Application-Aware Access List

Published: 2013-08-29

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

Application-Aware Access List

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

List of Tables

About the Documentation

Documentation and Release Notes

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

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

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

Supported Platforms

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

- MX Series

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 the *CLI User Guide*.

Documentation Conventions

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

Table 1: Notice Icons

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

Table 2 on page ix 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: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • 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: [edit] 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. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose 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** **(*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)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • 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, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at

https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

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

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

Application-Aware Access List

AACL Overview

NOTE: Starting with Junos OS Release 12.1, all interface-style services are supported for dynamic Point-to-Point Protocol over Ethernet (PPPoE) subscribers on all MX Series routers with modular Modular Port Concentrators (MPCs).

The application-aware access list (AACL) service adds support for a new service that uses application names and groups as matching criteria for filtering traffic. AACL is a stateless, rules-based service that must be combined with application identification to enable policies to be applied to flows based on application and application group membership in addition to traditional packet matching rules. It is supported on MX Series routers equipped with Multiservices DPCs and on M120 or M320 routers equipped with Multiservices 400 PICs. Starting with Junos OS Release 11.3, AACL is supported on T320, T640, and T1600 routers also.

AACL is configured in a similar way to other rules-based services such as Network Address Translation (NAT), class of service (CoS), and stateful firewall. To configure AACL, include rule specifications for match criteria and actions at the **[edit services aacl]** hierarchy level. You can chain AACL rules along with other service rules by including them in a service-set definition at the **[edit services service-set]** hierarchy level, as previously documented.

There is one pair of related operational commands, **show/clear application-aware-access-list statistics**.

For more information on the CLI configuration, see the *Application-Aware Access List*. For more information on the operational command, see the *Junos OS Operational Mode Commands*.

NOTE: Because the Junos OS extension-provider package framework lacks aggressive constraint checks, you should not set the **policy-db-size** statement at the **[edit chassis fpc** *slot-number* **pic** *pic-number* **adaptive-services service-package extension-provider]** hierarchy level to a high value. For Junos Application Aware (previously known as dynamic application awareness) configurations, the recommended values for the extension-provider options at this hierarchy level are as follows:

- **control-cores** = 1
- **data-cores** = 7
- **object-cache-size** = 1280 (for Multiservices 400 PIC and Multiservices DPC)
- **policy-db-size** = 200
- Include these **package** values: **jservices-idp**, **jservices-appid**, **jservices-llpdf**, **jservices-aacl**

For more information about this configuration, see the following topics in the *SDK Applications Configuration Guide and Command Reference*:

- *Configuring Control and Data Cores*
- *Configuring Memory Settings*
- *Configuring Packages on the PIC*

Related Documentation

- Configuring AACL Rules on page 11
- Configuring AACL Rule Sets on page 16
- Configuring Logging of AACL Flows on page 16
- Example: Configuring AACL Rules on page 19

Best-Effort Application Identification of DPI-Serviced Flows

This topic describes the following information:

Features that Support Application-Level Filtering

On MX Series routers equipped with Multiservices DPCs and M120 or M320 routers equipped with Multiservices 400 PICs, Intrusion Detection and Prevention (IDP) is accomplished by Deep Packet Inspection (DPI) of TCP, UDP, and ICMP flows. The application identification (APPID) feature defines applications as members of application groups in TCP/UDP/ICMP traffic. IDP depends on APPID for identification and detection of some Layer 7 applications.

The application-aware access list (AACL) service uses application names and groups as matching criteria for filtering traffic. The service defines the applications and application groups for which statistics are collected for a specific user or interface.

The local policy decision function (L-PDF) enables you to configure properties for statistics output. L-PDF supports a process that regulates collection of statistics related to applications and application groups and tracking of information about dynamic subscribers and static interfaces.

Best-Effort Application Determination

Typically, APPID conclusively determines the Layer 7 application associated with a given DPI-serviced flow. In these cases, the application identification is final. Occasionally, APPID is only able to make an initial, inconclusive determination of the Layer 7 application associated with a given flow. This is referred to as a "best-effort" application identification. In such cases, the APPID process continues processing packets on that flow and might subsequently make a conclusive determination of the application associated with that flow. In some cases of best-effort application identification, the flow ends before a final application determination can be made.

