintains SSL sessions between itself and an SSL server. SRX device receives unencrypted data from an HTTP client, and encrypts and transmits the data as ciphertext to the SSL server.
- SSL termination: The SRX Series device, actings as an SSL proxy server, terminates the SSL session from the client and then establishing a new SSL connection to the server. The SRX Series device decrypts the data and then sends the data as un-encrypted request to the other servers (HTTP server).

The SSL proxy profile will be applied to the security policy as application services.

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# test-only-mode

**Syntax**

```
test-only-mode (true | false):
```

**Hierarchy Level**

```
[edit services unified-access-control ]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Configure the device in test-only mode to log access decisions from the IC Series device without actually enforcing the decisions. When configured in test-only mode, the SRX Series device enables all UAC traffic to go through so you can test the implementation without impeding traffic.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

# then (Security Policies)

**Syntax**

```
then {
    count {
        alarm {
            per-minute-threshold number;
            per-second-threshold number;
        }
    }
    deny;
    log {
        session-close;
        session-init;
    }
    permit {
        application-services {
            application-firewall {
                rule-set rule-set-name;
            }
            application-traffic-control {
                rule-set rule-set-name;
            }
            gprs-gtp-profile profile-name;
            gprs-sctp-profile profile-name;
            idp;
            redirect-wx | reverse-redirect-wx;
            ssl-proxy {
                profile-name profile-name;
            }
            uac-policy {
                captive-portal captive-portal;
            }
            utm-policy policy-name;
        }
        destination-address {
            drop-translated;
            drop-untranslated;
        }
        firewall-authentication {
            pass-through {
                access-profile profile-name;
                client-match user-or-group-name;
```

```
            ssl-termination-profile profile-name;
            web-redirect;
            web-redirect-to-https;
        }
        user-firewall {
            access-profile profile-name;
            domain domain-name
            ssl-termination-profile profile-name;
        }
        web-authentication {
            client-match user-or-group-name;
        }
    }
    services-offload;
    tcp-options {
        initial-tcp-mss mss-value;
        reverse-tcp-mss mss-value;
        sequence-check-required;
        syn-check-required;
    }
    tunnel {
        ipsec-group-vpn group-vpn;
        ipsec-vpn vpn-name;
        pair-policy pair-policy;
    }
  }
  reject;
}
```

**Hierarchy Level**

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name*]

**Release Information**

Statement introduced in Junos OS Release 8.5. Support for the **services-offload** option added in Junos OS Release 11.4. Support for the **ssl-termination-profile** and **web-redirect-to-https** options added in Junos OS Release 12.1X44-D10. Support for the **user-firewall** option added in Junos OS Release 12.1X45-D10. Support for the **initial-tcp-mss** and **reverse-tcp-mss** options added in Junos OS Release 12.3X48-D20.

**Description**

Specify the policy action to be performed when packets match the defined criteria.

**Options**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

- *Security Policies Overview*
- *Understanding Security Policy Rules*
- *Understanding Security Policy Elements*

# timeout (Access LDAP)

### Syntax

**timeout** ***seconds*****;**

### Hierarchy Level

[edit access ldap-server *server-address*]
[edit access profile *profile-name* ldap-server *server-address*]

### Release Information

Statement introduced in Release 8.5 of Junos OS.

### Description

Configure the amount of time that the local device waits to receive a response from an LDAP server.

### Options

***seconds***—Amount of time to wait.

**Range:** 1 through 90 seconds

**Default:** 3 seconds

### Required Privilege Level

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

# timeout (Access RADIUS)

### Syntax

```
timeout seconds;
```

### Hierarchy Level

```
[edit access radius-server server-address]
[edit access profile profile-name radius-server server-address]
```

### Release Information

Statement modified in Release 8.5 of Junos OS.

### Description

Configure the amount of time that the local device waits to receive a response from a RADIUS server.

### Options

***seconds***—Amount of time to wait.

**Range:** 1 through 90 seconds

**Default:** 3 seconds

### Required Privilege Level

secret—To view this statement in the configuration.

secret-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# timeout (Services)

**Syntax**

```
timeout seconds;
```

**Hierarchy Level**

```
[edit services unified-access-control ]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify the value, in seconds, that the SRX Series device should wait to get a heartbeat response from an IC Series UAC Appliance (default is 300). If the SRX Series device does not receive it in the specified time, it takes the action specified by the **timeout-action** configuration statement. It also tries again to make a connection to the IC Series appliance. After the second failed attempt, the SRX Series device fails over to the next IC Series appliance in the cluster. The SRX Series device continues trying to reach IC Series appliances in the cluster until a connection is established.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series appliance. When working with a cluster of IC Series appliances, the Junos OS Enforcer connects to one at a time, failing over to other IC Series appliances in the cluster as required.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

interval (Services) | 525
timeout-action | 633

# timeout-action

**Syntax**

```
timeout-action (close | no-change | open):
```

**Hierarchy Level**

```
[edit services unified-access-control ]
```

**Release Information**
Statement introduced in Junos OS Release 9.4.

**Description**
Specify what the SRX Series device should do when a timeout occurs and the device cannot connect to an Infranet Enforcer.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series UAC Appliance.

**Options**

- **close**—Close existing sessions and block any further traffic. This is the default option.
- **no-change**—Preserve existing sessions and require authentication for new sessions.
- **open**—Preserve existing sessions and allow new sessions access.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

interval (Services) | 525
timeout (Services) | 632

# tls-min-version

**Syntax**

```
tls-min-version (v1.1 | v1.2);
```

**Hierarchy Level**

```
[edit access profile profile-name ldap-server ip-address]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D70.

**Description**

Configure Transport Layer Security (TLS) version to limit the lowest supported versions of TLS that are enabled for SSL connections.

**Options**

**v1.1**—Accept TLS version 1.1. This enhanced version of TLS provides protection against cipher-block chaining (CBC) attacks.

**v1.2** —Accept TLS version 1.2. This enhanced version of TLS provides improved flexibility for negotiation of cryptographic algorithms.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

Example: Configuring Integrated User Firewall on SRX Series | 197

# tls-peer-name

**Syntax**

```
tls-peer-name peer-host-name;
```

**Hierarchy Level**

```
[edit access profile profile-name ldap-server ip-address]
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D70.

**Description**

Configure the peer hostname to be authenticated.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION


- Firewall User Authentication Overview | 35
- Obtaining Username and Role Information Through Firewall Authentication | 172
- LDAP Functionality in Integrated User Firewall | 194

# tls-timeout

**Syntax**

```
tls-timeout seconds;
```

**Hierarchy Level**

```
[edit access profile profile-name ldap-server ip-address]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D70.

**Description**
Specify timeout value on the Transport Layer Security (TLS) handshake. The TLS handshake is responsible for the encryption keys exchange necessary to establish secure sessions between client and server.

**Range**: 3 through 90 seconds.

**Required Privilege Level**
services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION


- Firewall User Authentication Overview | 35
- Obtaining Username and Role Information Through Firewall Authentication | 172
- LDAP Functionality in Integrated User Firewall | 194

# tls-type

### Syntax

```
tls-type {
    start-tls;
}
```

### Hierarchy Level

```
[edit access profile profile-name ldap-server ip-address]
```

### Release Information

Statement introduced in Junos OS Release 15.1X49-D70.

### Description

Configure Lightweight Directory Access Protocol (LDAP) over Secure Sockets Layer/Transport Layer Security (SSL/TLS) for secure communication. Transport Layer Security StartTLS extension for LDAP is used for the firewall user authentication and the integrated user firewall authentication for obtaining username and role information through firewall authentication. StartTTLS allows protocol data transfers between the LDAP server and client over the TLS layer after successful negotiation between the peers. StartTLS upgrades an existing insecure LDAP connection to a secure Secure Sockets Layer/Transport Layer Security (SSL/TLS) connection.

### Options

- **start-tls**—Configure LDAP over StartTLS. The StartTLS communications occurs over TCP port 389.

### Required Privilege Level

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


Firewall User Authentication Overview | 35

Obtaining Username and Role Information Through Firewall Authentication | 172

LDAP Functionality in Integrated User Firewall | 194

# token-api (Services User Identification)

**Syntax**

```
token-api token-api
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Configure the token API that is used in generating the URL for acquiring an access token. The token API is combined with the connection method and the IP address of the ClearPass webserver to produce the complete URL used for acquiring an access token.

For example, if the token API is oauth, the connection method is HTTPS, and the IP address of the ClearPass webserver is 192.0.2.199, the complete URL for acquiring an access token would be https://192.0.2.199/api/oauth. This is a required parameter. There is no default value.

The SRX Series device user query function requires an access token to be able to query the ClearPass webserver. If the user query function is configured, the SRX Series device can request from the ClearPass webserver user authentication and identity information for an individual user.

**Required Privilege Level**
services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# token-api

**Syntax**

```
token-api token-api;
```

**Hierarchy Level**

```
[edit services user-identification identity-management connection]
```

**Release Information**
Statement introduced in Junos OS Release 15.1X49-D100.

**Description**
The path of the URL for acquiring the access token for OAuth2 authentication (RFC 6749). The Juniper Identity Management Service server requires that the SRX Series device authenticate to it using OAuth2. The SRX Series device uses the Client Credentials grant type for this purpose.

The following example shows the default tokenAPI, **oauth_token/oauth**, combined with the connection method, **https**, and the Juniper Identity Management Service server IP address placeholder to create the complete URL:

**https://JIMS/oauth_token/oauth.**

The advanced user identity query feature, to which this statement belongs, allows you to obtain user identity information from the Juniper Identity Management Service through queries. It allows you to provision users locally and have their authentication information made available to other sites in your network for policy enforcement and reporting.

**WARNING:** Before you use this feature, you must disable active-directory-access and authentication-source options under the user-identification hierarchy. You cannot commit this configuration if active directory authentication or the ClearPass query and webapi functions are configured and committed.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275

address | 405

authentication-entry-timeout | 420

batch-query | 426

ca-certificate | 433

client-id | 446

ip-query | 529

port | 569

primary | 575

query-api | 585

# to-zone (Security Policies)

**Syntax**

```
to-zone zone-name {
    policy policy-name {
        description description;
        match {
            application {
                [application];
                any;
            }
            destination-address {
                [address];
                any;
                any-ipv4;
                any-ipv6;
            }
            source-address {
                [address];
                any;
                any-ipv4;
                any-ipv6;
            }
            source-identity {
                [role-name];
                any;
                authenticated-user;
                unauthenticated-user;
                unknown-user;
            }
        }
        scheduler-name scheduler-name;
        then {
            count {
                alarm {
                    per-minute-threshold number;
                    per-second-threshold number;
                }
            }
            deny;
            log {
                session-close;
                session-init;
```

```
    }
```

```
permit {
    application-services {
        application-firewall {
            rule-set rule-set-name;
        }
        application-traffic-control {
            rule-set rule-set-name;
        }
        gprs-gtp-profile profile-name;
        gprs-sctp-profile profile-name;
        idp;
        redirect-wx | reverse-redirect-wx;
        ssl-proxy {
            profile-name profile-name;
        }
        uac-policy {
            captive-portal captive-portal;
        }
        utm-policy policy-name;
    }
    destination-address {
        drop-translated;
        drop-untranslated;
    }
    firewall-authentication {
        pass-through {
            access-profile profile-name;
            client-match user-or-group-name;
            ssl-termination-profile profile-name;
            web-redirect;
            web-redirect-to-https;
        }
        web-authentication {
            client-match user-or-group-name;
        }
    }
    services-offload;
    tcp-options {
        sequence-check-required;
        syn-check-required;
    }
    tunnel {
        ipsec-group-vpn group-vpn;
        ipsec-vpn vpn-name;
```

```
                pair-policy pair-policy;
            }
        }
        reject;
    }
  }
}
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name]
```

**Release Information**

Statement introduced in Junos OS Release 8.5. Support for the **services-offload** and **junos-host** options added in Junos OS Release 11.4. Support for the **source-identity** option added in Junos OS Release 12.1. Support for the **ssl-termination-profile** and **web-redirect-to-https** options added in Junos OS Release 12.1X44-D10.

**Description**

Specify a destination zone to be associated with the security policy.

**Options**

- **zone-name**—Name of the destination zone object.
- **junos-host**—Default security zone for self-traffic of the device.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Security Policies Overview*

*Understanding Security Policy Rules*

*Understanding Security Policy Elements*

# traceoptions (Access)

**Syntax**

```
traceoptions {
    file filename {
        files number;
        match regular-expression;
        size maximum-file-size;
        <world-readable | no-world-readable>;
    }
    flag flag;
}
```

**Hierarchy Level**

```
[edit access firewall-authentication]
```

**Release Information**

Statement introduced in Release 8.5 of Junos OS.

**Description**

Define Routing Engine firewall authentication tracing options.

**Options**

- **file *filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.
- **files *number***—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed to ***trace-file.0***, then ***trace-file.1***, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.
- If you specify a maximum number of files, you also must specify a maximum file size with the size option and a filename.

**Range:** 2 through 1000 files

**Default:** 10 files

- **match *regular-expression***—(Optional) Refine the output to include lines that contain the regular expression.
- **size*maximum-file-size***—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file.0***. When the **trace-file** again reaches its maximum size, ***trace-file.0*** is renamed ***trace-file.1*** and ***trace-file*** is renamed ***trace-file.0***. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

- If you specify a maximum file size, you also must specify a maximum number of trace files with the files option and filename.

**Syntax:** ***x*** **k** to specify KB, ***x*** **m** to specify MB, or ***x*** **g** to specify GB

**Range:** 10 KB through 1 GB

**Default:** 128 KB

- **world-readable | no-world-readable**—(Optional) By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.
- **flag *flag***—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags.
- **all**—All tracing operations
- **authentication**—Trace authentication events
- **configuration**—Trace configuration events
- **setup**—Trace setup of firewall authentication service

**Required Privilege Level**

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# traceoptions (Active Directory Access)

### Syntax

```
traceoptions {
    file filename ;
    flag {
        active-directory-authentication;
        all;
        configuration;
        db;
        ip-user-mapping;
        ip-user-probe;
        ipc;
        user-group-mapping;
        wmic;
    }
    level {
        all
        error
        info
        notice
        verbose
        warning
    }
    no-remote-trace;
}
```

### Hierarchy Level

```
[edit services user-identification active-directory-access]
```

### Release Information

Statement introduced in Junos OS Release 12.1X47-D10.

### Description

Define Active Directory trace options for the integrated user firewall feature.

### Options

**file *filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

**flag**—Trace the operation or operations to perform on the integrated user firewall. To specify more than one trace operation, include multiple flag statements.

**active-directory-authentication**—Trace the building of and modifications to the Active Directory authentication table.

**all**—Trace everything.

**configuration**—Trace configuration events.

**db**—Trace the database.

**ip-user-mapping**—Trace the ip-user-mapping module.

**ip-user-probe**—Trace PC client probing.

**ipc**—Trace communication events with the Packet Forwarding Engine.

**user-group-mapping**—Trace the process of getting user-to-group-mapping.

**wmic**—Trace the Windows Management Instrumentation Client process.

**level**—Level of trace operation to perform.

**all**—Match all levels.

**error**—Match error conditions.

**info**—Match informational messages.

**notice**—Match conditions that should be handled specially.

**verbose**—Match verbose messages.

**warning**—Match warning messages.

**no-remote-trace**—Disallow tracing from a remote device.

**Required Privilege Level**
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

active-directory-access | 400

user-identification (Services) | 665

Overview of Integrated User Firewall | 177

# traceoptions (Security Firewall Authentication)

**Syntax**

```
traceoptions {
    flag {
        all <detail | extensive | terse>;
        authentication <detail | extensive | terse>;
        proxy <detail | extensive | terse>;
    }
}
```

**Hierarchy Level**

```
[edit security firewall-authentication]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.

**Description**

Define data-plane firewall authentication tracing options.

**Options**

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple **flag** statements.
  - **all**—Enable all tracing operations
  - **authentication**—Trace data-plane firewall authentication events
  - **proxy**—Trace data-plane firewall authentication proxy events
- **detail**—Display moderate amount of data in trace.
- **extensive**—Display extensive amount of data in trace.
- **terse**—Display minimum amount of data in trace.

**Required Privilege Level**

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

RELATED DOCUMENTATION

# traceoptions (Services SSL)

**Syntax**

```
traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }
    flag flag;
    level [brief | detail | extensive | verbose];
    no-remote-trace;
    packet-filter {
        destination-ip;
        destination-port;
        source-ip;
        source-port;
    }
}
```

**Hierarchy Level**

```
[edit services ssl]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10. This statement is supported on the SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX. Junos OS Release 19.3R1 introduces **packet-filter** statement.

**Description**

Specify the trace file information.

Debug tracing on both Routing Engine and the Packet Forwarding Engine can be enabled for SSL proxy by using **[edit services ssl traceoptions]** command.

**Options**

- *file-name*—Specify the name of file in which to write trace information.
  - **files**—Specify the maximum number of trace files. Range: 2 to 1000.

- **match**—Specify the regular expression for lines to be logged. This statement is supported on the SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX.
- **no-world-readable size**—Do not allow any user to read the log file.
- **size**—Specify the maximum trace file size. Range: 10,240 to 1,073,741,824.
- **world-readable**—Allow any user to read the log file.

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple **flag** statements.
  - *all*—Trace all the parameters.
  - *cli-configuration*—Trace CLI configuration events.
  - *initiation*—Trace initiation service events.
  - *proxy*—Trace proxy service events.
  - *selected-profile*—Trace events for profiles with **enable-flow-tracing** set.
  - *termination*—Trace termination service events.
- **level**—Set the level of debugging the output option.
  - **brief**—Match brief messages.
  - **detail**—Match detail messages.
  - **extensive**—Match extensive messages.
  - **verbose**—Match verbose messages.
- **no-remote-trace**—Set remote tracing as disabled.
- **packet-filter**—Set packet filter to capture the traffic details.
  - **destination-ip** *ipvaddress*—Specify a destination IP address.

    Range—1 through 65535
  - **destination-port** *port-number*—Specify a destination port.
  - **source-ip** *ip-address*—Specify a source IP address.
  - **source-port** *port-number*—Specify a source IP port.

    Range—1 through 65535

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

Firewall User Authentication Overview | 35

# traceoptions (Services UAC)

### Syntax

```
traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }
    flag flag;
    no-remote-trace;
}
```

### Hierarchy Level

```
[edit services unified-access-control ]
```

### Release Information

Statement introduced in Junos OS Release 9.4.

### Description

Define Unified Access Control (UAC) tracing options.

Use this statement when you are configuring the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series UAC Appliance.

### Options

**flag**—Trace operation to perform. To specify more than one trace option, include multiple flag statements.

- **all**—Trace with all flags enabled
- **config**—Trace configuration information for all UAC-related configurations. This includes all configuration controlled through the **unified-access-control** statements at the **edit services** hierarchy level. It also includes other standard Junos OS configurations required for UAC enforcement such as zones, policies, and interfaces.
- **connect**—Trace communications between the Junos OS Enforcer and the IC Series appliance, including SSL handshakes and timeouts.
- **ipc**—Trace interprocess communications. Use this option to trace communications between the Routing Engine (RE) and the UACD enforcement plugin inside the Packet Forwarding Engine (PFE).

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

Understanding Unified Access Control | 151

Acquiring User Role Information from an Active Directory Authentication Server | 151

# traceoptions (Services User Identification)

**Syntax**

```
traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }
    flag flag;
    level level ;
    no-remote-trace;
}
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Specify the name of the trace log file and its characteristics. Messages about the behavior of the authentication source are written to this log file. Aruba ClearPass Policy Manager (CPPM) is the authentication source for the SRX Series device integrated ClearPass authentication and enforcement feature.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# trusted-ca (Services)

**Syntax**

```
trusted-ca (all | [ca-profile] );
```

**Hierarchy Level**

```
[edit services ssl proxy profile profile-name]
[edit services ssl termination profile profile-name]
[edit services ssl initiation profile profile-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10.

**Description**

Specify the list of trusted certificate authority profiles. This statement is supported on the SRX1500, SRX5400, SRX5600, and SRX5800 devices, and vSRX.

**Options**

- *trusted-ca-name*—Specify the certificate authority profile name.
- **all**—Select all certificate authority profiles.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

Firewall User Authentication Overview | 35

# uac-policy (Application Services)

### Syntax

```
uac-policy {
    captive-portal captive-portal;
}
```

### Hierarchy Level

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit application-services]
```

### Release Information

Statement modified in Junos OS Release 9.4.

### Description

Enable Unified Access Control (UAC) for the security policy. This statement is required when you are configuring the SRX Series device to act as a Junos OS Enforcer in a UAC deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series UAC Appliance .

### Options

The remaining statements are explained separately. See CLI Explorer.

### Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

> *Understanding User Role Firewalls*
>
> *Example: Configuring a User Role Firewall on an SRX Series Device*

# uac-service

### Syntax

```
uac-service {
    command binary-file-path;
    disable;
    failover (alternate-media | other-routing-engine);
}
```

### Hierarchy Level

```
[edit system processes]
```

### Release Information

Statement introduced in Junos OS Release 8.5.

### Description

Specify the unified access control daemon process.

### Options

- **command *binary-file-path***—Path to the binary process.
- **disable**—Disable the unified access control daemon process.
- **failover**—Configure the device to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.
  - **alternate-media**—Configure the device to switch to backup media that contains a version of the system if a software process fails repeatedly.
  - **other-routing-engine**—Instruct the secondary Routing Engine to take mastership if a software process fails. If this statement is configured for a process, and that process fails four times within 30 seconds, then the device reboots from the secondary Routing Engine.

### Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# unified-access-control (Security)

**Syntax**

```
unified-access-control priority priority;
```

**Hierarchy Level**

```
[edit security user-identification authentication-source]
```

**Release Information**

Statement introduced in Junos OS Release 12.1. Support for **disable** option dropped in Junos OS Release 12.1X47-D10.

**Description**

An authentication table pushed from a configured authentication device, such as the Junos Pulse Access Control Service.

**Options**

**priority *priority***—A unique value between 0 and 65535 that determines the sequence for searching multiple tables to retrieve a user role. Each authentication table is given a unique priority value. The lower the value, the higher the priority. A table with priority 120 is searched before a table with priority 200. The default priority value of the unified-access-control authentication table is 200.

Setting the priority value of the unified-access-control authentication table to 0 is equivalent to disabling the table and eliminating it from the search sequence.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

- *authentication-source (Security)*
- *Understanding User Role Firewalls*
- *Understanding the User Identification Table*

# unified-access-control (Services)

### Syntax

```
unified-access-control {
    captive-portal redirect-policy-name{
        redirect-traffic (all | unauthenticated);
        redirect-url redirect-url;
    }
    certificate-verification [ optional | required | warning ];
    infranet-contoller host-name {
        address ip-address;
        ca-profile [ca-profile];
        interface interface-name;
        password password;
        port port-number;
        server-certificate-subject subject;
    }
    interval seconds;
    test-only-mode;
    timeout seconds;
    timeout-action (close | no-change | open);
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            (no-world-readable | world-readable);
            size maximum-file-size;
        }
        flag flag;
        no-remote-trace;
    }
}
```

### Hierarchy Level

```
[edit services]
```

### Release Information

Statement introduced in Junos OS Release 9.4.

### Description

Use this statement to configure the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series device.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

# user-group-mapping

**Syntax**

```
user-group-mapping {
    ldap {
        address ip-address {
            port port;
        }
        authentication-algorithm {
            simple;
        }
        base base;
        ssl;
        user username {
            password password;
        }
    }
}
```

**Hierarchy Level**

```
[edit services user-identification active-directory-access domain]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X47-D10.

**Description**

Configure the SRX Series device to connect to an LDAP server, so that the server can provide the SRX Series with user-to-group mappings. These mappings are used to implement the integrated user firewall feature. The domain controller acts as the LDAP server in typical customer scenarios.

Most of this statement is optional, because the default communication method is LDAP and most arguments have default values. Only the LDAP keyword and the base are required.

**Options**

**ldap**—Required. LDAP is the protocol used to access the LDAP server to get user-to-group mappings.

**address *ip-address***—Optional. Specify the IP address of the LDAP server. If no address is specified, the system uses one of the configured Active Directory domain controllers.

**port *port***—Optional. Specify the port number of the LDAP server. If no port number is specified, the system uses port 389 for plaintext or port 636 for encrypted text.

**authentication-algorithm**—Optional. Specify the algorithm used while the SRX Series communicates with the LDAP server. The default method is Kerberos.

**simple**—Configure simple (plaintext) authentication method.

**base *base***—Required. LDAP base distinguished name (DN).

**ssl**—Optional. Enable Secure Sockets Layer (SSL) to ensure secure transmission with the LDAP server. Disabled by default, which means that the password is sent in plaintext.

**user *username***—Optional. Username of the LDAP account. If no username is specified, the system will use the configured domain controller's username.

**password *password***—Optional. Specify the password for the account. If no password is specified, the system uses the configured domain controller's password.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION


- active-directory-access | **400**
- clear services user-identification active-directory-access | **711**
- show services user-identification active-directory-access statistics | **795**
- show services user-identification active-directory-access user-group-mapping | **799**
- traceoptions (Active Directory Access) | **647**
- user-identification (Services) | **665**
- LDAP Functionality in Integrated User Firewall | **194**

# user-identification (Services)

**Syntax**

```
user-identification {
  active-directory-access {
    domain domain-name {
      user username;
        password password;
      domain-controller domain-controller-name {
        address domain-controller-address;
      }
      ip-user-mapping {
        discovery-method {
          wmi {
            event-log-scanning-interval seconds;
            initial-event-log-timespan hours;
          }
        }
      }
      user-group-mapping {
        ldap {
          address ip-address {
            port port;
          }
          authentication-algorithm {
            simple;
          }
          base base;
          ssl;
          user username {
            password password;
          }
        }
      }
    }
    authentication-entry-timeout minutes;
    filter {
      include address;
      exclude address;
    }
    no-on-demand-probe;
    wmi-timeout seconds;
    traceoptions {
```

```
    file file;
    flag {
        active-directory-authentication;
        all;
        configuration;
        db;
        ip-user-mapping;
        ip-user-probe;
        ipc;
        user-group-mapping;
        wmic;
    }
    level {
        all;
        error;
        info;
        notice;
        verbose;
        warning;
    }
    no-remote-trace;
}
```

```
logical-domain-identity-management {
    active {
        authentication-entry-timeout minutes;
        filter {
            domain name;
            exclude-ip {
                address-book book-name;
                address-set address-set;
            }
            include-ip {
                address-book book-name;
                address-set address-set;
            }
        }
        invalid-authentication-entry-timeout minutes;
        ip-query {
            query-delay-time seconds;
        }
        query-server name {
            batch-query {
                items-per-batch items-per-batch;
                query-interval seconds;
            }
            connection {
                connect-method (http | https);
                port port;
                primary {
                    address address;
                    ca-certificate ca-certificate;
                    client-id client-id;
                    client-secret client-secret;
                }
                query-api query-api;
                secondary {
                    address address;
                    ca-certificate ca-certificate;
                    client-id client-id;
                    client-secret client-secret;
                }
                token-api token-api;
            }
        }
    }
    traceoptions {
```

```
            file <filename> <files files> <match match> <size size> <(world-readable | no-world-readable)>;
            flag name;
            level (all | error | info | notice | verbose | warning);
            no-remote-trace;
        }
    }
}
```

**Hierarchy Level**

```
[edit services]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X47-D10.
**logical-domain-identity-management** option introduced in Junos OS Release 19.3R1.

**Description**

Configure the integrated user firewall feature, including access to the Active Directory domain and domain controller, IP address-to-user mapping, and user-to-group mapping. One or two Active Directories are allowed under one domain. The IP address-to-user mapping and user-to-group mapping are configured per domain.

**Options**

**authentication-entry-timeout *minutes***—Timeout interval starting from the Active Directory/domain controller login time, the last active session, or the last successful probe. A setting of 0 means the authentication does not need a timeout. We recommend that you configure a setting of 0 when you disable on-demand-probe to prevent someone from accessing the Internet without logging in again.

**Range:** 10 through 1440 minutes

**Default:** 30 minutes

**filter**—Optional. Range of IP addresses that needs to be monitored or not monitored.

**include *address***—Include IP address or range. Maximum of 20 addresses.

**exclude *address***—Exclude IP address or range. Maximum of 20 addresses.

**no-on-demand-probe**—Do not use traffic to discover user. Default is disabled.

**wmi-timeout *seconds***—(Optional) Configures the number of seconds that the domain PC has to respond to the SRX Series device's query through WMI/DCOM.

- If the PC responds within that timeframe to the WMI query, the SRX creates an authentication entry for this PC.
- If the PC does not respond within that timeframe, the WMI query failed. In the case of a failed query, if the SRX had an authentication entry about the queried PC before the WMI query, that authentication entry is deleted. If the SRX had no authentication entry before the WMI query, the SRX does not create an authentication entry.

**Range:** 3 through 120 seconds

**Default:** 10 seconds

**logical-domain-identity-management**—Configures the logical domain identity management.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

active-directory-access | 400
traceoptions (Active Directory Access) | 647

# webapi (System Services)

### Syntax

```
webapi {
client ip-address;
(
    http {
        port  port-number;
    }
    https {
        certificate  certificate-filename;
        certificate-key  local-certificate-key;
        default-certificate
        pki-local-certificate;
        port port-number;
    }
    user {
        name;
        password password;
    }
    debug-log filename;
    debug-level level;
}
```

### Hierarchy Level

```
[edit system services]
```

### Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

### Description

Configure the Web API function daemon (webapi) component of the integrated ClearPass authentication and enforcement feature. The Web API daemon acts as a HTTP or HTTPS server. The SRX Series device exposes to the Aruba ClearPass Policy Manager (CPPM) the Web API that allows the CPPM, as a client, to send POST request messages to it that provide the SRX Series device with user authentication and identity information. The CPPM serves as the user authentication source for the SRX Series device.

The Web API function (webapi) facilitates efficient transmission of user authentication and identity information from the CPPM to the SRX Series device. The CPPM, which is the client in this relationship, initiates a session with the SRX Series device Web API daemon, which is the server in this relationship.

However, the CPPM can do this only if you have configured the Web API function on the SRX Series device. For security reasons, the Web API daemon is not enabled by default.

The configuring statements are explained separately. See CLI Explorer.

**Required Privilege Level**
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

# webapi-clear-text (Security)

**Syntax**

```
web-api-cleartext
```

**Hierarchy Level**

```
[edit security zones security-zone zone host-inbound-traffic system-services]
```

**Release Information**
Statement introduced in Junos OS Release 12.3X48-D30.

**Description**
Enable the Web API (webapi) service over HTTP host inbound traffic on TCP port 8080 for unencrypted data.

**Required Privilege Level**
security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

# webapi-ssl (Security)

**Syntax**

```
webapi-ssl
```

**Hierarchy Level**

```
[edit security zones security-zone zone host-inbound-traffic system-services]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Enable the Web API service over HTTPS host inbound traffic on TCP port 8443.

**Required Privilege Level**

**security**—To view this statement in the configuration.

**security-control**—To add this statement to the configuration.

# web-authentication

**Syntax**

```
web-authentication {
    client-match user-or-group-name;
}
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.

HTTPS for Web authentication is supported on SRX5400, SRX5600, and SRX5800 devices starting from Junos OS Release 12.1X44-D10 and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.

**Description**

Specify that the policy allows access to users who have previously been authenticated by Web authentication. Web authentication must be enabled on one of the addresses on the interface to which the HTTP or HTTPS request is redirected.

**Options**

**client-match** ***user-or-group*** —(Optional) Username or user group name.

**Required Privilege Level**

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Understanding User Role Firewalls*

# web-authentication (Access)

**Syntax**

```
web-authentication {
    banner {
        success string;
    }
    default-profile profile-name;
    timeout seconds;
}
```

**Hierarchy Level**

```
[edit access firewall-authentication]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.
HTTPS for Web authentication is supported on SRX5400, SRX5600, and SRX5800 devices starting from Junos OS Release 12.1X44-D10 and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.
Option **timeout** introduced in Junos OS Release 15.1X49-D130.

**Description**

Specify that users go through the Web authentication process. The user uses HTTP or HTTPS to access an IP address on the device that is enabled for Web authentication. In this scenario, the user does not use HTTP or HTTPS to access the IP address of the protected resource. The user is prompted for a username and password, which are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the results of this authentication. This method of authentication differs from pass-through authentication in that users need to access the protected resource directly after accessing the Web authentication IP address and being authenticated.

**Options**

**timeout *seconds***—Specify the **timeout** option in seconds. If you do not specify a timeout value, and if the web authentication process takes more than 3 seconds, your browser may display **invalid username and password**, even though the username and password is correct. For example, when you type a username and password in a browser for authentication, SRX Series device checks your account in the database, and after 3 seconds your web browser displays a message **invalid username and password**. However, after 10 seconds, SRX Series device receives a response from the database that the user authentication is successful, but SRX Series device could not notify you about successful authentication, due to 3 seconds timeout value. If you configure the timeout value from 5 through 60 seconds, then the browser waits for the SRX Series device to respond for the specified time.

**Default:** 3 seconds

**Range:** 5 through 60 seconds

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

# web-authentication (Interfaces)

**Syntax**

```
web-authentication {
    http;
    https;
    redirect-to-https;
}
```

**Hierarchy Level**

**[edit interfaces** ***interface-name*** **unit** ***logical-unit-number*** **family** ***family-name*** **address** ***address*** **]**

**Release Information**

Statement introduced in Junos OS Release 9.2.

Support for **https** and **redirect-to-https** introduced for SRX5400, SRX5600, and SRX5800 Services Gateways starting from Junos OS Release 12.1X44-D10 and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.

**Description**

Enable the Web authentication process for firewall user authentication.

**Options**

**http**—Enable HTTP service.

**https**—Enable authentication through HTTPS.

**redirect-to-https**—Redirect Web authentication to HTTPS.

**Required Privilege Level**

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Understanding Interfaces*

# web-management (System Services)

### Syntax

```
web-management {
    http {
        interfaces interface-names ;
        port port;
    }
    https {
        interfaces interface-names;
        local-certificate name;
        pki-local-certificate name;
        system-generated-certificate name;
        port port;
    }
    management url management url;
    session {
        idle-timeout minutes;
        session-limit number;
    }
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            size maximum-file-size;
            (no-world-readable | world-readable);
        }
        flag flag;
        level level;
        no-remote-trace;
    }
}
```

### Hierarchy Level

```
[edit system services]
```

### Release Information

Statement introduced in Junos OS Release 9.0.

Support for **https** introduced for SRX5400, SRX5600, and SRX5800 devices starting from Junos OS Release 12.1X44-D10 and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices starting from Junos OS Release 15.1X49-D40.

**Description**

Configure settings for HTTP or HTTPS access. HTTP access allows management of the device using the J-Web interface. HTTPS access allows secure management of the device using the J-Web interface. With HTTPS access, communication is encrypted between your browser and the webserver for your device.

> **NOTE:** On SRX340 and SRX345 devices, the factory-default configuration has a generic HTTP configuration. To use ge and fxp0 ports as management ports, you must use the set system services web-management http command. The Web management HTTP and HTTPS interfaces are changed to fxp0.0 and from ge-0/0/1.0 through ge-0/0/7.0.

**Options**

**control**—Disable the SBC process.

- **max-threads**—Maximum simultaneous threads to handle requests.

  **Range:** 0 through 16

**http**—Configure HTTP.

- **interface [*value*]**—Interface value that accepts HTTP access.
- **port *number***—TCP port for incoming HTTP connections.

  **Range:** 1 through 65,535

**https**—Configure HTTPS.

- **interface [*value*]**—Interface value that accept HTTP access.
- **port *number***—TCP port for incoming HTTP connections.

  **Range:** 1 through 65,535
- **local-certificate**—X.509 certificate to use from the configuration.
- **pki-local-certificate**—X.509 certificate to use from the PKI local store.
- **system-generated-certificate**—X.509 certificate generated automatically by the system.

**management url *management url***—URL path for Web management access.

**session**—Configure the Web-management session.

- **idle-timout *minutes***—Default timeout of Web-management sessions in minutes.
- **session-limit *number***—Maximum number of Web-management sessions to allow.

**traceoptions**—Set the trace options.

- **file**—Configure the trace file information.
  - *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. All files are placed in the directory **/var/log**. By default, the name of the file is the name of the process being traced.
  - **files *number***— Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

    If you specify a maximum number of files, you also must specify a maximum file size with the **size *maximum file-size*** option.

    **Range:** 2 through 1000 files

**Default:** 10 files

- **match *regular-expression***—Refine the output to include lines that contain the regular expression.
- **size *maximum-file-size***—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB).

  **Range:** 10 KB through 1 GB

  **Default:** 128 KB

  If you specify a maximum file size, you also must specify a maximum number of trace files with the **files *number*** option.
- **(world-readable | no-world-readable)**— By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.
- **flag *flag***—Specify which tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can include the following flags.
  - **all**—Trace all areas.
  - **configuration**—Trace configuration.
  - **dynamic-vpn**—Trace dynamic VPN events.
  - **init**—Trace the daemon init process.
  - **mgd**—Trace MGD requests.
  - **webauth**—Trace Web authentication requests.
- **level *level*** —Specify the level of debugging output.
  - **all**—Match all levels.
  - **error**—Match error conditions.
  - **info**—Match informational messages.
  - **notice**—Match conditions that should be handled specially.
  - **verbose**—Match verbose messages.
  - **warning**—Match warning messages.
- **no-remote-trace**—Disable remote tracing.

**Required Privilege Level**
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

*Dynamic VPN Overview*

# web-redirect

**Syntax**

```
web-redirect;
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name  policy policy-name then permit firewall-authentication
    pass-through user-firewall]
```

**Release Information**

Statement introduced in Junos OS Release 8.5.

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, support for user-firewall added on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Services Gateways.

**Description**

Optionally, redirect HTTP requests to the device's internal webserver by sending a redirect HTTP response to the client system to reconnect to the webserver for user authentication. The interface on which the client's request arrived is the interface to which the request is redirected.

**Required Privilege Level**

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

*Understanding User Role Firewalls*

# web-redirect-to-https

**Syntax**

```
web-redirect-to-https;
```

**Hierarchy Level**

```
[edit security policies from-zone zone-name to-zone zone-name policy policy-name then permit firewall-authentication
    pass-through user-firewall]
```

**Release Information**

Statement introduced on on SRX5600 and SRX5800 Services Gateways starting from Junos OS Release 12.1X44-D10, and on vSRX, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Services Gateways starting from Junos OS Release 15.1X49-D40.

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, support for **user-firewall** added on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Services Gateways.

**Description**

Redirect unauthenticated HTTP requests to the internal HTTPS webserver of the device.

**Required Privilege Level**

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Unified Threat Management User Guide*

Firewall User Authentication Overview | 35

# web-server (Services)

**Syntax**

```
web-server server-name;
```

**Hierarchy Level**

```
[edit services user-identification authentication-source aruba-clearpass user-query]
```

**Release Information**

Statement introduced in Junos OS Release 12.3X48-D30.

**Description**

Specify the name of the webserver configuration on the SRX Series device used for the user query integrated ClearPass authentication and enforcement function. The webserver is the ClearPass server to which the SRX Series device connects to request authentication and identity information for an individual user.

When information for the individual user is not posted to the SRX Series device by ClearPass through Web API POST request messages, the SRX Series device can request this information from the ClearPass Policy Manager (CPPM) under certain circumstances. You must enable the user query function by configuring it.

**Required Privilege Level**

services—To view this statement in the configuration.

services-control—To add this statement to the configuration.

# whitelist (Services)

**Syntax**

```
whitelist [global-address-book-addresses];
```

**Hierarchy Level**

```
[edit services ssl proxy profile profile-name]
[edit services ssl termination profile profile-name]
```

**Release Information**

Statement introduced in Junos OS Release 12.1X44-D10.

**Description**

Specify the addresses exempted from the SSL proxy. This statement is supported on the SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX.

You can selectively bypass SSL proxy processing for some sessions by configuring a whitelist. Typically, you might configure the whitelist to include trusted servers or domains with which you are very familiar. Whitelists include addresses that you want to exempt from undergoing SSL proxy processing.

To configure the whiltelist, you need to specify the domain that you want to exempt in an address book and then configure the address in the SSL proxy profile.

**Options**

- *whitelist-address*—Specify address from the global address book.

**Required Privilege Level**

services—To view this statement in the configuration.
services-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring SSL Forward Proxy | 99

Firewall User Authentication Overview | 35

# wins-server (Access)

**Syntax**

```
wins-server address
```

**Hierarchy Level**

```
[edit access address-assignment pool <name> family (inet | inet6) xauth-attributes]
[edit access profile profile-name]
```

**Release Information**

Statement introduced in Release 10.4 of Junos OS. Starting with Junos OS 15.1X49-D80 and Junos OS Release 17.3R1, the wins-server option at the [edit access profile] hierarchy level allows you to configure the IPv4 address of a Windows Internet Name Service (WINS) server.

**Description**

Specify the wins-server IP address.

**Required Privilege Level**

access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

## RELATED DOCUMENTATION

*Junos OS Security Configuration Guide*

8

CHAPTER

# Operational Commands

clear network-access requests pending | 691

clear network-access requests statistics | 693

clear network-access securid-node-secret-file | 694

clear security firewall-authentication history | 695

clear security firewall-authentication history address | 698

clear security firewall-authentication history identifier | 700

clear security firewall-authentication users | 702

clear security firewall-authentication users address | 705

clear security firewall-authentication users identifier | 707

clear security user-identification local-authentication-table | 709

clear service user-identification identity-management counter | 710

clear services user-identification active-directory-access | 711

clear services user-identification authentication-table | 712

request security user-identification local-authorization-table add | 714

request services user-identification active-directory-access active-directory-authentication-table delete | **716**

request services user-identification active-directory-access domain-controller | **718**

request services user-identification active-directory-access ip-user-probe | **720**

request services user-identification authentication-source aruba-clearpass user-query | **722**

request services user-identification authentication-table delete | **724**

show network-access requests pending | **733**

show network-access requests statistics | **736**

show network-access securid-node-secret-file | **738**

show security firewall-authentication history | **740**

show security firewall-authentication history address | **743**

show security firewall-authentication history identifier | **746**

show security firewall-authentication jims statistics | **749**

show security firewall-authentication users | **751**

show security firewall-authentication users address | **755**

show security firewall-authentication users identifier | **759**

show security user-identification local-authentication-table | **763**

show security policies | **766**

show services unified-access-control counters | **783**

show services unified-access-control policies | **786**

show services unified-access-control roles | **789**

show services unified-access-control status | **791**

show services user-identification active-directory-access domain-controller status | **792**

show services user-identification active-directory-access statistics | **795**

show services user-identification active-directory-access user-group-mapping | **799**

show service user-identification authentication-source aruba-clearpass user-query counters | **802**

show service user-identification authentication-source aruba-clearpass user-query status | **804**

show services user-identification authentication-table | 805

show service user-identification identity-management | 826

show services user-identification device-information table | 829

# clear network-access requests pending

**Syntax**

```
clear network-access requests pending
<index index-number >
```

**Release Information**

Command introduced in Release 8.5 of Junos OS.

**Description**

Clear or cancel all pending authentication requests.

**Options**

- none—Clear all network access requests pending.
- **index** ***index-number*** —Clear the specified authentication request. To display index numbers, use the **show network-access requests pending** command.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

show network-access requests pending | 733

**List of Sample Output**

clear network-access requests pending on page 691

## Sample Output

The following example displays the network access requests that are pending, clears the requests, and displays the results of the clear operation:

**clear network-access requests pending**

```
user@host> show network-access requests pending
```

```
Information about pending authentication entries
  Total pending authentication requests: 2
```

```
Index User                  Status
1     Sun                   Processing
2     Sam                   Processed
```

user@host> **clear network-access requests pending**

user@host> **show network-access requests pending**

```
Information about pending authentication entries
  Total pending authentication requests: 2
Index User                  Status
1     Sun                   Cancelled by Admin
2     Sam                   Cancelled by Admin
```

# clear network-access requests statistics

**Syntax**

```
clear network-access requests statistics
```

**Release Information**

Command introduced in Release 8.5 of Junos OS.

**Description**

Clear general authentication statistics for the configured authentication type.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

*authentication-order (Access Profile)*

show network-access requests statistics | 736

**Output Fields**

This command produces no output.

# clear network-access securid-node-secret-file

**Syntax**

```
clear network-access securid-node-secret-file
```

**Release Information**

Command introduced in Junos OS Release 9.1.

**Description**

Delete the node secret file for the SecurID authentication type.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

- Firewall User Authentication Overview | 35
- configuration-file | 459
- securid-server | 606
- show network-access securid-node-secret-file | 738

**List of Sample Output**

clear network-access securid-node-secret-file on page 694

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**clear network-access securid-node-secret-file**

```
user@host> clear network-access securid-node-secret-file
```

# clear security firewall-authentication history

**Syntax**

```
clear security firewall-authentication history
<node (node-id | all | local | primary)>
<address>
<identifier>
<logical-system (logical-system-name | all)>
<root-logical-system (address | auth-type | from-zone | identifier | tenant | to-zone>
<tenant (tenant-name |all)>
```

**Release Information**

Command introduced in Junos OSRelease 8.5. The **node** options added in Junos OS Release 9.0. The **tenant** option introduced in Junos OS Release 18.3R1.