APPID, AACL, and L-PDF Processing in Preconvergence Scenarios

The following sections describe APPID, AACL, and L-PDF processing in various stages of application identification for a DPI-serviced flow of TCP/UDP/ICMP traffic.

Prior to a Final or Best-Effort Application Identification

During the time that APPID has not yet made either a final or best-effort determination of the application associated with a given flow, the flow does not contribute to any per-subscriber or per-application statistics collection.

The output of the following operational mode commands includes flows for which APPID has not yet made either a final or best-effort determination of the associated application:

- **show services local-policy-decision-function flows (interface** ***interface-name*** **| subscriber** ***subscriber-name*****)**
- **show services application-aware-access-list flows (interface** ***interface-name*** **| subscriber** ***subscriber-name*****)**

In the command output, the **Action** field displays "accept" and the **Application** or **Application group** field displays “unknown” for a flow for which APPID has not yet made either a final or best-effort determination of the associated application.

Upon Best-Effort Application Identification

When a best-effort application determination is made, AACL does not apply any AACL term actions configured for that flow. There are a number of reasons for this, one being that the action itself (such as "discard") could make a final application determination impossible. Instead, AACL or L-PDF tracks the flow and accepts all packets for that flow until a final determination is made, at which time the normal AACL or L-PDFL actions are fully applied to the flow.

While Application Identification Is on a Best-Effort Basis

During the time that APPID identification of the application associated with a given flow is on a best-effort basis, the flow does not contribute to any per-subscriber or per-application statistics collection.

The output of the following operational mode commands includes flows for which APPID has only made a best-effort determination of the associated application:

- **show services local-policy-decision-function flows (interface *interface-name* | subscriber *subscriber-name*)**
- **show services application-aware-access-list flows (interface *interface-name* | subscriber *subscriber-name*)**

In the command output, the **Action** field displays "accept" and the **Application** or **Application group** field displays “unknown” for a flow for which APPID has only made a best-effort determination of the associated application.

If a Flow Ends Before an Application Identification Is Made

If a flow ends before APPID has made either a final or a best-effort application identification, AACL or L-PDF uses the "unknown" application ID as a final determination and performs any necessary collection, aggregation, and reporting of statistics based on that Layer 7 application. In particular, if the **count** AACL term action is configured for the "application-group-any" application, then the statistics for that flow will be collected and aggregated against the count bucket type, and reported as such.

If a Flow Ends While Application Identification on a Best-Effort Basis

If a flow ends while the application identification is on a best-effort basis, AACL or L-PDF uses that best-effort determination as a final determination. AACL or L-PDF performs any necessary collection, aggregation, and reporting of statistics based on that Layer 7 application. In particular, if the **count** AACL term action is configured for that Layer 7 application, then the statistics for the flow will be collected and aggregated against the AACL or L-PDF statistics. However, in the case of nested applications, AACL and L-PDF will not consider the best-effort determination as final and the nested application will be reported as an unknown application.

Related Documentation

- Configuring AACL Rules on page 11
- *Configuring Statistics Profiles*
- *aacl-fields*

- *aacl-statistics-profile*
- rule on page 27
- services on page 28
- term on page 31
- then on page 32

PART 2

Configuration

CHAPTER 2

Configuration Tasks for AACL

Configuring AACL Rules

To configure an AACL rule, include the **rule** ***rule-name*** statement at the **[edit services aacl]** hierarchy level:

```
rule rule-name {
  match-direction (input | output | input-output);
  term term-name {
    from {
      application-group-any;
      application-groups [ application-group-names ];
      applications [ application-names ];
      destination-address address <any-unicast>;
      destination-address-range low minimum-value high maximum-value;
      destination-prefix-list list-name;
      nested-applications [ nested-application-names ];
      nested-application-unknown
      source-address address <any-unicast>;
      source-address-range low minimum-value high maximum-value;
      source-prefix-list list-name;
    }
    then {
      (accept | discard);
      count (application | application-group | application-group-any | nested-application
        | none);
      forwarding-class class-name;
      policer policer-name;
    }
  }
}
```

Each AACL rule consists of a set of terms, similar to a filter configured at the **[edit firewall]** hierarchy level. A term consists of the following:

- **from** statement—Specifies the match conditions and applications that are included and excluded.

- **then** statement—Specifies the actions and action modifiers to be performed by the router software.