**Description**

Clear all firewall authentication history information.

**Options**

- **node**—(Optional) For chassis cluster configurations, clear all firewall authentication history on a specific node (device) in the cluster.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all** —Clear all nodes.
  - **local** —Clear the local node.
  - **primary**—Clear the primary node.
- address—Display authentication entries based on ip address.
- identifier—Display authentication entries by id.
- logical-system—Display firewall authentication tables based on logical system name.
- **node**—(Optional) For chassis cluster configurations, display firewall authentication details for all users on a specific node.
  - *node-id*—Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.
- root-logical-system—Display firewall authentication tables for root logical system.
- tenant—Display firewall authentication tables based on tenant name.

**Required Privilege Level**
clear

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35
show security firewall-authentication history | 740

**List of Sample Output**
clear security firewall-authentication history on page 696
clear security firewall-authentication history node 1 on page 696
clear security firewall-authentication history tenant all on page 697

**Output Fields**
When you enter this command, you are provided feedback on the status of your request.

This command produces no output.

## Sample Output

**clear security firewall-authentication history**

```
user@host> clear security firewall-authentication history

node0:
--------------------------------------------------------------------------
node1:
--------------------------------------------------------------------------
```

## Sample Output

**clear security firewall-authentication history node 1**

```
user@host> clear security firewall-authentication history node 1

node1:
--------------------------------------------------------------------------
```

**clear security firewall-authentication history tenant all**

```
user@host> clear security firewall-authentication history tenant all
```

# clear security firewall-authentication history address

**Syntax**

```
clear security firewall-authentication history address  address
<node ( node-id  | all | local | primary)>
```

**Release Information**

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

**Description**

Clear firewall authentication history for this source IP address.

**Options**

- **address** ***address*** —Source IP address for which to clear firewall authentication history.
- none—Clear all firewall authentication history for this address.
- **node**—(Optional) For chassis cluster configurations, clear firewall authentication history for this address on a specific node.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all** —Clear all nodes.
  - **local** —Clear the local node.
  - **primary**—Clear the primary node.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

clear security firewall-authentication history address 10.0.0.1 on page 699
clear security firewall-authentication history address 192.0.2.2 node 1 on page 699

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**clear security firewall-authentication history address 10.0.0.1**

```
user@host> clear security firewall-authentication history address 10.0.0.1

node0:
--------------------------------------------------------------------------
node1:
--------------------------------------------------------------------------
```

## Sample Output

**clear security firewall-authentication history address 192.0.2.2 node 1**

```
user@host> clear security firewall-authentication history address 192.0.2.2 node 1

node1:
--------------------------------------------------------------------------
```

# clear security firewall-authentication history identifier

**Syntax**

```
clear security firewall-authentication history identifier  identifier
<node ( node-id  | all | local | primary)>
```

**Release Information**

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

**Description**

Clear firewall authentication history information for the authentication with this identifier.

**Options**

- **identifier** ***identifier*** —Identification number of the authentication for which to clear authentication history.
- none—Clear all firewall authentication history information for the authentication with this identifier.
- **node**—(Optional) For chassis cluster configurations, clear firewall authentication history on a specific node for the authentication with this identifier.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all** —Clear all nodes.
  - **local** —Clear the local node.
  - **primary**—Clear the primary node.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

clear security firewall-authentication history identifier 2 on page 701
clear security firewall-authentication history identifier 2 node 1 on page 701

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**clear security firewall-authentication history identifier 2**

```
user@host> clear security firewall-authentication history identifier 2
```

```
node0:
--------------------------------------------------------------------------
node1:
--------------------------------------------------------------------------
```

## Sample Output

**clear security firewall-authentication history identifier 2 node 1**

```
user@host> clear security firewall-authentication history identifier 2 node 1
```

```
node1:
--------------------------------------------------------------------------
```

# clear security firewall-authentication users

**Syntax**

```
clear security firewall-authentication users
<node (node-id | all | local | primary)>
<address>
<identifier>
<logical-system (logical-system-name | all)>
<root-logical-system (address | auth-type | from-zone | identifier | tenant | to-zone>
<tenant (tenant-name |all)>
```

**Release Information**

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0. The **tenant** option introduced in Junos OS Release 18.3R1.

**Description**

Clear firewall authentication tables for all users.

**Options**

- **node**—(Optional) For chassis cluster configurations, clear firewall authentication details for all users on a specific node.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all** —Clear all nodes.
  - **local** —Clear the local node.
  - **primary**—Clear the primary node.
- address—Display authentication entries based on ip address.
- identifier—Display authentication entries by id.
- logical-system—Display firewall authentication tables based on logical system name.
- **node**—(Optional) For chassis cluster configurations, display firewall authentication details for all users on a specific node.
  - *node-id*—Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.
- root-logical-system—Display firewall authentication tables for root logical system.
- tenant—Display firewall authentication tables based on tenant name.

**Required Privilege Level**
clear

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35
show security firewall-authentication users | 751

**List of Sample Output**
clear security firewall-authentication users on page 703
clear security firewall-authentication users node 1 on page 703
clear security firewall-authentication users tenant all on page 704

**Output Fields**
When you enter this command, you are provided feedback on the status of your request.

This command produces no output.

## Sample Output

**clear security firewall-authentication users**

```
user@host> clear security firewall-authentication users node 1
```

```
node0:
--------------------------------------------------------------------------
node1:
--------------------------------------------------------------------------
```

## Sample Output

**clear security firewall-authentication users node 1**

```
user@host> clear security firewall-authentication users node 1
```

```
node1:
--------------------------------------------------------------------------
```

**clear security firewall-authentication users tenant all**

```
user@host> clear security firewall-authentication users tenant all
```

# clear security firewall-authentication users address

**Syntax**

```
clear security firewall-authentication users address  address
<node ( node-id  | all | local | primary)>
```

**Release Information**

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

**Description**

Clear information about the users at the specified IP address that are currently authenticated.

**Options**

- **address** ***address*** —IP address for which to clear user firewall authentication information.
- none—Clear all the firewall authentication information for users at this IP address.
- **node**—(Optional) For chassis cluster configurations, clear user firewall authentication entries on a specific node.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all** —Clear all nodes.
  - **local** —Clear the local node.
  - **primary**—Clear the primary node.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

clear security firewall-authentication users address 198.51.100.11 on page 706
clear security firewall-authentication users address 198.51.100.11 node 1 on page 706

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**clear security firewall-authentication users address 198.51.100.11**

```
user@host> clear security firewall-authentication users address 198.51.100.11
```

```
node0:
--------------------------------------------------------------------------
node1:
--------------------------------------------------------------------------
```

## Sample Output

**clear security firewall-authentication users address 198.51.100.11 node 1**

```
user@host> clear security firewall-authentication users address 198.51.100.11 node 1
```

```
node1:
--------------------------------------------------------------------------
```

# clear security firewall-authentication users identifier

**Syntax**

```
clear security firewall-authentication users identifier identifier
<node ( node-id | all | local | primary)>
```

**Release Information**

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

**Description**

Clear firewall authentication details about the user with this identification number.

**Options**

- none—Identification number of the user for which to clear authentication details.
- **node**—(Optional) For chassis cluster configurations, clear the firewall authentication details on a specific node (device) in the cluster for the user with this identification number.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all** —Clear all nodes.
  - **local** —Clear the local node.
  - **primary**—Clear the primary node.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

clear security firewall-authentication users identifier 2 on page 708
clear security firewall-authentication users identifier 2 node 1 on page 708

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**clear security firewall-authentication users identifier 2**

```
user@host> clear security firewall-authentication users identifier 2
```

```
node0:
--------------------------------------------------------------------------
node1:
--------------------------------------------------------------------------
```

## Sample Output

**clear security firewall-authentication users identifier 2 node 1**

```
user@host> clear security firewall-authentication users identifier 2 node 1
```

```
node1:
--------------------------------------------------------------------------
```

# clear security user-identification local-authentication-table

**Syntax**

```
clear security user-identification local-authentication-table
```

**Release Information**

Command introduced in Junos OS Release 12.1.

**Description**

This command removes all entries from the local authentication table.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

> *Understanding Application Security*
>
> Firewall User Authentication Overview | 35

**List of Sample Output**

clear security user-identification local-authentication-table on page 709

**Output Fields**

When you enter this command, all entries are cleared from the local authentication table.

## Sample Output

**clear security user-identification local-authentication-table**

`user@host>` **clear security user-identification local-authentication-table**

`user@host>` **show security user-identification local-authentication-table all**

```
Total entries: 0
```

# clear service user-identification identity-management counter

**Syntax**

```
clear service user-identification identity-management counter
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D100.

**Description**

Clear the counters associated with the batch queries and IP queries for the advanced user query feature.

**Options**

This command has no options.

**Required Privilege Level**

clear

RELATED DOCUMENTATION

# clear services user-identification active-directory-access

**Syntax**

```
clear services user-identification active-directory-access (active-directory-authentication-table |
statistics (ip-user-mapping | ip-user-probe | user-group-mapping))
```

**Release Information**

Command introduced in Junos OS Release 12.1X47-D10.

**Description**

Delete entries from the Active Directory authentication table or statistics related to integrated user firewall mappings.

**Options**

- **active-directory-authentication-table**—Remove all entries from the Active Directory authentication table.
- **statistics**—Remove the specified type of statistics:
  - **ip-user-mapping**—IP address-to-user mappings
  - **ip-user-probe**—PC probe statistics
  - **user-group-mapping**—User-to-group mappings

**Required Privilege Level**

clear

RELATED DOCUMENTATION


ip-user-mapping | 531

request services user-identification active-directory-access ip-user-probe | 720

show services user-identification active-directory-access statistics | 795

show services user-identification active-directory-access user-group-mapping | 799

user-group-mapping | 663

user-identification (Services) | 665


**Output Fields**

This command produces no output.

# clear services user-identification authentication-table

**Syntax**

clear services user-identification authentication-table authentication-source *authentication-source* (all | active-directory | aruba-clearpass | identity-management)

**Release Information**

Command introduced in Junos OS Release 12.1X47-D10 for active directory as an authentication source.
Support added for Aruba ClearPass as an authentication source in Junos OS release 12.3X48-D30.
Support added for identity-management as an authentication source in Junos OS Release 15.1X49-D100.

**Description**

Clears the user identity and authentication entries content of the specified authentication source's authentication table.

**Options**

***authentication-source***—Active Directoy, Aruba ClearPass, or the identity management server, which could be the Juniper Identity Management Service (JIMS) or any third-party authentication source.

**Additional Information**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

clear

**List of Sample Output**

**clear services user-identification authentication-table authentication-source on page 713**
**clear services user-identification authentication-table authentication-source identity-management on page 713**

**Output Fields**

For Aruba ClearPass, if there are no entries in the ClearPass authentication table, the following warning message is displayed after you enter the clear command.

**There is no authentication-table entry.**

If there are entries in the ClearPass authentication table, no messages are displayed after you enter the **clear** command.

## Sample Output

**clear services user-identification authentication-table authentication-source**

user@host> **clear services user-identification authentication-table authentication-source aruba-clearpass**

```
warning: "There is no authentication-table entry."
```

**clear services user-identification authentication-table authentication-source identity-management**

user@host> **clear services user-identification authentication-table authentication-source identity-management**

```
warning: "There is no authentication-table entry."
```

# request security user-identification local-authorization-table add

**Syntax**

request security user-identification local-authorization-table add user *user-name* ip-address *ip-address* roles [*role-name*]

**Release Information**

Command introduced in Junos OS Release 12.1. Command updated in Junos OS Release 12.1X44-D10.

**Description**

This command adds user and role information to the local authentication table. The table is used to retrieve user and role information for traffic from the specified IP address to enforce a user role firewall.

To add an entry, specify the user name, IP address, and up to 40 roles to be associated with this user. Subsequent commands for the same user and IP address aggregates any new roles to the existing list. An authentication entry can contain up to 200 roles.

**NOTE:** To change the user name of an entry or to remove or change entries in a role list, you must delete the existing entry and create a new one.

An IP address can be associated with only one user. If a second request is made to add a different user using the same IP address, the second authentication entry overwrites the existing entry.

**Options**

user *user-name*—Specify the name of the user to be added to the table.

ip-address *ip-address*—Specify the IP address of the user. Either IPv4 or IPv6 addresses are supported.

roles [*role-name*]—(Optional) Specify the role or list of roles to be associated with the specified user. If the specified user and IP address already exist, any roles specified in the command are added to the existing role list.

**Required Privilege Level**

maintenance

## RELATED DOCUMENTATION

*request security user-identification local-authentication-table delete*

*Understanding the User Identification Table*

**List of Sample Output**

request security user-identification local-authentication-table add on page 715

**Output Fields**

When you enter this command, either an entry is added to the local authentication table, or the roles of an existing entry are aggregated with additional roles.

## Sample Output

**request security user-identification local-authentication-table add**

`user@host>` **request security user-identification local-authentication-table add user user1 ip-address 192.0.2.1 roles role1**

`user@host>` **request security user-identification local-authentication-table add user user2 ip-address 203.0.113.2 roles [role2 role3]**

`user@host>` **request security user-identification local-authentication-table add user user2 ip-address 203.0.113.2 roles role1**

`user@host>` **show security user-identification local-authentication-table all**

```
Total entries: 2
Source IP       Username      Roles
192.0.2.1       user1      role1
203.0.113.2     user2      role2, role3, role1
```

# request services user-identification active-directory-access active-directory-authentication-table delete

**Syntax**

```
request services user-identification active-directory-access active-directory-authentication-table delete
(domain name | ip-address ip-address | group group-name <domain name> | user name <domain name>
```

**Release Information**

Command introduced in Junos OS Release 12.1X47-D10.

**Description**

Delete entries from the active directory authentication table by domain, address, group, or user. This command provides the network administrator with flexibility and control over the table entries beyond what is automatically added to or deleted from the table. For example, if a person leaves the company, the corresponding username can be deleted; after a department reorganization, a group can be deleted.

**Options**

- **domain *name***—Delete the entries from the authentication table for the specified domain.
- **ip-address *ip-address***—Delete the entry from the authentication table for the specified IP address.
- **group *group-name***—Delete the entries from the authentication table for the specified group.
  - **domain *name***—Delete the group only from the specified domain.
- **user *name***—Delete the entries from the authentication table for the specified username.
  - **domain *name***—Delete the user only from the specified domain.

**Required Privilege Level**

maintenance

RELATED DOCUMENTATION

*show services user-identification active-directory-access active-directory-authentication-table*

user-identification (Services) | 665

Understanding Active Directory Authentication Tables | 182

**Output Fields**

This command produces no output.

# request services user-identification active-directory-access domain-controller

**Syntax**

```
request services user-identification active-directory-access domain-controller discovery
domain name
```

**Release Information**

Command introduced in Junos OS Release 12.1X47-D10.

**Description**

Discover and display the name and address of all domain controllers in the specified domain.

**Options**

- **domain *name***—Name of the domain for which to get and display domain controller names and addresses.

**Required Privilege Level**

maintenance

RELATED DOCUMENTATION


active-directory-access | 400

show services user-identification active-directory-access domain-controller status | 792

user-identification (Services) | 665


**List of Sample Output**

**request services user-identification active-directory-access domain-controller discovery domain <domain-name> on page 719**

**Output Fields**

This command displays the discovered domain controllers.

## Sample Output

**request services user-identification active-directory-access domain-controller discovery domain <domain-name>**

`user@host>` **request services user-identification active-directory-access domain-controller discovery domain example.net**

```
Domain: example.net
 Domain controller: example-dc.example.net
   Address: 192.0.2.2
```

# request services user-identification active-directory-access ip-user-probe

### Syntax

```
request services user-identification active-directory-access ip-user-probe
address ip-address <domain name>
```

### Release Information

Command introduced in Junos OS Release 12.1X47-D10.

### Description

Probe the PC at the specified IP address to get an authentication entry, which is used for the integrated user firewall feature. You can display the authentication table to see the results. If the probe succeeded, there will be a valid authentication entry. If the probe failed, there will be an invalid authentication entry.

### Options

- **address *ip-address***—Probe the PC at this IP address.

  **domain *name***—Probe the IP address in the specified domain.

### Required Privilege Level

maintenance

RELATED DOCUMENTATION


clear services user-identification active-directory-access | **711**

*show services user-identification active-directory-access active-directory-authentication-table*

show services user-identification active-directory-access statistics | **795**

user-identification (Services) | **665**


### List of Sample Output

**show services user-identification active-directory-access active-directory-authentication-table address <ip-address> on page 721**

### Output Fields

The following command displays the results of the IP address probe:

## Sample Output

**show services user-identification active-directory-access active-directory-authentication-table address <ip-address>**

`user@host>` **show services user-identification active-directory-access active-directory-authentication-table address 192.0.2.3**

```
Domain: example.net
 Source-ip: 192.0.2.3
  Username: user1
  Groups:group1
  State: Valid
  Source: wmic
  Access start date: 2014-03-10
  Access start time: 13:59:56
  Age time: 1437
```

`user@host>` **show services user-identification active-directory-access active-directory-authentication-table address 2001:db8::1:1**

```
Domain: example.net
 Source-ip: 2001:db8::1:1
  Username: user2
  Groups:group1
  State: Valid
  Source: wmic
  Access start date: 2017-03-10
  Access start time: 13:59:56
  Age time: 1437
```

# request services user-identification authentication-source aruba-clearpass user-query

**Syntax**

request services user-identification authentication-source *authentication-source* user-query address *ip-address*

**Release Information**

Command introduced in Junos OS Release 12.3X48-D30.

**Description**

Manually send to the ClearPass website a request for user authentication and identity information for an individual user. The command specifies the IP address of the user's device to identify the user whose information you want to obtain. If the user query command executes successfully, an entry for the user (IP address) has been created in the ClearPass authentication table, and no output is displayed.

The user query function is part of the SRX Series integrated ClearPass authentication and enforcement feature. The user query function, if configured, allows the SRX Series device to send requests for individual user information. This command also allows you to manually send requests. Normally administrators send query requests manually to troubleshoot issues.

The user query function supplements use of the Web API function. The SRX Series device exposes to ClearPass a Web API that ClearPass uses to send POST request messages to the SRX Series device. These messages contain user authentication and identity information.

**Options**

***ip-address***—The IP address of the user's device for whom you are manually requesting authentication information.

**NOTE:** Starting with Junos OS Release 15.1X49-D130, SRX Series device can query ClearPass for IPv6 addresses, in addition to IPv4 addresses for an individual user.

**Required Privilege Level**

maintenance

**List of Sample Output**

request services user-identification authentication-source authentication-source user-query address ip-address on page 723

## Sample Output

**request services user-identification authentication-source authentication-source user-query address ip-address**

`user@host>` **request services user-identification authentication-source aruba-clearpass user-query address 40.0.0.1**

`user@host>` **request services user-identification authentication-source aruba-clearpass user-query address 2001:db8:4136:e378:8000:63bf:3fff:fdd2**

# request services user-identification authentication-table delete

**Syntax**

request services user-identification authentication-table delete (ip-address *ip-address* | authentication-source (all | active-directory | *authentication-source* (domain *domain-name* | group *group-name* |user *user-name*) )

**Release Information**

Command introduced in Junos OS Release 12.3X48-D30.

**Description**

Delete entries from the ClearPass authentication table based on the IP address of the user's device, or on the authentication source and the name of a domain, a group, or a user. When only the authentication source is specified, the entire ClearPass authentication table is deleted. For the integrated ClearPass authentication and enforcement feature, the authentication source is always aruba-clearpass.

**Options**

***ip-address***—Deletes a user authentication entry from the ClearPass authentication table, and the Active Directory (AD) table, based on the IP address of the user's device.

**NOTE:** Starting with Junos OS Release 15.1X49-D130, SRX Series device supports to delete IPv6 addresses if IPv6 addresses were configured.

***authentication-source*** —Deletes user entries from the ClearPass authentication table. In the CLI, ClearPass as the authentication source is referred to by the value aruba-clearpass as is the ClearPass authentication table. To identify the user entries to be deleted, you specify a domain, a group, or a username.

***domain-name***—Deletes from the ClearPass authentication table user entries for users who belong to the specified domain.

**group *group-name***—Deletes the entry entry from the ClearPass authentication table for users who belong to the group, regardless of whether they belong to other groups.

**user *user-name***—Deletes the entry for the specified user from the ClearPass authentication table.

**Required Privilege Level**

maintenance

**List of Sample Output**

request services user-identification authentication-table delete ip-address on page 725

request services user-identification authentication-table delete authentication-source aruba-clearpass domain on page 726
request services user-identification authentication-table delete authentication-source aruba-clearpass group on page 728
request services user-identification authentication-table delete authentication-source aruba-clearpass on page 731

**Output Fields**

The following examples cover how to delete various user entries from the ClearPass authentication table based on the specified parameter. It also shows how to check to ensure that the user entries were deleted successfully.

## Sample Output

**request services user-identification authentication-table delete ip-address**

`The following command deletes the entry for the user whose device IP address is specified.`

`user@host>` **request services user-identification authentication-table delete ip-address 50.0.0.1**

`user@host>` **request services user-identification authentication-table delete ip-address 2001:db8:4136:e378:8000:63bf:3fff:fdd2**

Before you delete the entry:

To ensure that the entry exists in the ClearPass authentication table, use the following command to display the entry for the user. Note that the ClearPass authentication table includes the user entry with the IP address 50.0.0.1 and 2001:db8:4136:e378:8000:63bf:3fff:fdd2.

`user@host>` **show services user-identification authentication-table ip-address 50.0.0.1**

```
Domain: GLOBAL
  Source-ip: 50.0.0.1
    Username: guest1
    Groups:posture-healthy, guest, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2015-12-14
    Access start time: 17:07:23
    Last updated timestamp: 2015-12-22 05:50:47
    Age time: 0
```

user@host> **show services user-identification authentication-table ip-address 2001:db8:4136:e378:8000:63bf:3fff:fdd2**

```
Domain: GLOBAL
  Source-ip: 2001:db8:4136:e378:8000:63bf:3fff:fdd2
    Username: guest2
    Groups:posture-healthy1, guest, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2015-12-14
    Access start time: 17:07:23
    Last updated timestamp: 2015-12-22 05:50:47
    Age time: 0
```

After you delete the user entry associated with the IP address, enter the command again to verify that the entry has been deleted.

user@host> **show services user-identification authentication-table ip-address 50.0.0.1**

```
warning: “This IP address isn’t in authentication table.”
```

user@host> **show services user-identification authentication-table ip-address 2001:db8:4136:e378:8000:63bf:3fff:fdd2**

```
warning: “This IP address isn’t in authentication table.”
```

**request services user-identification authentication-table delete authentication-source aruba-clearpass domain**

The following command deletes the specified domain.

user@host> **request services user-identification authentication-table delete authentication-source domain global**

Before you delete the domain contents from the ClearPass authentication table, use the following command to display the domain information to ensure that it exists. Note that the ClearPass authentication table includes the global domain.

user@host> **show services user-identification authentication-table authentication-source aruba-clearpass domain global extensive**

```
Domain: GLOBAL
Total entries: 6
  Source-ip: 10.0.0.1
    Username: viki2
    Groups:posture-healthy, accounting-grp, accounting-grp-and-company-device,
    corporate-limited, [user authenticated]
    Groups referenced by policy:accounting-grp-and-company-device
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:20:30
    Last updated timestamp: 2015-12-22 04:02:48
    Age time: 0
  Source-ip: 20.0.0.1
    Username: abew1
    Groups:posture-unknown, marketing-access-limited-grp, [user authenticated]
    Groups referenced by policy:marketing-access-limited-grp
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:31:40
    Last updated timestamp: 2015-12-22 04:18:48
    Age time: 0
  Source-ip: 30.0.0.1
    Username: jxchan
    Groups:posture-healthy, marketing-access-for-pcs-limited-group,
    marketing-general, sales-limited, corporate-limited, [user authenticated]
    Groups referenced by policy:marketing-access-for-pcs-limited-group
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:22:48
    Last updated timestamp: 2015-12-22 05:46:21
    Age time: 0
  Source-ip: 40.0.0.1
    Username: lchen1
    Groups:posture-healthy, human-resources-grp, accounting-limited,
    corporate-limited, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:21:37
    Last updated timestamp: 2015-12-22 05:41:18
    Age time: 0
```

```
  Source-ip: 50.0.0.1
    Username: guest1
    Groups:posture-healthy, guest, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:10
    Last updated timestamp: 2015-12-22 05:50:47
    Age time: 0
  Source-ip: 50.0.0.2
    Username: guest2
    Groups:posture-healthy, guest-device-byod, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:21
    Last updated timestamp: 2015-12-22 05:52:44
    Age time: 0
```

After you delete the domain, use the command again to verify that the domain and its user members was deleted.

`user@host>` **show services user-identification authentication-table authentication-source aruba-clearpass domain global**

```
warning: "There is no related auth entry in authentication-table."
```

#### request services user-identification authentication-table delete authentication-source aruba-clearpass group

The following command deletes the entries for any users who belong to the group posture-healthy.

`user@host>` **request services user-identification authentication-table delete authentication-source aruba-clearpass group posture-healthy**

Before you delete the group contents from the ClearPass authentication table, use the following command to display it to ensure that the group is used in some user entries. Notice that the appropriate user entries contain the posture-healthy group.

```
Domain: GLOBAL
Total entries: 6
  Source-ip: 10.0.0.1
    Username: viki2
```

```
  Groups:posture-healthy, accounting-grp, accounting-grp-and-company-device,
  corporate-limited, [user authenticated]
  Groups referenced by policy:accounting-grp-and-company-device
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:20:30
  Last updated timestamp: 2015-12-22 04:02:48
  Age time: 0
Source-ip: 20.0.0.1
  Username: abew1
  Groups:posture-unknown, marketing-access-limited-grp, [user authenticated]
  Groups referenced by policy:marketing-access-limited-grp
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:31:40
  Last updated timestamp: 2015-12-22 04:18:48
  Age time: 0
Source-ip: 30.0.0.1
  Username: jxchan
  Groups:posture-healthy, marketing-access-for-pcs-limited-group,
  marketing-general, sales-limited, corporate-limited, [user authenticated]
  Groups referenced by policy:marketing-access-for-pcs-limited-group
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:22:48
  Last updated timestamp: 2015-12-22 05:46:21
  Age time: 0
Source-ip: 40.0.0.1
  Username: lchen1
  Groups:posture-healthy, human-resources-grp, accounting-limited,
  corporate-limited, [user authenticated]
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:21:37
  Last updated timestamp: 2015-12-22 05:41:18
  Age time: 0
Source-ip: 50.0.0.1
  Username: guest1
  Groups:posture-healthy, guest, [user authenticated]
  State: Valid
```

```
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:10
    Last updated timestamp: 2015-12-22 05:50:47
    Age time: 0
  Source-ip: 50.0.0.2
    Username: guest2
    Groups:posture-healthy, guest-device-byod, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:21
    Last updated timestamp: 2015-12-22 05:52:44
    Age time: 0
```

Enter the **show services user-identification authentication-table authentication-source aruba-clearpass group posture-healthy** to display the entries for the users who belong to the group posture-healthy.

Notice that the group name does not show up in the column for groups referenced by policy because it is not one. Notice, too, that the output contains information for only those users who belong to the group. It does not include an entry for the user abew1, who does not belong to the group.

```
Domain: GLOBAL
Source IP       Username       groups(Ref by policy)          state
10.0.0.1        viki2          accounting-grp-and-company-dev Valid
30.0.0.1        jxchan         marketing-access-for-pcs-limit Valid
40.0.0.1        lchen1         corporate-limited              Valid
50.0.0.1        guest1                                        Valid
50.0.0.2        guest2                                        Valid
```

After you delete the group, use the command again to verify that it has been deleted.

`user@host>` **show services user-identification authentication-table authentication-source aruba-clearpass group posture-healthy**

```
warning: "There is no related auth entry in authentication-table."
```

For further verification, you can use the following command to check the entry for one of the users who belonged to the group:

user@host> **show services user-identification authentication-table authentication-source aruba-clearpass user viki2**

```
warning: "There is no related auth entry in authentication-table."
```

**request services user-identification authentication-table delete authentication-source aruba-clearpass**

The following command deletes the ClearPass authentication table (aruba-clearpass).

user@host> **request services user-identification authentication-table delete authentication-source aruba-clearpass**

Before you delete the ClearPass authentication table, use the following command to display it to ensure that the table exists.

user@host> **show services user-identification authentication-table authentication-source aruba-clearpass**

```
Domain: GLOBAL
Total entries: 6
  Source-ip: 10.0.0.1
    Username: viki2
    Groups:posture-healthy, accounting-grp, accounting-grp-and-company-device,
    corporate-limited, [user authenticated]
    Groups referenced by policy:accounting-grp-and-company-device
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:20:30
    Last updated timestamp: 2015-12-22 04:02:48
    Age time: 0
  Source-ip: 20.0.0.1
    Username: abew1
    Groups:posture-unknown, marketing-access-limited-grp, [user authenticated]
    Groups referenced by policy:marketing-access-limited-grp
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:31:40
    Last updated timestamp: 2015-12-22 04:18:48
    Age time: 0
  Source-ip: 30.0.0.1
    Username: jxchan
    Groups:posture-healthy, marketing-access-for-pcs-limited-group,
    marketing-general, sales-limited, corporate-limited, [user authenticated]
```

```
    Groups referenced by policy:marketing-access-for-pcs-limited-group
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:22:48
    Last updated timestamp: 2015-12-22 05:46:21
    Age time: 0
  Source-ip: 40.0.0.1
    Username: lchen1
    Groups:posture-healthy, human-resources-grp, accounting-limited,
    corporate-limited, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:21:37
    Last updated timestamp: 2015-12-22 05:41:18
    Age time: 0
  Source-ip: 50.0.0.1
    Username: guest1
    Groups:posture-healthy, guest, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:10
    Last updated timestamp: 2015-12-22 05:50:47
    Age time: 0
  Source-ip: 50.0.0.2
    Username: guest2
    Groups:posture-healthy, guest-device-byod, [user authenticated]
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:21
    Last updated timestamp: 2015-12-22 05:52:44
    Age time: 0
```

To verify that you deleted the authentication table successfully, enter the command again:

user@host> **show services user-identification authentication-table authentication-source aruba-clearpass**

```
warning: "There is no authentication-table entry."
```

# show network-access requests pending

**Syntax**

```
show network-access requests pending
<detail>
<index number >
```

**Release Information**

Command introduced in Release 8.5 of Junos OS.

**Description**

Display the status of pending authentication requests.

**Options**

- none—Show pending authentication requests.
- **show network-access requests pendingshow network-access requests pending****detail**—Display detailed information about all pending requests.
- **index *number***—(Optional) Display detailed information about the request specified by this index number. Use the command without options to obtain a list of requests and index numbers.

**Required Privilege Level**

view

RELATED DOCUMENTATION

clear network-access requests pending | **691**

**List of Sample Output**

**show network-access requests pending on page 734**
**show network-access requests pending detail on page 735**
**show network-access requests pending index 1 on page 735**

**Output Fields**

Table 28 on page 734 lists the output fields for the show network-access requests pending command. Output fields are listed in the approximate order in which they appear.

**Table 28: show network-access requests pending Output Fields**

| Field Name | Field Description |
|---|---|
| **Index** | Internal number identifying the pending request. Use this number to obtain more information on the record. |
| **User** | Originator of authentication request. |
| **Status** | The pending requests are requests and responses that are not yet sent back to the respective clients. The pending requests can be in one of the following states:<br>• **Processing**: This request is being processed by the device. The authentication process has started but is not complete.<br>• **Waiting on Auth Server**: The request is sent to an external authentication server, and the device is waiting for the response.<br>• **Processed:** This request has completed authentication (success or failure). The results are not yet forwarded back to the client.<br>• **Request cancelled by Admin**: This request was cancelled by the Admin. The reply with cancel code is not yet sent back to the client. |
| **Profile** | The profile determines how the user is authenticated.<br>Local clients defined with the statement **access profile client** are authenticated with the password authentication. Clients configured external to the device, on a RADIUS or LDAP server are authenticated with RADIUS or LDAP authentication. |

## Sample Output

**show network-access requests pending**

`user@host>` **show network-access requests pending**

```
Information about pending authentication entries
  Total pending authentication requests: 2
Index User                 Status
1     Sun                  Processing
2     Sam                  Processed
```

## Sample Output

**show network-access requests pending detail**

```
user@host> show network-access requests pending detail
```

```
Information about pending authentication entries
  Total pending authentication requests: 2
Index: 1  User: Sun
  Status: Processing
  Profile: Sunnyvale-firewall-users
Index: 2  User: Sam
  Status: Processed
  Profile: Westford-profile
```

## Sample Output

**show network-access requests pending index 1**

```
user@host> show network-access requests pending index 1
```

```
Index: 1  User: Sun
  Status: Processing
  Profile: Sunnyvale-firewall-users
```

# show network-access requests statistics

**Syntax**

```
show network-access requests statistics
```

**Release Information**

Command modified in Release 9.1 of Junos OS.

**Description**

Display authentication statistics for the configured authentication type.

**Required Privilege Level**

view

RELATED DOCUMENTATION

clear network-access requests statistics | 693

**Output Fields**

Table 29 on page 736 lists the output fields for the network-access requests statistics command. Output fields are listed in the approximate order in which they appear.

**Table 29: show network-access requests statistics Output Fields**

| Field Name | Field Description |
|---|---|
| **Total requests received** | Total number of authentication requests that the device received from clients. |
| **Total responses sent** | Total number of authentication responses that the device sent to the clients. |
| **Success responses** | Total number of clients that authenticated successfully. |
| **Failure responses** | Total number of clients that failed to authenticate. |

## show network-access requests statistics

```
user@host> show network-access requests statistics
```

```
General authentication statistics
  Total requests received: 100
  Total responses sent: 70
Radius authentication statistics
  Total requests received: 40
  Success responses: 20
  Failure responses: 20
Radius reauthentication statistics
  Total requests received: 0
  Success responses: 0
  Failure responses: 0
LDAP authentication statistics
  Total requests received: 30
  Success responses: 15
  Failure responses: 15
Local authentication statistics
  Total requests received: 5
  Success responses: 2
  Failure responses: 3
Local re-authentication statistics
  Total requests received: 0
  Success responses: 0
  Failure responses: 0
Securid authentication statistics
  Total requests received: 15
  Success responses: 3
  Failure responses: 12
```

# show network-access securid-node-secret-file

**Syntax**

```
show network-access securid-node-secret-file
```

**Release Information**

Command introduced in Release 9.1 of Junos OS.

**Description**

Display the path to the node secret file for the SecurID authentication type.

**Required Privilege Level**

view

RELATED DOCUMENTATION

- configuration-file | 459
- securid-server | 606
- clear network-access securid-node-secret-file | 694

**List of Sample Output**

**show network-access securid-node-secret-file on page 739**

**Output Fields**

Table 30 on page 738 lists the output fields for the network-access securid-node-secret-file command. Output fields are listed in the approximate order in which they appear.

**Table 30: show network-access securid-node-secret-file Output Fields**

| Field Name | Field Description |
|---|---|
| **SecurID Server** | Name of the SecurID authentication server. |
| **Node Secret File** | Path to the node secret file. |

## Sample Output

**show network-access securid-node-secret-file**

```
user@host> show network-access securid-node-secret-file
```

```
SecurID server node secret file:
SecurID Server       Node Secret File
ace-server1          /var/db/securid/ace-server1/node-secret
```

# show security firewall-authentication history

**Syntax**

**show security firewall-authentication history**
<address (*address*)>
<from-zone (*from-zone*)>
<identifier (*identifier*)>
<logical-system (*logical-system-name* | all)>
<node (***node-id*** | all | local | primary)>
<root-logical-system (address | from-zone | identifier | tenant | to-zone)>
<tenant (*tenant-name* | all)>
<to-zone (*to-zone*)>

**Release Information**
Command introduced in Junos OS Release 8.5. The **node** option is added in Junos OS Release 9.0. The **tenant** option is introduced in Junos OS Release 18.3R1.

**Description**
Displays security firewall authentication user history information and verify the number of firewall users who successfully authenticated and the number of firewall users who failed to log in.

**Options**
- none—Display history of firewall authentication information.
- address—Display authentication entries based on IP address.
- from-zone—Display authentication entries matching the given source zone, null for web-authentication and userfw-authentication.
- identifier—Display authentication entries by user identifier.
- logical-system—Display firewall authentication tables based on logical system name.
- **node**—(Optional) For chassis cluster configurations, display all firewall authentication history on a specific node (device) in the cluster.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.
- root-logical-system—Display firewall authentication tables for root logical system.

- tenant—Display firewall authentication tables based on tenant name.
- to-zone—Display authentication entry matching the given destination zone, null for web-auth and userfw-auth.

**Required Privilege Level**
view

RELATED DOCUMENTATION

*Understanding Logical System Firewall Authentication*

Firewall User Authentication Overview | 35

**List of Sample Output**
**show security firewall-authentication history on page 742**
**show security firewall-authentication history node all on page 742**
**show security firewall-authentication history tenant tn1 on page 742**

**Output Fields**
Table 31 on page 741 lists the output fields for the **show security firewall-authentication history** command. Output fields are listed in the approximate order in which they appear.

**Table 31: show security firewall-authentication history Output Fields**

| Field Name | Field Description |
|---|---|
| **Authentications** | Number of authentications. |
| **Id** | Identification number. |
| **Source IP** | IP address of the authentication source. |
| **Date** | Authentication date. |
| **Time** | Authentication time. |
| **Duration** | Authentication duration. |
| **Status** | Authentication status success or failure. |
| **User** | Name of the user. |

## Sample Output

**show security firewall-authentication history**

```
user@host> show security firewall-authentication history
```

```
History of firewall authentication data:
  Authentications: 1
          Id Source Ip       Date       Time     Duration   Status   User
           1 203.0.113.1     2007-04-03 11:43:06 00:00:45   Success  hello
```

## Sample Output

**show security firewall-authentication history node all**

```
user@host> show security firewall-authentication history node all
```

```
node0:
--------------------------------------------------------------------------
History of firewall authentication data:
Authentications: 2
Id Source Ip       Date       Time     Duration   Status   User
1 203.0.113.1      2008-01-04 12:00:10 0:05:49    Success  local1
2 203.0.113.1      2008-01-04 14:36:52 0:01:03    Success  local1
node1:
--------------------------------------------------------------------------
History of firewall authentication data:
  Authentications: 1
          Id Source Ip       Date       Time     Duration   Status   User
           203.0.113.1       2008-01-04 14:59:43 1193046:06: Success local1
```

**show security firewall-authentication history tenant tn1**

```
user@host> show security firewall-authentication history tenant tn1
```

```
History of firewall authentication data:
  Authentications: 0
```

# show security firewall-authentication history address

### Syntax

```
show security firewall-authentication history address   ip-address
<node (  node-id   | all | local | primary)>
```

### Release Information

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

### Description

Display security firewall authentication history for this source IP address.

### Options

- **address**   ***ip-address*** —IP address of the authentication source.
- none—Display all firewall authentication history for this address.
- **node**—(Optional) For chassis cluster configurations, display firewall authentication history for this address on a specific node.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.

### Required Privilege Level

view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

### List of Sample Output

**show security firewall-authentication history address 198.51.100.17 on page 744**
**show security firewall-authentication history address 198.51.100.17 node local on page 745**

### Output Fields

Table 32 on page 744 lists the output fields for the **show security firewall-authentication history address** command. Output fields are listed in the approximate order in which they appear.

Table 32: show security firewall-authentication history address Output Fields

| Field Name | Field Description |
|---|---|
| **Username** | User ID. |
| **Source IP** | IP address of the authentication source. |
| **Authentication state** | Status of authentication (success or failure). |
| **Authentication method** | Path chosen for authentication. |
| **Access start date** | Date when user authenticated. |
| **Access start time** | Time when user authenticated. |
| **Duration of user access** | Time duration of the accessing firewall. |
| **Policy name** | Name of the policy. |
| **Source zone** | User traffic received from the zone. |
| **Destination zone** | User traffic destined to the zone. |
| **Access profile** | Name of profile used for authentication. |
| **Bytes sent by this user** | Number of bytes sent by the user. |
| **Bytes received by this user** | Number of bytes received by the user. |

## Sample Output

**show security firewall-authentication history address 198.51.100.17**

`user@host>` **show security firewall-authentication history address 198.51.100.17**

```
Username: u1
Source IP: 198.51.100.17
Authentication state: Success
Authentication method: Pass-through using HTTP
Access start date: 2007-09-12
Access start time: 15:33:29
Duration of user access: 0:00:48
```

```
Policy name: Z1-Z2
Source zone: Z1
Destination zone: Z2
Access profile: profile-local
Bytes sent by this user: 0
Bytes received by this user: 449
```

## Sample Output

**show security firewall-authentication history address 198.51.100.17 node local**

`user@host>` **show security firewall-authentication history address 198.51.100.17 node local**

```
node0:
--------------------------------------------------------------------------
Username: local1
Source IP: 198.51.100.17
Authentication state: Success
Authentication method: Pass-through using Telnet
Access start date: 2008-01-04
Access start time: 12:00:10
Duration of user access: 0:05:49
Policy name: POL1
Source zone: z1
Destination zone: z2
Access profile: p1
Bytes sent by this user: 0
Bytes received by this user: 0
Username: local1
Source IP: 198.51.100.17
Authentication state: Success
Authentication method: Pass-through using Telnet
Access start date: 2008-01-04
Access start time: 14:36:52
Duration of user access: 0:01:03
Policy name: POL1
Source zone: z1
Destination zone: z2
Access profile: p1
Bytes sent by this user: 2178
Bytes received by this user: 4172
```

# show security firewall-authentication history identifier

### Syntax

**show security firewall-authentication history identifier** ***identifier***
<node ( ***node-id*** | all | local | primary)>

### Release Information

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

### Description

Display security firewall authentication history information for the authentication with this identifier.

### Options

- **identifier** ***identifier***—Identifying number of the authentication process.
- none—Display all firewall authentication history information for the authentication with this identifier.
- **node**—(Optional) For chassis cluster configurations, display firewall authentication history on a specific node for the authentication with this identifier.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.

### Required Privilege Level

view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

### List of Sample Output

**show security firewall-authentication history identifier 1 on page 747**
**show security firewall-authentication identifier 1 node primary on page 748**

### Output Fields

Table 33 on page 747 lists the output fields for the **show security firewall-authentication history identifier** command. Output fields are listed in the approximate order in which they appear.

Table 33: show security firewall-authentication history identifier Output Fields

| Field Name | Field Description |
|---|---|
| **Username** | User ID. |
| **Source IP** | IP address of the authentication source. |
| **Authentication state** | Status of authentication (success or failure). |
| **Authentication method** | Path chosen for authentication. |
| **Access start date** | Date when user authenticated. |
| **Access start time** | Time when user authenticated. |
| **Duration of user access** | Time duration of the accessing firewall. |
| **Policy index** | Identification number of the policy. |
| **Policy name** | Name of the policy. |
| **Source zone** | User traffic received from the zone. |
| **Destination zone** | User traffic destined to the zone. |
| **Access profile** | Name of profile used for authentication. |
| **Bytes sent by this user** | Number of bytes sent by the user. |
| **Bytes received by this user** | Number of bytes received by the user. |
| **Client-groups** | Name of the client group. |

## Sample Output

**show security firewall-authentication history identifier 1**

`user@host>` **show security firewall-authentication history identifier 1**

```
Username: hello
Source IP: 192.0.2.5
Authentication state: Success
```

```
Authentication method: Pass-through using Telnet
Access start date: 2007-04-03
Access start time: 11:43:06
Duration of user access: 00:00:45
Policy index: 4
Source zone: z2
Destination zone: z1
Access profile: profile1
Bytes sent by this user: 0
Bytes received by this user: 1050
Client-groups: Sunnyvale Bangalore
```

## Sample Output

**show security firewall-authentication identifier 1 node primary**

`user@host>` **show security firewall-authentication history identifier 1 node primary**

```
node0:
--------------------------------------------------------------------------
Username: local1
Source IP: 192.0.2.5
Authentication state: Success
Authentication method: Pass-through using Telnet
Access start date: 2008-01-04
Access start time: 12:00:10
Duration of user access: 0:05:49
Policy name: POL1
Source zone: z1
Destination zone: z2
Access profile: p1
Bytes sent by this user: 0
Bytes received by this user: 0
```

# show security firewall-authentication jims statistics

**Syntax**

```
show security firewall-authentication jims statistics
```

**Release Information**

Command introduced in Junos OS Release 15.1X49-D100.

**Description**

Display statistics of primary and secondary Juniper Identity Management Service (JIMS) server.

**Required Privilege Level**

view

RELATED DOCUMENTATION

Configure Juniper Identity Management Service to Obtain User Identity Information | 275

**Output Fields**

Table 34 on page 749 lists the output fields for the **show security firewall-authentication jims statistics** command. Output fields are listed in the approximate order in which they appear.

**Table 34: show security firewall-authentication jims statistics Output Fields**

| Field Name | Field Description |
|---|---|
| **Push success counter** | Number of authentication entries successfully pushed to JIMS server. |
| **Push failure counter** | Number of authentication entries failed to be pushed to JIMS server. |

## Sample Output

**show security firewall-authentication jims statistics**

`user@host>` **show security firewall-authentication jims statistics**