The following sections explain how to configure the components of AACL rules:

- Configuring Match Direction for AACL Rules on page 12
- Configuring Match Conditions in AACL Rules on page 12
- Configuring Actions in AACL Rules on page 14
- Logging AACL Flows Based on Application on page 15

Configuring Match Direction for AACL Rules

Each rule must include a **match-direction** statement that specifies the direction in which the rule match is applied. To configure where the match is applied, include the **match-direction** statement at the **[edit services aacl rule *rule-name*]** hierarchy level:

```
match-direction (input | output | input-output);
```

If you configure **match-direction input-output**, bidirectional rule creation is allowed.

The match direction is used with respect to the traffic flow through the services PIC or DPC. When a packet is sent to the PIC or DPC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the services PIC or DPC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the PIC or DPC, the packet direction is output. For more information on inside and outside interfaces, see *Configuring Service Sets to be Applied to Services Interfaces*.

On the PIC or DPC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that matches the packet direction are considered.

Configuring Match Conditions in AACL Rules

To configure AACL match conditions, include the **from** statement at the **[edit services aacl rule *rule-name* term *term-name*]** hierarchy level:

```
from {
  application-group-any;
  application-groups [ application-group-names ];
  applications [ application-names ];
  destination-address address <any-unicast>;
  destination-address-range low minimum-value high maximum-value;
  destination-prefix-list list-name;
  nested-applications [ nested-application-names ];
  nested-application-unknown
  source-address address <any-unicast>;
```

```
    source-address-range low minimum-value high maximum-value;
    source-prefix-list list-name;
}
```

IPv4 and IPv6 source and destination addresses are supported. You can use either the source address or the destination address as a match condition, in the same way that you configure a firewall filter; for more information, see the *Routing Policy Feature Guide for Routing Devices*.

Alternatively, you can specify a list of source or destination prefixes by configuring the **prefix-list** statement at the **[edit policy-options]** hierarchy level and then including either the **destination-prefix-list** or the **source-prefix-list** statement in the AACL rule. For an example, see "Example: Configuring AACL Rules" on page 19.

If you omit the **from** term, the AACL rule accepts all traffic and the default protocol handlers take effect:

- User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Internet Control Message Protocol (ICMP) create a bidirectional flow with a predicted reverse flow.
- IP creates a unidirectional flow.

You can also include application and application group definitions you have configured at the **[edit services application-identification]** hierarchy level; for more information, see the topics in *Application Identification*.

- To apply one or more specific application protocol definitions, include the **applications** statement at the **[edit services aacl rule *rule-name* term *term-name* from]** hierarchy level.
- To apply one or more sets of application group definitions you have defined, include the **application-groups** statement at the **[edit services aacl rule *rule-name* term *term-name* from]** hierarchy level.

NOTE: If you include one of the statements that specifies application protocols, the router derives port and protocol information from the corresponding configuration at the **[edit services application-identification]** hierarchy level; you cannot specify these properties as match conditions.

- To consider any application group defined in the database as a match, include the **application-group-any** statement at the **[edit services aacl rule *rule-name* term *term-name* from]** hierarchy level.
- To consider any nested application defined in the database a match, include the **nested-applications** statement at the **[edit services aacl rule *rule-name* term *term-name* from]** hierarchy level. Nested applications are protocols that run on a parent application. For example, if the Facebook application runs on the parent application junos:http, the nested application will be junos:http:facebook.

Configuring Actions in AACL Rules

To configure AACL actions, include the **then** statement at the **[edit services aacl rule** ***rule-name*** **term** ***term-name*****]** hierarchy level:

```
then {
    (accept | discard);
    (count (application | application-group | application-group-any | nested-application |
      none) | forwarding-class class-name);
}
```

You must include one of the following actions:

- **accept**—The packet is accepted and sent on to its destination.
- **discard**—The packet is not accepted and is not processed further.

When you select **accept** as the action, you can optionally configure one or both of the following action modifiers. No action modifiers are allowed with the **discard** action.

- **count (application | application-group | application-group-any | nested-application | none)**—For all accepted packets that match the rules, record a packet count using AACL statistics practices. You can specify one of the following options; there is no default setting:
  - **application**—Count the application that matched in the **from** clause.
  - **application-group**—Count the application group that matched in the **from** clause.
  - **application-group-any**—Count all application groups that match **from application-group-any** under the **any** group name.
  - **nested-application**—Count all nested applications that matched in the **from** clause.
  - **none**—Same as not specifying **count** as an action.