```
Push success counter: 1
Push failure counter: 0
```

## Sample Output

Starting in Junos OS Release 18.3R2, the output for **show security firewall-authentication jims statistics** operational command is changed to display the statistics of both primary and secondary JIMS server.

**show security firewall-authentication jims statistics**

```
user@host> show security firewall-authentication jims statistics

Primary server:
  Push success counter: 1
  Push failure counter: 0

Secondary server:
  Push success counter: 1
  Push failure counter: 0
```

# show security firewall-authentication users

**Syntax**

```
show security firewall-authentication users
<address (ip-address )>
<auth-type (pass-through | user-firewall | web-authentication)>
<from-zone (from-zone)>
<identifier (identifier )>
<logical-system (logical-system-name | all)>
<node (node-id | all | local | primary)>
<root-logical-system (address | auth-type | from-zone | identifier | tenant | to-zone>
<tenant (tenant-name |all)>
<to-zone (to-zone )>
```

**Release Information**

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0. The **tenant** option is introduced in Junos OS Release 18.3R1.

**Description**

Display firewall authentication details about all users and verify the number of firewall users who successfully authenticated and firewall users who failed to log in.

**Options**

- none—Display details about all firewall authentication users.
- address—Display authentication entries based on ip address.
- auth-type—Display authentication entries matching the given auth-type.
- from-zone—Display authentication entries matching the given source zone, null for web-auth and userfw-auth.
- identifier—Display authentication entries by id.
- logical-system—Display firewall authentication tables based on logical system name.
- **node**—(Optional) For chassis cluster configurations, display firewall authentication details for all users on a specific node.
  - *node-id*—Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.
- root-logical-system—Display firewall authentication tables for root logical system.

- tenant—Display firewall authentication tables based on tenant name.
- to-zone—Display authentication entry matching the given destination zone, null for web-auth and userfw-auth.

**Required Privilege Level**
view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**
**show security firewall-authentication users on page 753**
**show security firewall-authentication users node 0 on page 753**
**show security firewall-authentication users node all on page 753**

**Output Fields**
Table 35 on page 752 lists the output fields for the **show security firewall-authentication users** command. Output fields are listed in the approximate order in which they appear.

**Table 35: show security firewall-authentication users Output Fields**

| Field Name | Field Description |
|---|---|
| **Total users in table** | Gives count of how many entries/users the command will display. |
| **Id** | Identification number. |
| **Source IP** | IP address of the authentication source. |
| **Src zone** | User traffic received from the zone. |
| **Dst zone** | User traffic destined to the zone. |
| **Profile** | Name of profile used for authentication. |
| **Age** | Idle timeout for the user. |
| **Status** | Authentication status success or failure. |
| **User** | Name of the user. |

## Sample Output

**show security firewall-authentication users**

```
user@host> show security firewall-authentication users
```

```
Firewall authentication data:
  Total users in table: 1
          Id Source Ip       Src zone Dst zone Profile    Age Status   User
           1 192.0.2.5/24         z1       z2       p1       0 Success  local1
```

## Sample Output

**show security firewall-authentication users node 0**

```
user@host> show security firewall-authentication users node 0
```

```
node0:
--------------------------------------------------------------------------
Firewall authentication data:
  Total users in table: 1
          Id Source Ip       Src zone Dst zone Profile    Age Status   User

           3 192.0.2.5/24        z1       z2       p1           1 Success  local1
```

## Sample Output

**show security firewall-authentication users node all**

```
user@host> show security firewall-authentication users node all
```

```
node0:
--------------------------------------------------------------------------
Firewall authentication data:
  Total users in table: 1
          Id Source Ip       Src zone Dst zone Profile    Age Status   User
           3 192.0.2.5        z1       z2       p1           1 Success  local1
```

```
node1:
--------------------------------------------------------------------------
Firewall authentication data:
  Total users in table: 1
          Id Source Ip       Src zone Dst zone Profile    Age Status   User
           2 192.0.2.5       z1       z2       p1           1 Success  local1
```

## show security firewall-authentication users tenant all

user@host> **show security firewall-authentication users tenant all**

```
Firewall authentication data:
  Total users in table: 1
          Id Source Ip                          Src zone Dst zone Profile    Age
Status   User
           2 192.0.2.10                        N/A      N/A      test-rad     1
Success  b1
```

# show security firewall-authentication users address

### Syntax

```
show security firewall-authentication users address   ip-address
<node ( node-id  | all | local | primary)>
```

### Release Information

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

### Description

Display information about the users at the specified IP address that are currently authenticated.

### Options

- **address *ip-address***—IP address of the authentication source.
- none—Display all the firewall authentication information for users at this IP address.
- **node**—(Optional) For chassis cluster configurations, display user firewall authentication entries on a specific node.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.

### Required Privilege Level

view

RELATED DOCUMENTATION

*Understanding User Role Firewalls*

### List of Sample Output

**show security firewall-authentication users address 192.0.2.9 on page 756**
**show security firewall-authentication users address 192.0.2.9 node local on page 757**
**show security firewall-authentication users address 198.51.100.29 on page 757**

### Output Fields

Table 36 on page 756 lists the output fields for the **show security firewall-authentication users address** command. Output fields are listed in the approximate order in which they appear.

Table 36: show security firewall-authentication users address Output Fields

| Field Name | Field Description |
|---|---|
| **Username** | User ID. |
| **Source IP** | IP address of the authentication source. |
| **Authentication state** | Status of authentication (success or failure). |
| **Authentication method** | Path chosen for authentication. |
| **Access time remaining** | Duration for which the connection exists. |
| **Lsys** | The logical system where the traffic was received. |
| **Source zone** | User traffic received from the zone. |
| **Destination zone** | User traffic destined to the zone. |
| **Policy index** | Identification number of the policy. |
| **Policy name** | Name of the policy. |
| **Access profile** | Name of profile used for authentication. |
| **Interface Name** | Name of the interface. |
| **Bytes sent by this user** | Number of bytes sent by the user. |
| **Bytes received by this user** | Number of bytes received by the user. |
| **Client-groups** | Name of the client group. |

## Sample Output

**show security firewall-authentication users address 192.0.2.9**

`user@host>` **show security firewall-authentication users address 192.0.2.9**

```
Username: hello
Source IP: 192.0.2.9
Authentication state: Success
```

```
Authentication method: Pass-through using Telnet
Access time remaining: 0
Source zone: z2
Destination zone: z1
Policy index: 5
Access profile: profile1
Interface Name: ge-0/0/2.0
Bytes sent by this user: 0
Bytes received by this user: 0
Client-groups: my-group1-example, my-group2-example
```

## Sample Output

**show security firewall-authentication users address 192.0.2.9 node local**

`user@host>` **show security firewall-authentication users address 192.0.2.9 node local**

```
node0:
--------------------------------------------------------------------------
Username: local1
Source IP: 192.0.2.9
Authentication state: Success
Authentication method: Pass-through using Telnet
Age: 2
Access time remaining: 4
Source zone: z1
Destination zone: z2
Policy name: POL1
Access profile: p1
Interface Name: reth1.0
Bytes sent by this user: 614
Bytes received by this user: 1880
```

**show security firewall-authentication users address 198.51.100.29**

`user@host>` **show security firewall-authentication users address 198.51.100.29**

```
Username: hello
Source IP: 198.51.100.29/24
Authentication state: Success
Authentication method: User-firewall
```

```
Age: 0
Access time remaining: 10
Lsys: root-logical-system
Source zone: N/A
Destination zone: N/A
Access profile: test
```

# show security firewall-authentication users identifier

### Syntax

**show security firewall-authentication users identifier** ***identifier***
<node ( ***node-id*** | all | local | primary)>

### Release Information

Command introduced in Junos OS Release 8.5. The **node** options added in Junos OS Release 9.0.

### Description

Display firewall authentication details about the user with this identification number.

### Options

- **identifier** ***identifier***—Identification number of the user for which to display authentication details.
- **node**—(Optional) For chassis cluster configurations, display the firewall authentication details security firewall authentication entry on a specific node (device) in the cluster for the user with this identification number.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.

### Required Privilege Level

view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

### List of Sample Output

**show security firewall-authentication users identifier 3 on page 760**
**show security firewall-authentication users identifier 3 node primary on page 761**
**show security firewall-authentication users identifier 10 on page 761**

### Output Fields

Table 37 on page 760 lists the output fields for the **show security firewall-authentication users identifier** command. Output fields are listed in the approximate order in which they appear.

Table 37: show security firewall-authentication users identifier Output Fields

| Field Name | Field Description |
|---|---|
| **Username** | User ID. |
| **Source IP** | IP address of the authentication source. |
| **Authentication state** | Status of authentication (success or failure). |
| **Authentication method** | Path chosen for authentication. |
| **Age** | Idle timeout for the user. |
| **Access time remaining** | Duration for which the connection exists. |
| **Source zone** | User traffic received from the zone. |
| **Destination Zone** | User traffic destined to the zone. |
| **Policy Name** | Name of the policy. |
| **Access profile** | Name of profile used for authentication. |
| **Interface Name** | Name of the interface |
| **Bytes sent by this user** | Number of bytes sent by the user. |
| **Bytes received by this user** | Number of bytes received by the user. |

## Sample Output

**show security firewall-authentication users identifier 3**

`user@host>` **show security firewall-authentication users identifier 3**

```
Username: u1
Source IP: 198.51.100.39
Authentication state: Success
Authentication method: Pass-through using HTTP
Age: 1
Access time remaining: 254
Source zone: Z1
```

```
Destination zone: Z2
Policy name: Z1-Z2
Access profile: profile-local
Interface Name: ge-0/0/1.0
Bytes sent by this user: 0
Bytes received by this user: 449
```

## Sample Output

**show security firewall-authentication users identifier 3 node primary**

`user@host>` **show security firewall-authentication users identifier 3 node primary**

```
node0:
--------------------------------------------------------------------------
Username: local1
Source IP: 198.51.100.39
Authentication state: Success
Authentication method: Pass-through using Telnet
Age: 1
Access time remaining: 5
Source zone: z1
Destination zone: z2
Policy name: POL1
Access profile: p1
Interface Name: reth1.0
Bytes sent by this user: 614
Bytes received by this user: 1880
```

**show security firewall-authentication users identifier 10**

`user@host>` **show security firewall-authentication users identifier 10**

```
Username: test
Source IP: 192.0.2.231
Authentication state: Success
Authentication method: Web-authentication using HTTP
Age: 1
Access time remaining: 9
Lsys: root-logical-system
Tenant: tenant-aa
```

```
Source zone: N/A
Destination zone: N/A
Access profile: test
Bytes sent by this user: 0
Bytes received by this user: 0
```

# show security user-identification local-authentication-table

**Syntax**

```
show security user-identification local-authentication-table [ ( all [brief | extensive]) |ip-address ip-address | role
  role-name | start value | count value | user user-name]
```

**Release Information**

Command introduced in Junos OS Release 12.1.

**Description**

This command displays the content of the local authentication table by IP address.

**all**—(Optional) All entries displayed from the beginning of the table or from the specified starting entry.

**brief**—(Default) Uses a tabular format and truncates longer entries: username—displays up to 13 characters, roles—displays up to 32 characters.

**extensive**—(Optional) Displays the full names and all items.

**count *value***—(Optional) The total number of entries to display.

**ip-address *ip-address***—(Optional) The IP address of the entry to display.

**role *role-name***—(Optional) The role name of the entries to display.

**start *value***—(Optional) The first entry to display.

**user *user-name***—(Optional) The username of the entry to display.

**Required Privilege Level**

view

RELATED DOCUMENTATION

request security user-identification local-authorization-table add | 714

*Understanding the User Identification Table*

**List of Sample Output**

show security user-identification local-authentication-table all on page 764
show security user-identification local-authentication-table ip-address on page 764
show security user-identification local-authentication-table start on page 765

show security user-identification local-authentication-table role on page 765

**Output Fields**

Table 38 on page 764 lists the output fields for the **show security user-identification local-authentication-table** command. Output fields are listed in the approximate order in which they appear.

**Table 38: show security user-identification local-authentication-table Output Fields**

| Field Name | Field Description |
|---|---|
| **Total entries** | The number of entries in the table. |
| **IP address** | IP address of the associated user.<br>**NOTE:** Only one user can be associated with an IP address. |
| **Username** | User associated with the specified IP address. |
| **Roles** | A comma-separated list of all roles associated with this IP address and user. |

## Sample Output

**show security user-identification local-authentication-table all**

```
user@host> show security user-identification local-authentication-table all
```

```
Total entries: 3
Source IP       Username     Roles
192.0.2.1       user1        role1
203.0.113.2     user1        role2
198.51.100.3    user3        role1, role2
```

**show security user-identification local-authentication-table ip-address**

```
user@host> show security user-identification local-authentication-table ip-address 203.0.113.2
```

```
Ip-address: 203.0.113.2
    Username: user2
    Roles: role2, role3, role1
```

**show security user-identification local-authentication-table start**

```
user@host> show security user-identification local-authentication-table start 2 count 2
```

```
Total entries: 2
    Ip-address: 203.0.113.2
    Username: user2
    Roles: role2, role3, role1

    Ip-address: 198.51.100.3    Username: user3
    Roles: role2, role3
```

**show security user-identification local-authentication-table role**

```
user@host> show security user-identification local-authentication-table role qa3456
```

```
Total entries: 3
    Ip-address: 203.0.113.2
    Username: dev-grp-3
    Roles: qa432, qa3456, qa84, qa794

    Ip-address: 198.51.100.3
    Username: dev-qa
    Roles: qa3456, qa3985, qa23

    Ip-address: 203.0.113.2
    Username: brandall
    Roles: qa3456
```

# show security policies

**Syntax**

```
show security policies
application-firewall
count
detail
from-zone <zone-name>
global
hit-count
interface
logical-system <logical-system-name>
policy <policy-name>
root-logical-system
service-set
start
tenant <tenant-name>
to-zone <zone-name>
unknown-source-identity
zone-context
```

**Release Information**

Command modified in Junos OS Release 9.2.
Support for IPv6 addresses is added in Junos OS Release 10.2.
Support for wildcard addresses is added in Junos OS Release 11.1.
Support for global policy and services offloading is added in Junos OS Release 11.4.
Support for source-identities and the **Description** output field is added in Junos OS Release 12.1.
Support for negated address added in Junos OS Release 12.1X45-D10.
The output fields for Policy Statistics expanded, and the output fields for the **global** and **policy-name** options are expanded to include from-zone and to-zone global match criteria in Junos OS Release 12.1X47-D10.
Support for the **initial-tcp-mss** and **reverse-tcp-mss** options is added in Junos OS Release 12.3X48-D20.
Output field and description for **source-end-user-profile** option is added in Junos OS Release 15.1x49-D70.
Output field and description for **dynamic-applications** option is added in Junos OS Release 15.1x49-D100.
Output field and description for **dynapp-redir-profile** option is added in Junos OS Release 18.2R1.
The **tenant** option is introduced in Junos OS Release 18.3R1.

**Description**

Displays a summary of all security policies configured on the device. If a particular policy is specified, display information specific to that policy. The existing show commands for displaying the policies configured with multiple tenant support are enhanced. A security policy controls the traffic flow from one zone to another zone. The security policies allow you to deny, permit, reject (deny and send a TCP RST or ICMP

port unreachable message to the source host), encrypt and decrypt, authenticate, prioritize, schedule, filter, and monitor the traffic attempting to cross from one security zone to another.

**Options**

- **application-firewall**—Displays the information of application-firewall.
- **count**—Displays the number of policies. Range is 1 through 65,535.
- **detail**—(Optional) Displays a detailed view of all of the policies configured on the device.
- **from-zone**—Displays the policy information matching the given source zone.
- **global**—(Optional) Displays information about global policies.
- **hit-count**—Displays the policies hit count.
- **interface**—Displays the name of the adaptive services interface.
- **logical-system**—Displays the logical system name.
- **policy-name**—(Optional) Displays the information about a specified policy.
- **root-logical-system**—Displays root logical system as default.
- **service-set**—Displays the name of the service set.
- **start**—Displays the policies from a given position. Range is 1 through 65,535.
- **tenant**—Displays the name of the tenant system.
- **to-zone**—Displays the policy information matching the given destination zone.
- **unknown-source-identity**—Displays the unknown-source-identity of a policy.
- **zone-context**—Displays the count of policies in each context (from-zone and to-zone).

**Required Privilege Level**

view

## RELATED DOCUMENTATION

- *Security Policies Overview*
- *Understanding Security Policy Rules*
- *Understanding Security Policy Elements*
- *Unified Policies Configuration Overview*

**List of Sample Output**

**show security policies on page 772**
**show security policies (Dynamic Applications) on page 772**
**show security policies policy-name detail on page 773**

show security policies (Services-Offload) on page 775
show security policies (Device Identity) on page 775
show security policies detail on page 776
show security policies detail (TCP Options) on page 779
show security policies policy-name (Negated Address) on page 779
show security policies policy-name detail (Negated Address) on page 780
show security policies global on page 780
show security policies detail tenant on page 781

### Output Fields

Table 39 on page 768 lists the output fields for the **show security policies** command. Output fields are listed in the approximate order in which they appear.

**Table 39: show security policies Output Fields**

| Field Name | Field Description |
|---|---|
| **From zone** | Name of the source zone. |
| **To zone** | Name of the destination zone. |
| **Policy** | Name of the applicable policy. |
| **Description** | Description of the applicable policy. |
| **State** | Status of the policy:<br>• **enabled**: The policy can be used in the policy lookup process, which determines access rights for a packet and the action taken in regard to it.<br>• **disabled**: The policy cannot be used in the policy lookup process, and therefore it is not available for access control. |
| **Index** | Internal number associated with the policy. |
| **Sequence number** | Number of the policy within a given context. For example, three policies that are applicable in a from-zoneA-to-zoneB context might be ordered with sequence numbers 1, 2, 3. Also, in a from-zoneC-to-zoneD context, four policies might have sequence numbers 1, 2, 3, 4. |
| **Source addresses** | For standard display mode, the names of the source addresses for a policy. Address sets are resolved to their individual names.<br>For detail display mode, the names and corresponding IP addresses of the source addresses for a policy. Address sets are resolved to their individual address name-IP address pairs. |

**Table 39: show security policies Output Fields *(continued)***

| Field Name | Field Description |
|---|---|
| **Destination addresses** | Name of the destination address (or address set) as it was entered in the destination zone's address book. A packet's destination address must match this value for the policy to apply to it. |
| **source-end-user-profile** | Name of the device identity profile (referred to as **end-user-profile** in the CLI) that contains attributes, or characteristics of a device. Specification of the device identity profile in the **source-end-user-profile** field is part of the device identity feature. If a device matches the attributes specified in the profile and other security policy parameters, then the security policy's action is applied to traffic issuing from the device. |
| **Source addresses (excluded)** | Name of the source address excluded from the policy. |
| **Destination addresses (excluded)** | Name of the destination address excluded from the policy. |
| **Source identities** | One or more user roles specified for a policy. |
| **Applications** | Name of a preconfigured or custom application whose type the packet matches, as specified at configuration time.<br>• **IP protocol**: The Internet protocol used by the application—for example, TCP, UDP, ICMP.<br>• **ALG**: If an ALG is explicitly associated with the policy, the name of the ALG is displayed. If **application-protocol ignore** is configured, **ignore** is displayed. Otherwise, **0** is displayed.<br>However, even if this command shows **ALG: 0**, ALGs might be triggered for packets destined to well-known ports on which ALGs are listening, unless ALGs are explicitly disabled or when **application-protocol ignore** is not configured for custom applications.<br>• **Inactivity timeout**: Elapsed time without activity after which the application is terminated.<br>• **Source port range**: The low-high source port range for the session application. |
| **Dynamic Applications** | Application identification-based Layer 7 dynamic applications. |
| **Destination Address Translation** | Status of the destination address translation traffic:<br>• **drop translated**—Drop the packets with translated destination addresses.<br>• **drop untranslated**—Drop the packets without translated destination addresses. |

**Table 39: show security policies Output Fields *(continued)***

| Field Name | Field Description |
|---|---|
| **Application Firewall** | An application firewall includes the following:<br>• **Rule-set**—Name of the rule set.<br>• **Rule**—Name of the rule.<br>  ◦ **Dynamic applications**—Name of the applications.<br>  ◦ **Dynamic application groups**—Name of the application groups.<br>  ◦ **Action**—The action taken with respect to a packet that matches the application firewall rule set. Actions include the following:<br>  ▪ **permit**<br>  ▪ **deny**<br>• **Default rule**—The default rule applied when the identified application is not specified in any rules of the rule set. |
| **Action or Action-type** | • The action taken for a packet that matches the policy's tuples. Actions include the following:<br>  ◦ **permit**<br>  ◦ **firewall-authentication**<br>  ◦ **tunnel ipsec-vpn *vpn-name***<br>  ◦ **pair-policy *pair-policy-name***<br>  ◦ **source-nat pool *pool-name***<br>  ◦ **pool-set *pool-set-name***<br>  ◦ **interface**<br>  ◦ **destination-nat *name***<br>  ◦ **deny**<br>  ◦ **reject**<br>  ◦ **services-offload** |
| **Session log** | Session log entry that indicates whether the **at-create** and **at-close** flags were set at configuration time to log session information. |
| **Scheduler name** | Name of a preconfigured scheduler whose schedule determines when the policy is active and can be used as a possible match for traffic. |

**Table 39: show security policies Output Fields *(continued)***

| Field Name | Field Description |
|---|---|
| **Policy statistics** | • **Input bytes**—The total number of bytes presented for processing by the device.<br>  ◦ **Initial direction**—The number of bytes presented for processing by the device from the initial direction.<br>  ◦ **Reply direction**—The number of bytes presented for processing by the device from the reply direction.<br>• **Output bytes**—The total number of bytes actually processed by the device.<br>  ◦ **Initial direction**—The number of bytes from the initial direction actually processed by the device.<br>  ◦ **Reply direction**—The number of bytes from the reply direction actually processed by the device.<br>• **Input packets**—The total number of packets presented for processing by the device.<br>  ◦ **Initial direction**—The number of packets presented for processing by the device from the initial direction.<br>  ◦ **Reply direction**—The number of packets presented for processing by the device from the reply direction.<br>• **Output packets**—The total number of packets actually processed by the device.<br>  ◦ **Initial direction**—The number of packets actually processed by the device from the initial direction.<br>  ◦ **Reply direction**—The number of packets actually processed by the device from the reply direction.<br>• **Session rate**—The total number of active and deleted sessions.<br>• **Active sessions**—The number of sessions currently present because of access control lookups that used this policy.<br>• **Session deletions**—The number of sessions deleted since system startup.<br>• **Policy lookups**—The number of times the policy was accessed to check for a match. |
| **dynapp-redir-profile** | Displays unified policy redirect profile. See *profile(dynamic-application)*. |
| **Per policy TCP Options** | Configured syn and sequence checks, and the configured TCP MSS value for the initial direction, the reverse direction or, both. |

## Sample Output

**show security policies**

user@host> **show security policies**

```
From zone: trust, To zone: untrust
Policy: p1, State: enabled, Index: 4, Sequence number: 1
  Source addresses:
  sa-1-ipv4: 198.51.100.11/24
  sa-2-ipv6: 2001:db8:a0b:12f0::1/32
  sa-3-ipv6: 2001:db8:a0b:12f0::22/32
  sa-4-wc:   203.0.113.1/255.255.0.255
  Destination addresses:
  da-1-ipv4: 2.2.2.2/24
  da-2-ipv6: 2001:db8:a0b:12f0::8/32
  da-3-ipv6: 2001:db8:a0b:12f0::9/32
  da-4-wc:   192.168.22.11/255.255.0.255
  Source identities: role1, role2, role4
  Applications: any
  Action: permit, application services, log, scheduled
  Application firewall : my_ruleset1
Policy: p2, State: enabled, Index: 5, Sequence number: 2
  Source addresses:
  sa-1-ipv4:  198.51.100.11/24
  sa-2-ipv6:  2001:db8:a0b:12f0::1/32
  sa-3-ipv6:  2001:db8:a0b:12f0::22/32
  Destination addresses:
  da-1-ipv4:  2.2.2.2/24
  da-2-ipv6:  2001:db8:a0b:12f0::1/32
  da-3-ipv6:  2001:db8:a0b:12f0::9/32
  Source identities: role1, role4
  Applications: any
  Action: deny, scheduled
```

**show security policies (Dynamic Applications)**

user@host>**show security policies**

```
Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
  Source addresses: any
  Destination addresses: any
```

```
    Applications: any
    Dynamic Applications: junos:YAHOO
    Action: deny, log
  Policy: p2, State: enabled, Index: 5, Scope Policy: 0, Sequence number: 2
    Source addresses: any
    Destination addresses: any
    Applications: any
    Dynamic Applications: junos:web, junos:web:social-networking:facebook,
    junos:TFTP, junos:QQ
    Action: permit, log
  Policy: p3, State: enabled, Index: 6, Scope Policy: 0, Sequence number: 3
    Source addresses: any
    Destination addresses: any
    Applications: any
    Dynamic Applications: junos:HTTP, junos:SSL
    Action: permit, application services, log
```

The following example displays the output with unified policies configured.

`user@host>` **show security policies**

```
Default policy: deny-all
Pre ID default policy: permit-all
From zone: trust, To zone: untrust
  Policy: p2, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: junos-defaults
    Dynamic Applications: junos:GMAIL, junos:FACEBOOK-CHAT
    dynapp-redir-profile: profile1
```

**show security policies policy-name detail**

`user@host>` **show security policies policy-name p1 detail**

```
  Policy: p1, action-type: permit, State: enabled, Index: 4, Scope Policy: 0
  Description: The policy p1 is for the sales team
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses:
    sa-1-ipv4:  198.51.100.11/24
```

```
    sa-2-ipv6:  2001:db8:a0b:12f0::1/32
    sa-3-ipv6:  2001:db8:a0b:12f0::9/32
    sa-4-wc:    203.0.113.1/255.255.0.255
  Destination addresses:
    da-1-ipv4: 192.0.2.0/24
    da-2-ipv6: 2001:db8:a0b:12f0::1/32
    da-3-ipv6: 2001:db8:a0b:12f0::9/32
    da-4-wc:   192.168.22.11/255.255.0.255
Source identities:
  role1
  role2
  role4
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Destination Address Translation: drop translated
  Application firewall :
  Rule-set: my_ruleset1
    Rule: rule1
      Dynamic Applications: junos:FACEBOOK-ACCESS, junos:YMSG
      Dynamic Application groups: junos:web, junos:chat
      Action: deny
    Default rule: permit
  Session log: at-create, at-close
  Scheduler name: sch20
  Per policy TCP Options: SYN check: No, SEQ check: No
  Policy statistics:
    Input  bytes       :               18144                 545 bps
      Initial direction:                9072                272 bps
      Reply direction  :                9072                 272 bps
    Output bytes       :               18144                 545 bps
      Initial direction:                9072                 272 bps
      Reply direction  :                9072                 272 bps
    Input  packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Output packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Session rate       :                 108                   3 sps
    Active sessions    :                  93
    Session deletions  :                  15
    Policy lookups     :                 108
```

The following example displays the output with unified policies configured.

user@host> **show security policies policy-name p1 detail**

```
Default policy: permit-all
Pre ID default policy: permit-all
From zone: trust, To zone: trust
  Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: any
    Action: reject
    dynapp-redir-profile: profile1
```

**show security policies (Services-Offload)**

user@host> **show security policies**

```
  Policy: p1, action-type: reject, State: enabled, Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: trust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  dynapp-redir-profile: profile1(1)
  Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
```

**show security policies (Device Identity)**

user@host> **show security policies**

```
From zone: trust, To zone: untrust
      Policy: dev-id-marketing, State: enabled, Index: 5, Scope Policy: 0, Sequence
```

```
 number: 1
         Source addresses: any
         Destination addresses: any
         source-end-user-profile: marketing-profile
         Applications: any
         Action: permit
```

**show security policies detail**

user@host> **show security policies detail**

```
Default policy: deny-all
Policy: p1, action-type: permit, services-offload:enabled , State: enabled, Index:
 4, Scope Policy: 0
  Policy Type: Configured
  Description: The policy p1 is for the sales team
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
 Source identities:
  role1
  role2
  role4
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Per policy TCP Options: SYN check: No, SEQ check: No
  Policy statistics:
    Input  bytes       :               18144                 545 bps
      Initial direction:                9072                272 bps
      Reply direction  :                9072                 272 bps
    Output bytes       :               18144                 545 bps
      Initial direction:                9072                 272 bps
      Reply direction  :                9072                 272 bps
    Input  packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
```

```
    Output packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Session rate       :                 108                   3 sps
    Active sessions    :                  93
    Session deletions  :                  15
    Policy lookups     :                 108
Policy: p2, action-type: permit, services-offload:enabled , State: enabled, Index:
 5, Scope Policy: 0
  Policy Type: Configured
  Description: The policy p2 is for the sales team
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
 Source identities:
  role1
  role2
  role4
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Per policy TCP Options: SYN check: No, SEQ check: No
```

The following example displays the output with unified policies configured.

`user@host>` **show security policies detail**

```
Default policy: deny-all
Pre ID default policy: permit-all
Policy: p2, action-type: reject, State: enabled, Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
```

```
  any-ipv4(global): 0.0.0.0/0
  any-ipv6(global): ::/0
Application: junos-defaults
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [443-443]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [5432-5432]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [80-80]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [3128-3128]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [8000-8000]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [8080-8080]
  IP protocol: 17, ALG: 0, Inactivity timeout: 60
    Source port range: [0-0]
    Destination port range: [1-65535]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [443-443]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [5432-5432]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [80-80]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [3128-3128]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [8000-8000]
  IP protocol: 6, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [8080-8080]
  IP protocol: 17, ALG: 0, Inactivity timeout: 60
    Source port range: [0-0]
```

```
      Destination port range: [1-65535]
  Dynamic Application:
    junos:FACEBOOK-CHAT: 10704
    junos:GMAIL: 51
  dynapp-redir-profile: profile1(1)
  Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
```

**show security policies detail (TCP Options)**

user@host> **show security policies policy-name p2 detail**

```
node0:
--------------------------------------------------------------------------
Policy:p2, action-type:permit, State: enabled,Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: trust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Application: junos-defaults
    IP protocol: tcp, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
 Destination port range: [80-80]
Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
  Dynamic-application: junos:HTTP
```

**show security policies policy-name (Negated Address)**

user@host> **show security policies policy-name p1**

```
node0:
--------------------------------------------------------------------------
From zone: trust, To zone: untrust
  Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses(excluded): as1
    Destination addresses(excluded): as2
    Applications: any
    Action: permit
```

**show security policies policy-name detail (Negated Address)**

```
user@host> show security policies policy-name p1 detail
```

```
node0:
--------------------------------------------------------------------------
Policy: p1, action-type: permit, State: enabled, Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses(excluded):
    ad1(ad): 255.255.255.255/32
    ad2(ad):  198.51.100.1/24
    ad3(ad):  198.51.100.6  ~ 198.51.100.56
    ad4(ad):  192.0.2.8/24
    ad5(ad):  198.51.100.99 ~ 198.51.100.199
    ad6(ad):  203.0.113.9/24
    ad7(ad):  203.0.113.23/24
  Destination addresses(excluded):
    ad13(ad2): 198.51.100.76/24
    ad12(ad2): 198.51.100.88/24
    ad11(ad2): 192.0.2.23 ~ 192.0.2.66
    ad10(ad2): 192.0.2.93
    ad9(ad2):  203.0.113.76 ~ 203.0.113.106
    ad8(ad2):  203.0.113.199
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Per policy TCP Options: SYN check: No, SEQ check: No
```

**show security policies global**

```
user@host> show security policies global policy-name Pa
```

```
node0:
--------------------------------------------------------------------------
  Global policies:
  Policy: Pa, State: enabled, Index: 6, Scope Policy: 0, Sequence number: 1
    From zones: any
    To zones: any
    Source addresses: H0
    Destination addresses: H1
    Applications: junos-http
```

```
    Action: permit
```

**show security policies detail tenant**

`user@host>` **show security policies detail tenant TN1**

```
Default policy: deny-all
Pre ID default policy: permit-all
Policy: p1, action-type: permit, State: enabled, Index: 4, Scope Policy: 0
Policy Type: Configured
Sequence number: 1
From zone: trust, To zone: untrust
Source addresses: any
Destination addresses: any
Application: junos-ping
IP protocol: 1, ALG: 0, Inactivity timeout: 60
ICMP Information: type=255, code=0
Application: junos-telnet
IP protocol: tcp, ALG: 0, Inactivity timeout: 1800
Source port range: [0-0]
Destination port range: [23-23]
Application: app_udp
IP protocol: udp, ALG: 0, Inactivity timeout: 1800
Source port range: [0-0]
Destination port range: [5000-5000]
Application: junos-icmp6-all
IP protocol: 58, ALG: 0, Inactivity timeout: 60
ICMP Information: type=255, code=0
Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
Session log: at-create, at-close
Policy statistics:
Input  bytes     :                   0                   0 bps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Output bytes     :                   0                   0 bps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Input  packets   :                   0                   0 pps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Output packets   :                   0                   0 pps
Initial direction:                   0                   0 bps
```

```
Reply direction  :                  0                   0 bps
Session rate     :                  0                   0 sps
Active sessions  :                  0
Session deletions:                  0
Policy lookups   :                  0
```

# show services unified-access-control counters

**Syntax**

```
show services unified-access-control counters
```

**Release Information**

Command introduced in Junos OS Release 12.1X44-D10.

**Description**

Display the number of sessions allowed, denied, and terminated by the Unified Access Control (UAC) service when invoked by a firewall policy with the uac-policy action. Counts are reported for each action taken by UAC. Sessions that were allowed, denied, or terminated by other firewall policy actions are not included in these statistics.

On SRX1500, SRX5400, SRX5600, and SRX5800 devices, UAC counts are grouped and displayed for each PIC on the device. On SRX 300, SRX 320, SRX 340, SRX 345 SRX Series devices, UAC counts are accumulated by device only. There is no PIC specification on these devices.

**Required Privilege Level**

view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

**show services unified-access-control counters on page 784**

**Output Fields**

Table 40 on page 783 lists the output fields for the **show services unified-access-control counters** command. Output fields are listed in the approximate order in which they appear.

**Table 40: show services unified-access-control counters Output Fields**

| Field Name | Field Description |
|---|---|
| **PIC** | If applicable, the number of each PIC implementing UAC. UAC statistics are grouped by PIC. |
| **Sessions allowed** | The sessions permitted by UAC when invoked by a user role firewall policy. |
| **Policy action** | Number of sessions permitted by UAC based on the UAC policy action. |

Table 40: show services unified-access-control counters Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **Timeout action** | Number of sessions permitted by the timeout action while the SRX was disconnected from the UAC device. |
| **Sessions denied** | The sessions denied by UAC when invoked by a user role firewall policy. |
| **Unauthenticated** | Number of sessions denied by UAC because the user was not authenticated. |
| **Policy action** | Number of sessions denied by UAC based on the UAC policy action. |
| **Policy not matched** | Number of sessions denied because no UAC policy match was found. |
| **Timeout action** | Number of sessions denied by the timeout action while the SRX was disconnected from the access control device. |
| **Sessions terminated** | The sessions originally permitted that were later terminated. |
| **Reevaluation** | Number of sessions terminated due to a change in the UAC user roles associated with the session. |
| **Signout** | Number of sessions terminated due to the user signing out. |

## Sample Output

**show services unified-access-control counters**

`user@host>` **show services unified-access-control counters**

```
PIC: fpc2.pic0
  Sessions allowed
    Policy action: 0
    Timeout action: 0
  Sessions denied
    Unauthenticated: 0
    Policy action: 0
    Policy not matched: 0
    Timeout action: 0
  Sessions terminated
    Reevaluation: 0
```

```
    Signout: 0
```

Statistics on SRX 300, SRX 320, SRX 340, and SRX 345 devices are accumulated by device only. There is no PIC specification on these devices.

`user@host>` **show services unified-access-control counters**

```
Sessions allowed
    Policy action: 0
    Timeout action: 0
  Sessions denied
    Unauthenticated: 0
    Policy action: 0
    Policy not matched: 0
    Timeout action: 0
  Sessions terminated
    Reevaluation: 0
    Signout: 0
```

# show services unified-access-control policies

**Syntax**

```
show services unified-access-control policies
```

**Release Information**

Command introduced in Junos OS Release 9.4.

**Description**

Display a summary of resource access policies configured from the IC Series UAC Appliance.

Use this command when you have configured the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series appliance.

**Options**

- **detail**—Display a detailed view of all policies.
- **identifier *id***—Display information about a specific policy by identification number.

**Required Privilege Level**

view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

**show services unified-access-control policies on page 786**
**show services unified-access-control policies detail on page 787**
**show services unified-access-control policies identifier 1 on page 787**

## Sample Output

**show services unified-access-control policies**

```
user@host> show services unified-access-control policies
```

```
Id    Resource                Action Apply        Role identifier
1     10.100.15.0/24:*        allow  selected     1113249951.100616.0
2     10.100.17.0/24:*        deny   all
```

## Sample Output

**show services unified-access-control policies detail**

`user@host>` **show services unified-access-control policies detail**

```
Identifier: 1
  Resource: 10.100.15.0/24:*
  Resource: 10.100.16.23-10.100.16.60:*
  Action: allow
  Apply: selected
  Role identifier       Role name
    1113249951.100616.0 Personal Firewall
    1112927873.881659.0 Antivirus
    1183670148.427197.0 UAC
Identifier: 2
  Resource: 10.100.17.0/24:*
  Resource: 10.100.16.23-10.100.16.60:*
  Resource: 10.100.18.0/24:*
  Action: deny
  Apply: all
```

## Sample Output

**show services unified-access-control policies identifier 1**

`user@host>` **show services unified-access-control policies identifier 1**

```
Identifier: 1
  Resource: 10.100.15.0/24:*
  Resource: 10.100.16.23-10.100.16.60:*
  Action: allow
  Apply: selected
  Role identifier       Role name
    1113249951.100616.0 Personal Firewall
```

```
    1112927873.881659.0 Antivirus
    1183670148.427197.0 UAC
```

# show services unified-access-control roles

**Syntax**

```
show services unified-access-control roles
```

**Release Information**

Command introduced in Junos OS Release 12.1.

**Description**

When implementing user role firewall, display a summary of the roles that have been pushed to the SRX Series device from the access control service.

**Required Privilege Level**

view

RELATED DOCUMENTATION

> *Security Policies User Guide for Security Devices*
>
> Firewall User Authentication Overview | 35

**List of Sample Output**

**show services unified-access-control roles on page 790**

**Output Fields**

Table 41 on page 789 lists the output fields for the **show services unified-access-control roles** command. Output fields are listed in the approximate order in which they appear.

**Table 41: show services unified-access-control roles Output Fields**

| Field Name | Field Description |
|---|---|
| **Name** | Name of the user role. |
| **Identifier** | Unique identifier associated with the specified user role. |
| **Total** | Total number of user roles specified in the table. |

## Sample Output

**show services unified-access-control roles**

```
user@host> show services unified-access-control roles
```

```
Name                                    Identifier
Users                                   0000000001.000005.0
admin-1                                 1420298444.225667.0
Total: 2
```

# show services unified-access-control status

**Syntax**

```
show services unified-access-control status
```

**Release Information**

Command introduced in Junos OS Release 9.4.

**Description**

Display the status of the connection between the SRX Series device and the IC Series UAC Appliance as well as statistics to help debug connections to the IC Series appliance.

Use this command when you have configured the SRX Series device to act as a Junos OS Enforcer in a Unified Access Control (UAC) deployment. When deployed as a Junos OS Enforcer, the SRX Series device enforces the policies that are defined on the UAC's IC Series appliance.

**Required Privilege Level**

view

RELATED DOCUMENTATION

Firewall User Authentication Overview | 35

**List of Sample Output**

**show services unified-access-control status on page 791**

## Sample Output

**show services unified-access-control status**

`user@host>` **show services unified-access-control status**

```
Host          Address         Port   Interface     State
dev106vm26    10.64.11.106    11123  ge-0/0/0.0    connected
dev107vm26    10.64.11.106    11123  ge-0/0/0.0    closed
```

# show services user-identification active-directory-access domain-controller status

**Syntax**

```
show services user-identification active-directory-access domain-controller status
<domain name> <node (node-id | all | local | primary)> <brief | extensive>
```

**Release Information**

Command introduced in Junos OS Release 12.1X47-D10.

**Description**

Display status information for the Active Directory domain controllers configured for the integrated user firewall feature.

**Options**

- **domain *name***—(Optional) Display the status of the domain controllers for a specific domain.
- **node**—(Optional) For chassis cluster configurations, display the status of the domain controllers for a specific node.
  - ***node-id***—Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.
- **brief | extensive**—Display the specified level of output (the default is brief).

**Required Privilege Level**

view

RELATED DOCUMENTATION

active-directory-access | 400

*show services user-identification active-directory-access active-directory-authentication-table*

**List of Sample Output**

show services user-identification active-directory-access domain-controller status on page 793
show services user-identification active-directory-access domain-controller status brief domain on page 794

show services user-identification active-directory-access domain-controller status extensive domain on page 794

**Output Fields**

Table 42 on page 793 lists the output fields for the **show services user-identification active-directory-access domain-controller status** command.

**Table 42: show services user-identification active-directory-access domain-controller Output Fields**

| Field Name | Field Description |
|---|---|
| **Domain controller** | Domain controller name. |
| **Address** | IP address of the domain controller. |
| **Status** | Connection status of the domain controller: connected or disconnected. |
| **Reason** | Reason for a disconnected status: network issue, authentication failed, or host unreachable. |

## Sample Output

**show services user-identification active-directory-access domain-controller status**

Displays brief information for domain controllers in all configured domains.

`user@host>` **show services user-identification active-directory-access domain-controller status**

```
Domain: example-domain-controller.com
    Domain controller       Address         Status
    DC1                   203.0.113.51     Connected
    DC2                   203.0.113.12     Connected
    DC3                   203.0.113.6      Connected
    DC4                   203.0.113.11     Disconnected
    DC5                   203.0.113.7      Disconnected

Domain: example-domain
    Domain controller       Address          Status
    example-domain10          10.1.1.1       Disconnected
    example-domain20          10.2.2.2       Disconnected
    example-domain30          10.3.3.3       Disconnected
```

## Sample Output

**show services user-identification active-directory-access domain-controller status brief domain**

`user@host>` **show services user-identification active-directory-access domain-controller status brief domain example-domain-controller.com**

```
Domain: example-domain-controller.com
    Domain controller       Address           Status
    DC1                   203.0.113.51        Connected
    DC2                   203.0.113.12        Connected
    DC3                   203.0.113.6         Connected
    DC4                   203.0.113.11        Disconnected
    DC5                   203.0.113.7         Disconnected
```

## Sample Output

**show services user-identification active-directory-access domain-controller status extensive domain**

`user@host>` **show services user-identification active-directory-access domain-controller status extensive domain example-domain**

```
Domain: example-domain
  Domain controller: example-domain10
    Address: 10.1.1.1
    Status: Disconnected
    Reason: Network issue
  Domain controller: example-domain20
    Address: 10.2.2.2
    Status: Disconnected
    Reason: Authentication failed
  Domain controller: example-domain30
    Address: 10.3.3.3
    Status: Disconnected
    Reason: Host unreachable
```

# show services user-identification active-directory-access statistics

### Syntax

```
show services user-identification active-directory-access statistics
(ip-user-mapping | ip-user-probe | user-group-mapping) <domain name>
```

### Release Information

Command introduced in Junos OS Release 12.1X47-D10.

### Description

Display statistics about IP address-to-user mapping, user-to-group mapping, and IP user probes used for the integrated user firewall feature. If two domains are configured, output is provided per domain.

### Options

- **ip-user-mapping**—Number of total queries and failed queries to the event log on the domain controller for address-to-user mappings. Includes additional information, such as the log scan interval and the timestamp of the last event read.
- **ip-user-probe**—Number of total PC probes and failed probes.
- **user-group-mapping**—Number of total queries and failed queries to the LDAP server for user-to-group mappings
- **domain *name***—(Optional) Display the statistics for the specified domain.

### Required Privilege Level

view

## RELATED DOCUMENTATION


clear services user-identification active-directory-access | 711

ip-user-mapping | 531

request services user-identification active-directory-access ip-user-probe | 720

user-group-mapping | 663


### List of Sample Output


show services user-identification active-directory-access statistics ip-user-mapping on page 797
show services user-identification active-directory-access statistics ip-user-probe on page 798
show services user-identification active-directory-access statistics user-group-mapping on page 798

### Output Fields

Table 43 on page 796 lists the output fields for the **show services user-identification active-directory-access statistics ip-user-mapping** command.

**Table 43: show services user-identification active-directory-access statistics ip-user-mapping Output Fields**

| Field Name | Field Description |
|---|---|
| **Host** | IP address of the domain controller. |
| **Initial event log timespan** | When the feature is first deployed, the number of previous hours for which the event log on the domain controller is read. A one means the last hour of the event log is read. |
| **Eventlog scan interval** | Number of seconds between event log scans. |
| **Total log query number** | Count of the queries on the event log. |
| **Failed log query number** | Count of the failed queries on the event log. |
| **Log read number** | Count of the times the event log was read. |
| **Latest timestamp** | Year:month:date:hours:minutes:seconds is the timestamp taken from the event log. Timestamp records the latest statistics updated time of the SRX Series devices. |

Table 44 on page 796 lists the output fields for the **show services user-identification active-directory-access statistics ip-user-probe** command.

**Table 44: show services user-identification active-directory-access statistics ip-user-probe Output Fields**

| Field Name | Field Description |
|---|---|
| **Total user probe number** | Count of the probes of IP addresses to get IP address-to-user mappings. |
| **Failed user probe number** | Count of failed probe attempts. |

Table 45 on page 796 lists the output fields for the **show services user-identification active-directory-access statistics user-group-mapping** command.

**Table 45: show services user-identification active-directory-access statistics user-group-mapping Output Fields**

| Field Name | Field Description |
|---|---|
| **Host** | IP address and port being queried. |

**Table 45: show services user-identification active-directory-access statistics user-group-mapping Output Fields *(continued)***

| Field Name | Field Description |
|---|---|
| **Total query number** | Count of queries. |
| **Failed query number** | Count of failed query attempts. |

## Sample Output

**show services user-identification active-directory-access statistics ip-user-mapping**

`user@host>` **show services user-identification active-directory-access statistics ip-user-mapping**

```
Domain: example-domain1.com
     Host: 192.0.2.192
     Initial event log timespan : 1
     Eventlog scan interval : 60
     Total log query number : 240
     Failed log query number : 0
     Log read number : 838
     Latest timestamp :2013-10-11:15:11:54
     Host: 192.0.2.50
     Initial event log timespan : 1
     Eventlog scan interval : 60
     Total log query number : 273
     Failed log query number : 0
     Log read number : 2012
     Latest timestamp :2013-10-11:15:11:23
Domain: example-domain2.com
     Host: 192.0.2.39
     Initial event log timespan : 1
     Eventlog scan interval : 10
     Total log query number : 1596
     Failed log query number : 0
     Log read number : 6691
     Latest timestamp :2013-10-11:15:25:03
     Host: 192.0.2.1
     Initial event log timespan : 1
     Eventlog scan interval : 10
     Total log query number : 2628
     Failed log query number : 0
```

```
    Log read number : 114953
    Latest timestamp :2013-10-11:15:24:01
```

## Sample Output

**show services user-identification active-directory-access statistics ip-user-probe**

`user@host>` **show services user-identification active-directory-access statistics ip-user-probe**

```
Domain: example-domain3.com
Total user probe number : 176116
Failed user probe number : 916
Domain: example-domain3.com
Total user probe number : 17632
Failed user probe number : 342
```

## Sample Output

**show services user-identification active-directory-access statistics user-group-mapping**

`user@host>` **show services user-identification active-directory-access statistics user-group-mapping**

```
Domain: example-domain3.com
Host: 192.0.2.1 Port  389
Total query number : 176116
Failed query number : 916
Domain: example-domain3.com
Host: 192.0.2.5 Port  389
Total query number : 8965
```

# show services user-identification active-directory-access user-group-mapping

### Syntax

```
show services user-identification active-directory-access user-group-mapping
(group name | status | user name) domain name
```

### Release Information

Command introduced in Junos OS Release 12.1X47-D10.

### Description

Display user-to-group mapping information used in the integrated user firewall feature. Note that the LDAP server is often part of the domain controller.

### Options

- **group *group-name***—Display the users mapped to the specified group.
- **status**—Display the status of the last query to the LDAP server for user-group mapping.
- **user *name***—Display the groups for the specified username.
- **domain *name***—(Optional) Display the group, status, or user information for the specified domain.

### Required Privilege Level

view

RELATED DOCUMENTATION

LDAP Functionality in Integrated User Firewall | 194

user-group-mapping | 663

### List of Sample Output

**show services user-identification active-directory-access user-group-mapping group domain on page 801**
**show services user-identification active-directory-access user-group-mapping status on page 801**
**show services user-identification active-directory-access user-group-mapping user on page 801**

### Output Fields

Table 46 on page 800 lists the output fields for the **show services user-identification active-directory-access user-group-mapping group** command.

**Table 46: show services user-identification active-directory-access user-group-mapping group Output Fields**

| Field Name | Field Description |
|---|---|
| **Domain** | Domain of the specified group. |
| **Users** | Usernames mapped to the specified group. |

Table 47 on page 800 lists the output fields for the **show services user-identification active-directory-access user-group-mapping status** command.

**Table 47: show services user-identification active-directory-access user-group-mapping status Output Fields**

| Field Name | Field Description |
|---|---|
| **Domain** | Domain for which the status is displayed. |
| **LDAP server** | IP address of the LDAP server. |
| **Port** | Port number on the LDAP server. |
| **Last-query-status** | Status of the last query from the SRX Series device. |
| **Last-query-time** | Year-month-date:hour:minutes:seconds when the SRX device last queried the LDAP server. |

Table 48 on page 800 lists the output fields for the **show services user-identification active-directory-access user-group-mapping user** command.

**Table 48: show services user-identification active-directory-access user-group-mapping user Output Fields**

| Field Name | Field Description |
|---|---|
| **Domain controller** | Domain controller about which the user information is displayed. |
| **Groups** | Groups to which the user belongs. |
| **Referenced by policy** | Groups to which the user belongs and that are referenced by a firewall policy. |

## Sample Output

**show services user-identification active-directory-access user-group-mapping group domain**

`user@host>` **show services user-identification active-directory-access user-group-mapping group finance domain www.apac-acme.net**

```
show services user-identification active-directory-access user-group-mapping group
 finance-group
Domain: example-domain.net
Users: user1, user2
Domain: example2.domain.net
Users: user3
```

## Sample Output

**show services user-identification active-directory-access user-group-mapping status**

`user@host>` **show services user-identification active-directory-access user-group-mapping status**

```
Domain: example-domain.net
LDAP server   Port      Last-query-status    Last-query-time
192.0.2.87     389       Query success        2014-02-07:15:50:52

Domain: example2.domain.net
LDAP server   Port      Last-query-status    Last-query-time
192.0.2.144    389       Idle                0
```

## Sample Output

**show services user-identification active-directory-access user-group-mapping user**

`user@host>` **show services user-identification active-directory-access user-group-mapping user user1**

```
Domain example-domain.net
Groups: Dev, NAT, SBU
Referenced by policy: SBU
Domain: example2.domain.net
Groups: HR, USA
```

# show service user-identification authentication-source aruba-clearpass user-query counters

**Syntax**

```
show service user-identification authentication-source aruba-clearpass user-query counters
```

**Release Information**
Command introduced in Junos OS Release 12.3X48-D30.

**Description**
Display statistics on the counters maintained by the user query function. The output identifies the ClearPass webserver as the destination of the user query requests. It displays the number of requests sent from the SRX Series device to the ClearPass webserver and the number of responses that the SRX Series device received from it. You can use this command to identify that a problem exists–the number of responses received is less than the number of requests sent.—and then analyze and correct it.

If there are no problems with the communication between the ClearPass Policy Manager (CPPM) and the SRX Series device, the number of requests sent is equal to the number of responses received and the number of error responses.

*number-of-requests = number-of-responses + error-message-responses*

The user query function is part of the SRX Series integrated ClearPass authentication and enforcement feature. The SRX Series device can automatically send requests for individual user authentication and identity information to ClearPass in the event that ClearPass does not post that information to it. For this to occur, you must have configured the user query function.

The SRX Series device exposes to ClearPass a Web API (webapi) that ClearPass uses to send POST request messages to it automatically. These messages contain user authentication and identity information.

The user query function supplements use of the SRX Series Web API function.

**Options**
***authentication-source***—Specify aruba-clearpass to identifies Aruba ClearPass as the authentication source.

**Required Privilege Level**
view

**Output Fields**

- Webserver Address—The IP address of the ClearPass webserver.
- Access token—The token string that the SRX Series device obtains from ClearPass which allows the SRX Series device to query the ClearPass webserver for an individual user's authentication and identity information.
- Requests sent number—A counter that shows the number of individual user authentication information queries that the SRX Series device sent to the ClearPass webserver.
- Total response received number—A counter that shows the number of returns from the ClearPass webserver in response to the individual user authentication information queries that the SRX Series device sent to it. The number of responses should match the number of requests unless an error occurred.
- Error response received number—The number errors that occurred in relation to requests.
- Time of last response—A timestamp showing when the last response from the ClearPass webserver was received.

## Sample Output

**show service user-identification authentication-source aruba-clearpass user-query counters**

`user@host>` **show service user-identification authentication-source aruba-clearpass user-query counters**

```
Web server Address: 4.0.0.20
Access token: 433feffae5c3eb3ff8ffdc49f968b03437ca1ce5
Request sent number: 7
Total response received number: 7
Error response received number: 0
Time of last response: 2000-01-01 11:57:17
```

# show service user-identification authentication-source aruba-clearpass user-query status

**Syntax**

```
show service user-identification authentication-source authentication-source user-query status
```

**Release Information**

Command introduced in Junos OS Release 12.3X48-D30.

**Description**

Checks to determine if the ClearPass webserver is online. The SRX Series device sends user query requests to the ClearPass webserver. The user query function is part of the SRX Series ClearPass Authentication and Enforcement feature.

**Options**

***authentication-source***—Identifies the authentication source. For the integrated ClearPass feature, you must specify the predefined term aruba-clearpass to determine if the ClearPass webserver is online.

**Required Privilege Level**

view

# show services user-identification authentication-table

**Syntax**

```
show services user-identification authentication-table
<authentication-source | counter | ip-address>
show services user-identification authentication-table authentication-source
<active-directory | all | aruba-clearpass | identity-management>
show services user-identification authentication-table authentication-source active-directory
<brief | domain | extensive | group | logical-system | root-logical-system |summary |user>
show services user-identification authentication-table authentication-source all
<brief | domain | extensive |group | logical-system | root-logical-system |summary | user>
<domain domain>
<group (group-name | brief | domain | extensive | logical-system | root-logical-system | summary)>
<logical-system (logical-system-name| all)>
<node (node-id | all | local | primary)>
<root-logical-system (enter |brief | domain | extensive | node)>
<user (user-name | brief | domain | extensive | logical-system | node | root-logical-system | summary)>
show services user-identification authentication-table authentication-source active-directory
<brief | domain | extensive | group | logical-system | root-logical-system |summary |user>
show services user-identification authentication-table authentication-source identity-management source-name
show services user-identification authentication-table authentication-source identity-management tenant
   <tenant-name> extensive
show services user-identification authentication-table counter
show services user-identification authentication-table ip-address
<summary>
<logical-system logical-system-name>
<root-logical-system>
<tenant tenant-name>
<node node-id>
<IP address ip-address>
```

**Release Information**

Command introduced in Junos OS Release 12.

Support for Aruba ClearPass added in Junos OS release 12.3X48-D30.

Support added for identity-management as an authentication source in Junos OS Release 15.1X49-D100.

Support added for logical-system for **authentication-source all** in Junos OS Release 18.2R1.

Support added for tenant system for **authentication-source identity management** in Junos OS Release 19.1R1.

**Description**

Display the user identity information authentication table entries for the specified authentication source. You can display the entire contents of the specified authentication source's authentication table, or you can constrain the displayed information to a specific domain, group, or user based on the user name. You

can also display identity information for a user based on the IP address of the user's device. You can show brief or extensive information for all of these instances.

***authentication-source***—User authentication source whose authentication table or identity management server entries are to be displayed.

Authentication sources include:

**active-directory**—Display the SRX Series active-directory table contents. You can display all of the table's contents or you can delimit the display of user identity information by domain, group, or user name. You can display brief or extensive information for each of these categories.

- domain—Display the entries in the authentication table for the specified domain. You can display summary, group, or user entries for the specified domain.
- group—Display the entries from the authentication table for the specified group.
- user—Display the entries from the authentication table for the specified user based on the user name.

**aruba-clearpass**—Display the SRX Series Aruba ClearPass authentication table contents. You can display all of the table's contents or you can delimit the display of user information by domain, group, or user name. You can display brief or extensive information for each of these categories.

- domain—Display the entries in the authentication table for the specified domain. You can display summary, group, or user entries for the specified domain.
- group—Display the entries from the authentication table for the specified group.
- user—Display the entries from the authentication table for the specified user based on the user name.

**identity-management** —Display user identity entries contained in the identity-management authentication system.

- source-name—Name of the identity -management source. This could be the Juniper Identity Management Service (JIMS) or any third-party authentication source.
  - If you specify a source, such as "JIMS – Active Directory" for Juniper Identity Management Service, the SRX Series device will show entries only for that authentication source.

    Possible values include:

    - For JIMS: "JIMS – Active Directory", "JIMS – Exchange"
    - For ClearPass: "Aruba ClearPass"
- domain—Display the entries in the identity management system for the specified domain. You can display summary, group, or user entries for the specified domain.

- group—Display the entries in the identity management system for the specified group.
- user—Display the entries in the identity management system for the specified user based on the user name.
- tenant—Display the entries in the identity management system for the specified tenant system.

**Options**

- **all**—Summary of the authentication entry information for all entries.
- **group *group-name***—Entries from the authentication table or identity management system for the specified group.
- **ip-address *ip-address***—Entries from the authentication table or identity management system for the specified IP address.
- **user *name***—Entries from the authentication table for the specified username.
- **domain *name***—Summary, group, or user entries for the specified domain.
- **node**—(Optional) For chassis cluster configurations, the summary, IP address, or user entries for a specific node.
  - ***node-id***—Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.
- **brief | extensive**—Display the specified level of output (the default is brief).
- **logical-system**—Display the authentication entries based on the logical system name.
- **root-logical-system**—Display the authentication entries based on the root logical system.
- **tenant *tanant-name***—Display the authentication entries based on the specified tenant system name.

**Required Privilege Level**

view

**List of Sample Output**

show services user-identification active-directory-access active-directory-authentication-table ip-address on page 809
show services user-identification authentication-table ip-address on page 810
show services user-identification active-directory-access active-directory-authentication-table all on page 810
show services user-identification active-directory-access active-directory-authentication-table all extensive on page 811

show services user-identification active-directory-access active-directory-authentication-table all domain on page 812
All Authentication Sources on page 812
show services user-identification authentication-table authentication-source all all-logical-systems-tenants on page 815
Aruba ClearPass on page 816
show services user-identification authentication-table authentication-source aruba-clearpass domain brief on page 818
show services user-identification authentication-table authentication-source aruba-clearpass extensive on page 818
show services user-identification authentication-table authentication-source identity-management brief on page 821
show services user-identification authentication-table authentication-source identity-management extensive on page 821
show services user-identification authentication-table authentication-source all extensive on page 822
show services user-identification authentication-table authentication-source identity-management brief on page 822
show services user-identification authentication-table authentication-source identity-management extensive on page 823
show services user-identification authentication-table authentication-source identity-management tenant tn1 extensive on page 824
show services user-identification authentication-table authentication-source all extensive on page 825

**Output Fields**

| Field Name | Field Description |
|---|---|
| Domain | Name of the domain that the users belong to. User identity and authentication information is display for all users who belong to the domain and for whom there are entries in the specified authentication source table or repository. |
| Total entries | Number of user entries in the authentication table, by domain. |
| For each entry: | |
| Source IP | The IP address of the user's device. If a user is logged in to the network with more than one device, a separate entry is created for the user for each device. It showing the devices IP address. |
| Username | The name by which the user is logged in to the network. |
| Groups | A list of the groups that the user belongs to. The list can include a group that identifies the device posture. |

| Field Name | Field Description |
|---|---|
| State | The state of the entry. There are four states for an authentication entry: initial, valid, invalid, and pending.<br>• An initial state is a temporary state, and it can be created from either a valid or an invalid entry.<br>The entry had not been pushed to the Packet Forwarding Engine.<br>• A valid state indicates that the authentication entry has a valid IP address, domain, and username.<br>The authentication entry is pushed to the Packet Forwarding Engine.<br>• An invalid state indicates that the entry does not have a valid IP address, domain, and username. If the entry is invalid, it is put in the null domain.<br>• A pending state indicates that the entry was created after the user query was sent and before the response was received. The IP address is being probed. |
| Source | Authentication source. |
| Access start date | The date when the authentication entry was created by the SRX Series device. |
| Access start time | The time when the authentication entry was created by the SRX Series device. |
| Last updated timestamp | The time when the user information was created. This value is taken from the timestamp field in the user information. |
| Age time | The time, in minutes, after which the entry expires, as configured by the authentication-entry-timeout statement. If a value of 0 was specified, the entry never expires. |
| Forced Age time | The rest value and the forced value.<br>This information is made available if you configure the firewall-authentication-forced-timeout statement for active directory. |

## Active Directory

**show services user-identification active-directory-access active-directory-authentication-table ip-address**

Output of this command displays authentication and identity information for a specific user based on the IP address of the user's device.

user@host> **show services user-identification active-directory-access active-directory-authentication-table ip-address 198.51.100.3.**

```
Domain: ad.example.net
 Source-ip: 198.51.100.3
  Username: user1
  Groups:group1
  State: Valid
  Source: wmic
  Access start date: 2014-03-10
  Access start time: 13:59:56
  Age time: 1437
```

**show services user-identification authentication-table ip-address**

Output of this command displays authentication and identity information for a specific user based on the IP address of the user's device.

user@host> **show services user-identification authentication-table ip-address 2001:db8::1:1**

```
Domain: ac.example.net
 Source-ip: 2001:db8::1:1
  Username: user1
  Groups:group1
  State: Valid
  Source: wmic
  Access start date: 2017-05-10
  Access start time: 13:59:56
  Age time: 1437
```

**show services user-identification active-directory-access active-directory-authentication-table all**

Output of this command displays user authentication and identity information for all users for whom there are entries in the active directory authentication table.

user@host> **show services user-identification active-directory-access active-directory-authentication-table all**

```
Domain: www.engineering-example.net
Total count: 2
Source IP       Username      Groups          State
198.51.100.22     u2           r1, r3, r4      initial
198.51.100.23     u3           r5, r6, r4      pending
```

```
Domain: www.hr-example.net
Total count: 2
Source IP       Username      Groups          State
 198.51.100.26    u4           r1, r3, r4      initial
 198.51.100.27    u5           r5, r6, r4      pending
```

**show services user-identification active-directory-access active-directory-authentication-table all extensive**

Output of this command, which specifies the **extensive** option, shows state and access information for all entries in the active directory authentication table, in addition to basic information displayed when the **brief** option is used and by default.

`user@host>` **show services user-identification active-directory-access active-directory-authentication-table all extensive**

```
Domain: www.mycompany-example.com
Total entries: 2

Source IP: 198.51.100.29
Username: u2
Groups: r1, r3, r4
State: inital
Access start date: 2013-05-22
Access start time: 10:56:58
Age time: 20 min

Source IP: 198.51.100.30
Username: u3
Groups: r5, r6, r4
State: pending
Access start date: 2013-05-22
Access start time: 10:56:58
Age time: 20 min

Domain: www.hr-example.net
Total entries: 2

Source IP: 198.51.100.31
Username: u2
Groups: r1, r3, r4
State: inital
Access start date: 2013-05-22
```

```
Access start time: 10:56:58
Age time: 20 min

Source IP: 198.51.100.32
Username: u3
Groups: r5, r6, r4
State: pending
Access start date: 2013-05-22
Access start time: 10:56:58
Age time: 20
```

**show services user-identification active-directory-access active-directory-authentication-table all domain**

Output of this command shows by default brief user identity and authentication information for all users for whom there are entries in the active directory authentication table and whose devices belong to the specified domain.

`user@host>` **show services user-identification active-directory-access active-directory-authentication-table all domain www.mydomain-example.com**

```
Domain: www.mydomain-example.com
Total count: 2
Source IP       Username      Groups          State
 198.51.100.36    u2          r1, r3, r4      initial
 198.51.100.37    u3          r5, r6, r4      pending
```

**All Authentication Sources**

Output of this command shows extensive user identity and authentication information for all users with entries in authentication tables of any authentication source. This example shows only one entry to illustrate the content that is displayed with the extensive option.

`user@host>` **show services user-identification authentication-table authentication-source all extensive**

```
Domain: ad-userfw-example.net
       Total entries: 1
       Source-ip: 198.51.100.1/24
       Username: administrator
       State: Valid
       Source: firewall-authentication
       Access start date: 2016-10-27
```

```
       Access start time: 09:30:27
       Age time: 30
```

`user@host>` **show services user-identification authentication-table authentication-source all logical-system**

```
lsys1
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad2012.jims.com
Total entries: 18003
Source IP       Username       groups(Ref by policy)          state
bbbb:bbbb:bbbb: jimsuser18000                                 Valid
bbbb:bbbb:bbbb: jimsuser17999                                 Valid
bbbb:bbbb:bbbb: jimsuser17998                                 Valid
bbbb:bbbb:bbbb: jimsuser17997                                 Valid
bbbb:bbbb:bbbb: jimsuser17996                                 Valid
bbbb:bbbb:bbbb: jimsuser17995                                 Valid
bbbb:bbbb:bbbb: jimsuser17994                                 Valid
bbbb:bbbb:bbbb: jimsuser17993                                 Valid
```

`user@host>` **show services user-identification authentication-table authentication-source all root-logical-system**

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad2012.jims.com
Total entries: 18003
Source IP       Username       groups(Ref by policy)          state
bbbb:bbbb:bbbb: jimsuser10745
bbbb:bbbb:bbbb: jimsuser18000                                 Valid
bbbb:bbbb:bbbb: jimsuser17999                                 Valid
bbbb:bbbb:bbbb: jimsuser17998                                 Valid
bbbb:bbbb:bbbb: jimsuser17997                                 Valid
bbbb:bbbb:bbbb: jimsuser17996                                 Valid
bbbb:bbbb:bbbb: jimsuser17995                                 Valid
```

```
bbbb:bbbb:bbbb: jimsuser17994                                  Valid
bbbb:bbbb:bbbb: jimsuser17993                                  Valid
bbbb:bbbb:bbbb: jimsuser17992                                  Valid
user@host> show services user-identification authentication-table
authentication-source all node 0
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad2012.jims.com
Total entries: 18003
Source IP       Username       groups(Ref by policy)          state
bbbb:bbbb:bbbb: jimsuser14716
bbbb:bbbb:bbbb: jimsuser18000                                  Valid
bbbb:bbbb:bbbb: jimsuser17999                                  Valid
bbbb:bbbb:bbbb: jimsuser17998                                  Valid
bbbb:bbbb:bbbb: jimsuser17997                                  Valid
bbbb:bbbb:bbbb: jimsuser17996                                  Valid
bbbb:bbbb:bbbb: jimsuser17995                                  Valid
bbbb:bbbb:bbbb: jimsuser17994                                  Valid
bbbb:bbbb:bbbb: jimsuser17993                                  Valid
```

`user@host>` **show services user-identification authentication-table authentication-source all node 0 logical-system lsys1**

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad2012.jims.com
Total entries: 18003
Source IP       Username       groups(Ref by policy)          state
bbbb:bbbb:bbbb: jimsuser18000                                  Valid
bbbb:bbbb:bbbb: jimsuser17999                                  Valid
bbbb:bbbb:bbbb: jimsuser17998                                  Valid
bbbb:bbbb:bbbb: jimsuser17997                                  Valid
bbbb:bbbb:bbbb: jimsuser17996                                  Valid
bbbb:bbbb:bbbb: jimsuser17995                                  Valid
bbbb:bbbb:bbbb: jimsuser17994                                  Valid
```

```
bbbb:bbbb:bbbb: jimsuser17993                                  Valid
bbbb:bbbb:bbbb: jimsuser17992                                  Valid
```

user@host> **show services user-identification authentication-table authentication-source all node 0**

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad2012.jims.com
Total entries: 18003
Source IP       Username       groups(Ref by policy)          state
bbbb:bbbb:bbbb: jimsuser1213
bbbb:bbbb:bbbb: jimsuser18000                                  Valid
bbbb:bbbb:bbbb: jimsuser17999                                  Valid
bbbb:bbbb:bbbb: jimsuser17998                                  Valid
bbbb:bbbb:bbbb: jimsuser17997                                  Valid
bbbb:bbbb:bbbb: jimsuser17996                                  Valid
bbbb:bbbb:bbbb: jimsuser17995                                  Valid
bbbb:bbbb:bbbb: jimsuser17994                                  Valid
bbbb:bbbb:bbbb: jimsuser17993                                  Valid
```

**show services user-identification authentication-table authentication-source all all-logical-systems-tenants**

Output of this command displays brief user authentication and identity information for all users for whom there are entries in the identity-management authentication source.

user@host> **show services user-identification authentication-table authentication-source all all-logical-systems-tenants**

```
node0:
--------------------------------------------------------------------------
Logical System: ld1
Domain: ad03.net
Total entries: 4
Source IP       Username       groups(Ref by policy)          state
12.0.0.2        administrator  posture-healthy                Valid
12.0.0.15       administrator  posture-healthy                Valid
3000::5         N/A            posture-healthy                Valid
2001:db8:::302b N/A        posture-healthy               Valid

Logical System: tn1
```

```
Domain: ad03.net
Total entries: 4
Source IP       Username       groups(Ref by policy)          state
12.0.0.2        administrator  posture-healthy                Valid
12.0.0.15       administrator  posture-healthy                Valid
3000::5         N/A            posture-healthy                Valid
2001:db8:::302b N/A            posture-healthy                Valid
```

**Aruba ClearPass**

**show services user-identification authentication-table authentication-source aruba-clearpass domain extensive**

Output of this command shows extensive user identity and authentication information, when Aruba ClearPass is used as the authentication source, for all users whose devices belong to the GLOBAL domain.

`user@host>` **show services user-identification authentication-table authentication-source aruba-clearpass domain GLOBAL extensive**

```
Domain: GLOBAL
Total entries: 7
  Source-ip: 203.0.113.21
    Username: vikiyr
    Groups:posture-healthy, accounting-grp, accounting-grp-and-company-device,
    corporate-limited, [user authenticated]
    Groups referenced by policy:accounting-grp-and-company-device,
    corporate-limited
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:20:30
    Last updated timestamp: 2015-12-22 04:02:48
    Age time: 0
  Source-ip: 203.0.113.89
    Username: abewhfy
    Groups:posture-unknown, marketing-access-limited-grp, [user authenticated]
    Groups referenced by policy:marketing-access-limited-grp
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:31:40
    Last updated timestamp: 2015-12-22 04:18:48
    Age time: 0
  Source-ip: 203.0.113.52
```

```
  Username: jjxchan
  Groups:posture-healthy, marketing-access-for-pcs-limited-group,
  marketing-general, sales-limited, corporate-limited, [user authenticated]
  Groups referenced by policy:marketing-access-for-pcs-limited-group,
  corporate-limited
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:22:48
  Last updated timestamp: 2015-12-22 05:46:21
  Age time: 0
Source-ip: 203.0.113.53
  Username: ltchen1
  Groups:posture-healthy, human-resources-grp, accounting-limited,
  corporate-limited, [user authenticated]
  Groups referenced by policy:corporate-limited
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:21:37
  Last updated timestamp: 2015-12-22 05:41:18
  Age time: 0
Source-ip: 203.0.113.54
  Username: guest1
  Groups:posture-healthy, guest, [user authenticated]
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:23:10
  Last updated timestamp: 2015-12-22 05:50:47
  Age time: 0
Source-ip: 203.0.113.55
  Username: guest2
  Groups:posture-healthy, guest-device-byod, [user authenticated]
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:23:21
  Last updated timestamp: 2015-12-22 05:52:44
  Age time: 0
Source-ip: 2001:db8:4136:e378:8000:63bf:3fff:fdd2
  Username: guest3
  Groups:posture-healthy, guest-device-grp, [user authenticated]
  State: Valid
```

```
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:23:21
    Last updated timestamp: 2015-12-22 05:52:44
    Age time: 0
```

**show services user-identification authentication-table authentication-source aruba-clearpass domain brief**

Output of this command shows brief user identity and authentication information for users whose devices belong to the GLOBAL domain.

If you do not specify brief, the same information would be displayed. The default behavior is to show brief output.

`user@host>` **show services user-identification authentication-table authentication-source aruba-clearpass domain GLOBAL brief**

```
Domain: GLOBAL
Total entries: 6
Source IP                               Username       groups(Ref by policy)
      state
203.0.113.71                            taviki2
accounting-grp-and-company-dev  Valid
203.0.113.89                            gabewb1
marketing-access-limited-grp    Valid
203.0.113.92                            tljxchan
marketing-access-for-pcs-limit  Valid
203.0.113.93                            tjlchen1         corporate-limited
         Valid
203.0.113.94                            guest1
         Valid
203.0.113.95                            guest2
         Valid
2001:db8:4136:e378:8000:63bf:3fff:fdd2   guest2
         Valid
```

**show services user-identification authentication-table authentication-source aruba-clearpass extensive**

Output of the following command shows extensive user identity and authentication information for all users authenticated by Aruba ClearPass for whom entries exist in the aruba-clearpass authentication table.

user@host> **show services user-identification authentication-table authentication-source aruba-clearpass extensive**

```
Domain: GLOBAL
Total entries: 7
  Source-ip: 203.0.113.31
    Username: vjki2
    Groups:posture-healthy, accounting-grp, accounting-grp-and-company-device,
    corporate-limited, [user authenticated]
    Groups referenced by policy:accounting-grp-and-company-device,
    corporate-limited
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:20:30
    Last updated timestamp: 2015-12-22 04:02:48
    Age time: 0
  Source-ip: 203.0.113.89
    Username: labew11
    Groups:posture-unknown, marketing-access-limited-grp, [user authenticated]
    Groups referenced by policy:marketing-access-limited-grp
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:31:40
    Last updated timestamp: 2015-12-22 04:18:48
    Age time: 0
  Source-ip: 203.0.113.62
    Username: dxchan45
    Groups:posture-healthy, marketing-access-for-pcs-limited-group,
    marketing-general, sales-limited, corporate-limited, [user authenticated]
    Groups referenced by policy:marketing-access-for-pcs-limited-group,
    corporate-limited
    State: Valid
    Source: Aruba ClearPass
    Access start date: 2016-03-08
    Access start time: 17:22:48
    Last updated timestamp: 2015-12-22 05:46:21
    Age time: 0
  Source-ip: 2001:db8:4136:e378:8000:63bf:3fff:fdd2
    Username: efchan47
    Groups:posture-healthy, marketing-access-for-pcs-limited-group,
    marketing-general, sales-limited, corporate-limited, [user authenticated]
```

```
  Groups referenced by policy:marketing-access-for-pcs-limited-group,
  corporate-limited
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:22:48
  Last updated timestamp: 2015-12-22 05:46:21
  Age time: 0
Source-ip: 203.0.113.83
  Username: ljhen1
  Groups:posture-healthy, human-resources-grp, accounting-limited,
  corporate-limited, [user authenticated]
  Groups referenced by policy:corporate-limited
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:21:37
  Last updated timestamp: 2015-12-22 05:41:18
  Age time: 0
Source-ip: 203.0.113.34
  Username: guest1
  Groups:posture-healthy, guest, [user authenticated]
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:23:10
  Last updated timestamp: 2015-12-22 05:50:47
  Age time: 0
Source-ip: 203.0.113.95
  Username: guest2
  Groups:posture-healthy, guest-device-byod, [user authenticated]
  State: Valid
  Source: Aruba ClearPass
  Access start date: 2016-03-08
  Access start time: 17:23:21
  Last updated timestamp: 2015-12-22 05:52:44
  Age time: 0
```

## Identity Management

**show services user-identification authentication-table authentication-source identity-management brief**

Output of this command displays brief user authentication and identity information for all users for whom there are entries in the identity-management authentication source.

`user@host>` **show services user-identification authentication-table authentication-source identity-management brief**

```
Domain: ad-domaine-example.net
Total entries: 5
Source IP       Username       groups(Ref by policy)          state
198.51.100.63    N/A                                          Valid
203.0.113.30     administrator                                Valid
203.0.113.18     N/A                                          Valid
198.51.100.69    N/A                                          Valid
198.51.100.66    administrator                                Valid

Domain: NULL
Total entries: 1
Source IP       Username       groups(Ref by policy
```

**show services user-identification authentication-table authentication-source identity-management extensive**

Output of this command displays extensive user authentication and identity information for all users for whom there are entries in the identity-management authentication source.

`user@host>` **show services user-identification authentication-table authentication-source identity-management extensive**

```
Domain: ad-domain2-example.net
Total entries: 5
  Source-ip: 198.51.100.63
    Username: N/A
    Groups:posture-healthy
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-06-05
    Access start time: 09:28:45
    Last updated timestamp: 2017-06-06 08:41:56
    Age time: 0
  Source-ip: 198.51.100.66
    Username: administrator
```

```
   Groups:posture-healthy, group policy creator owners, enterprise admins, schema
admins, domain admins,
    administrators, denied rodc password replication group
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-06-05
    Access start time: 09:23:44
    Last updated timestamp: 2017-06-06 08:11:45
    Age time: 0
```

**show services user-identification authentication-table authentication-source all extensive**

Output of this command, which specifies the extensive option, shows state and access information for all entries.

`user@host>` **show services user-identification authentication-table authentication-source identity-management extensive**

```
Domain: jims-dom1.local
Total entries: 1
Source-ip: 2001:db8:4136:e378:8000:63bf:3fff:fdd2
Username: user1
Groups:posture-healthy
Groups referenced by policy:posture-healthy
State: Valid
Source: JIMS - Active Directory
Access start date: 2017-08-23
Access start time: 15:06:32
Last updated timestamp: 2017-06-07 02:50:10
Age time: 30
```

## Identity Management

**show services user-identification authentication-table authentication-source identity-management brief**

Output of this command displays brief user authentication and identity information for all users for whom there are entries in the identity-management authentication source.

`user@host>` **show services user-identification authentication-table authentication-source identity-management brief**

```
Domain: ad-domaine-example.net
Total entries: 5
Source IP       Username       groups(Ref by policy)          state
198.51.100.63    N/A                                          Valid
203.0.113.30     administrator                                Valid
203.0.113.18     N/A                                          Valid
198.51.100.69    N/A                                          Valid
198.51.100.66    administrator                                Valid

Domain: NULL
Total entries: 1
Source IP       Username       groups(Ref by policy
```

**show services user-identification authentication-table authentication-source identity-management extensive**

Output of this command displays extensive user authentication and identity information for all users for whom there are entries in the identity-management authentication source.

`user@host>` **show services user-identification authentication-table authentication-source identity-management extensive**

```
Domain: ad-domain2-example.net
Total entries: 5
  Source-ip: 198.51.100.63
    Username: N/A
    Groups:posture-healthy
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-06-05
    Access start time: 09:28:45
    Last updated timestamp: 2017-06-06 08:41:56
    Age time: 0
  Source-ip: 198.51.100.66
    Username: administrator
    Groups:posture-healthy, group policy creator owners, enterprise admins, schema
 admins, domain admins,
    administrators, denied rodc password replication group
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-06-05
    Access start time: 09:23:44
    Last updated timestamp: 2017-06-06 08:11:45
    Age time: 0
```

**show services user-identification authentication-table authentication-source identity-management tenant tn1 extensive**

Output of this command, which specifies the extensive option, shows state and access information for all entries.

`user@host>` **show services user-identification authentication-table authentication-source identity-management tenant tn1 extensive**

```
node0:
--------------------------------------------------------------------------
Logical System: root-logical-system

Domain: ad03.net
Total entries: 4
  Source-ip: 12.0.0.15
    Username: administrator
    Groups:posture-healthy, admin, group policy creator owners, domain admins,
enterprise admins, schema admins, administrators, denied rodc password replication
 group
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-12-05
    Access start time: 09:36:30
    Last updated timestamp: 2017-12-04 15:45:51
    Age time: 0
  Source-ip: 3000::12
    Username: jasonlee
    Groups:posture-healthy, domain users, users, group1
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-12-05
    Access start time: 09:36:30
    Last updated timestamp: 2017-12-04 15:46:46
    Age time: 0
  Source-ip: 3000::5
    Username: N/A
    Groups:posture-healthy
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-12-05
    Access start time: 09:36:30
    Last updated timestamp: 2017-12-04 16:01:18
    Age time: 0
  Source-ip: fe80::342c:302b:6cb4:e109
    Username: N/A
```

```
    Groups:posture-healthy
    State: Valid
    Source: JIMS - Active Directory
    Access start date: 2017-12-05
    Access start time: 09:36:30
Last updated timestamp: 2017-12-04 16:01:14
Age time: 0
```

## Firewall Authentication Forced Age Timeout

Output shows the “Forced Age timeout” value is displayed when the firewall authentication forced timeout function is configured, but only for when the extensive option is used. The value shows the remaining time left based on the forced timeout setting.

**show services user-identification authentication-table authentication-source all extensive**

```
user@host> show services user-identification authentication-table
authentication-source all extensive
```

```
Domain: ad-userfw.net
Total entries: 1
  Source-ip: 198.51.100.98
    Username: administrator
    State: Valid
    Source: firewall-authentication
    Access start date: 2016-10-27
    Access start time: 09:30:27
    Age time: 30
    Forced Age time:  30/180
```

# show service user-identification identity-management

**Syntax**

```
show service user-identification identity-management (counter | status)
```

**Release Information**

Command introduced in Junos OS Release 15.1X49-D100 for identity-management as the authentication source.

**Description**

Display statistical data about the advanced user query function batch queries and IP queries, or show status on the Juniper Identity Management Service servers.

**Options**

The following information is displayed for the primary server and the secondary server separately.

**counter**—Display counters for batch and IP queries send to the Juniper Identity Management Service device an responses received from the Juniper Identity Management Service server. This is displayed separately for the primary server and the secondary server, if more than one is configured.

**status**—Verify that the Juniper Identity Management Service server is online and which server is responding to queries from the SRX Series device.

**Required Privilege Level**

view

RELATED DOCUMENTATION

Understanding Advanced Query Feature for Obtaining User Identity Information from JIMS | 275

primary | 575

**List of Sample Output**

**show service user-identification identity-management counter on page 827**

**show service user-identification identity-management status on page 828**

**Output Fields**

| | |
|---|---|
| Access token | Token string |
| Batch queries sent number | A number indicating how many batch queries the SRX Series device sent to the Juniper Identity Management Service server. |

| | |
|---|---|
| Batch queries Response received number | A number indicating how many responses the SRX Series device received from the Juniper Identity Management Service server in response to its batch queries. |
| Time of last response | Timestamp indicating when the last response was received. |
| IP queries sent number | A number indicating how many IP queries the SRX Series device sent to the Juniper Identity Management Service server. |
| IP queries Response received number | A number indicating how many responses the SRX Series device received from the Juniper Identity Management Service server in response to its IP queries. |
| Primary server address | For the status option, the IP address of the primary server. |
| Secondary server address | For the status option, the IP address of the secondary server. |
| Current working server | The Juniper Identity Management Service server that is responding to SRX Series queries. |

## Sample Output

**show service user-identification identity-management counter**

`user@host>` **show service user-identification identity-management counter**

```
Primary server Address:
         Access token:   token-string
         Batch queries sent number: counter
         Batch queries Response received number: counter
        Time of last response: timestamp timestamp /* when received last response
 */
         IP queries sent number:  counter
         IP queries Response received number: counter
       Secondary Server
         Access token:   token-string
         Batch queries sent number: counter
         Batch queries Response received number: counter
        Time of last response: timestamp timestamp /* when received last response
 */
         IP queries sent number:  counter
```

```
         IP queries Response received number: counter
```

**show service user-identification identity-management status**

user@host> **show service user-identification identity-management status**

```
Primary server Address:  iP-address
          Status: Online
       Secondary server Address: iP-address
          Status: Offline
       Current working server:
          Primary server
```

# show services user-identification device-information table

**Syntax**

```
show services user-identification device-information table all (brief | domain | extensive) | device-id device-id ( brief
    |domain |extensive) | ip-address ip-address
```

**Release Information**

Statement introduced in Junos OS Release 15.1X49-D70.

**Description**

Display the contents of the device identity authentication table. The device identity authentication table includes entries for authenticated devices whose information is obtained from external authentication sources. A device identity entry contains the device's IP address, the device ID, and a list of groups that the device belongs to. It also contains attributes that are configured in the device identity profile—for example, the type of device, the vendor, and the operating system that is running on the device and its version.

The device identity authentication table is separate from the Active Directory authentication table or any other local authentication table that is used for Junos OS features, or for specific third-party authentication sources. Also, unlike local user authentication tables, which are specific to an authentication source, the device identity authentication table holds device identity information for devices authenticated by different sources.

Only one authentication source, such as Active Directory, can be active at a time. A result of this requirement is that there is less demand on the system to process information.

**Options**

**all**—Display information for all authenticated devices with entries in the table.

***device-id***—Display information for the authenticated device whose device ID is specified.

***ip-address***—Display information for the authenticated device whose IP address is specified.

**brief**—Display terse information for the entries in the device identity authentication table entries. You can specify **brief** as a keyword to the parameters **all** and **device-id**.

**domain** —Display the name of domain and information for all authenticated devices that belong to the domain. You can specify **domain** as a keyword to the parameters **all** and **device-id**.

**extensive**—Display**extensive** information for all of the authenticated devices for which there are table entries. It displays the domain name, the IP address of the device, the device's ID, the device category and vendor, the device type, and the operating system running on the device and its version.

**Required Privilege Level**
view

## RELATED DOCUMENTATION

- Understanding the Device Identity Attributes and Profiles for the Integrated User Firewall Device Identity Authentication Feature | 247
- Understanding the Device Identity Authentication Table and Its Entries | 252
- Understanding Access Control to Network Resources Based on Device Identity Information | 244
- authentication-source (Services User Identification Device Identity) | 424
- source-end-user-profile | 614

**Output Fields**

**Table 49: show services user-identification device-information table Output Fields**

| Field Name | Field Description |
|---|---|
| **Domain name** | The name of the domain to which the devices belong. |
| **NOTE:** For each authenticated device, the following information is displayed when the parameter **all** is specified after **table** and it is modified by the keyword **extensive**. | |
| **Source IP address** | The IP address of the device. |
| Device ID | The ID assigned to the device. |
| Device-Groups | The groups to which the device belongs. |
| device-category | The kind of device. For example, the device might be a laptop. You configured this value as part of the device identity profile. |
| device-vendor | The maker of the device. For example, the device vendor might be Lenovo. |
| device-type | The device type. If this device is a laptop made by Lenovo, it might be of type thinkpad-t430. |
| device-os | The operating system that is running on the device. The operating system might be Windows. |
| device-os-version | The version of the operating system running on the device. For example, for Windows, this might be 7.1. |

**Table 49: show services user-identification device-information table Output Fields *(continued)***

| Field Name | Field Description |
|---|---|
| Location1 | The location where the device is being used. The location might be specified as United States. |
| Referred by | The security policy that refers to the device in its **source-end-user-profile** field. The **source-end-user-profile** that you configure might pertain to a group of devices or a single device. |

## Sample Output

**show services user-identification device-information table**

`user@host>` **show services user-identification device-information table all extensive**

```
Domain: example.net
Total entries: 3
  Source IP:192.0.2.11
    Device ID: dev01
   Device-Groups: device_group01, device_group02, device_group03, device_group04,
 device_group05
    device-category: laptop
    device-vendor: lenovo
    device-type: thinkpad-t430
    device-os: windows
    device-os-version: 7.1
    Location1: us1
    Referred by: My-pf_0
  Source IP: 192.0.2.12
    Device ID: dev02
   Device-Groups: device_group06, device_group07, device_group08, device_group09,
 device_group10
    device-category: laptop
    device-vendor: lenovo
    device-type: thinkpad-t430
    device-os: windows
    device-os-version: 7.1
    Location1: us1
    Referred by: My-pf_0
  Source IP: 192.0.2.14
    Device ID: dev03
```

```
   Device-Groups: device_group01, device_group02, device_group03, device_group04,
device_group05
   device-category: laptop
   device-vendor: lenovo
   device-type: thinkpad-t430
   device-os: windows
   device-os-version: 7.1
   Location1: us1
   Referred by: My-pf_0
```

user@host> **show services user-identification device-information table all**

```
Domain: example.net
Total entries: 1
Source IP       Device ID      Device-Groups
2001:db8::1:1   dev04          device-group08
```

**show services user-identification device-information table all extensive**

user@host> **show services user-identification device-information table all extensive**

```
Domain: jims-dom1.local
Total entries: 1
Source IP: 2001:db8:4136:e378:8000:63bf:3fff:fdd2
Device ID: win-test$
Device-Groups: dev, pre-windows 2000 compatible access, cert publishers,
denied rodc password replication group
device-os: windows server 2012 r2 standard evaluation
device-os-version: 6.3 (9600)
Referred by: p1
```

**show services user-identification device-information table all**

user@host> **show services user-identification device-information table all**

```
example.net
Total entries: 1
Source IP                               Device ID Device-Groups
2001:db8:4136:e378:8000:63bf:3fff:fdd2  dev04     device-group08
```