NOTE:

- When a session closes before APPID has identified nested applications, the session is treated as a best-effort session and AACL does not get the nested application information. In such cases, nested applications will be reported as unknown applications.
- During the time that the application identification (APPID) feature has not yet made a final determination of the application associated with a given flow, the flow does not contribute to any per-subscriber or per-application statistics collection. For more information, see "Best-Effort Application Identification of DPI-Serviced Flows" on page 4.

- **forwarding-class** ***class-name***—Specify the packets' forwarding-class name.

You can optionally include a **policer** that has been specified at the **[edit firewall]** hierarchy level. Only the bit-rate and burst-size properties specified for the policer are applied in the AACL rule set. The only action application when a policer is configured is **discard**. For

more information on policer definitions, see the *Routing Policy Feature Guide for Routing Devices*.

Logging AACL Flows Based on Application

You can now log AACL flows based on application. You can select a specific application or request information on unknown applications.

You can now configure AACL rules to match unknown applications. All existing actions that can apply to recognized applications can also apply to unknown applications. You can use the following statements at the **[edit services aacl rule *rule-name* term *term-name* from]** hierarchy level:

- application-group-any
- application-groups
- application-unknown
- applications
- nested-application-unknown
- nested-applications

The addition of matching "application unknown" enables the specific logging of the input flows associated with applications that cannot be identified. Because logging is triggered by an input event, you must specify **match-direction** as **input-output** or **input**.

To configure logging of flows for AACL, include the **match-direction input** or **match-direction input-output** statement at the **[edit services aacl rule *rule-name*]** hierarchy level, include an **applications** or **application-unknown** statement at the **[edit services aacl rule *rule-name* term *term-name* from]** hierarchy level, and include only one **log** statement at the **[edit services aacl rule *rule-name* term *term-name* then]** hierarchy level. The log statements can include any of the following options:

- **session-start**
- **session-end**
- **session-start-end-no-stats**
- **session-start-interim-end**
- **session-interim-end**
- **session-end**

Related Documentation

- AACL Overview on page 3
- Configuring AACL Rule Sets on page 16
- Configuring Logging of AACL Flows on page 16
- Example: Configuring AACL Rules on page 19

Configuring AACL Rule Sets

The **rule-set** statement defines a collection of AACL rules that determine what actions the router software performs on packets in the data stream. You define each rule by specifying a rule name and configuring terms. Then, you specify the order of the rules by including the **rule-set** statement at the **[edit services aacl]** hierarchy level with a **rule** statement for each rule:

```
rule-set rule-set-name {
  rule rule-name;
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules matches the packet, the packet is dropped by default.

Related Documentation

- AACL Overview on page 3
- Configuring AACL Rules on page 11
- Configuring Logging of AACL Flows on page 16
- Example: Configuring AACL Rules on page 19

Configuring Logging of AACL Flows

You can configure logging of AACL flows for a given application or for all unknown applications using AACL rules. You must set **match-direction** to **input** or **input-output** for logging to occur.

1. Create a rule and term.

   user@host# **edit services aacl rule** *rule-name* **term** *term-name*

2. Specify selection of an application.

   [edit services aacl rule *rule-name* term *term-name*]
   user@host# **set from applications** *application-name*]

   OR

   Specify selection of all unknown applications.

   [edit services aacl rule <variable>rule-name</variable > term
   <variable>term-name</variable>]
   set from application-unknown

3. In the **then** statement, specify logging of input flow.

   [edit services aacl rule *rule-name* term *term-name*]
   user@host# **set then log input-flows**]

Example—Configuration of Logging of Input Flows for Unknown Applications

```
[edit services aacl rule aacl_rule5]
match-direction input-output;
  term t0 {
    from {
      application-unknown;
    }
    then {
      count application;
      log input-flow;
      accept;
    }
  }
```

Example—Setup of a Specific Log File

The following example shows how to direct the aacl flow log to a file other than the default syslog file on the Routing Engine file system.

```
[edit system syslog]
file aacl_log {
  external any;
  match aacl-flow-log;
}
```

Related Documentation

- AACL Overview on page 3
- Configuring AACL Rules on page 11
- Configuring AACL Rule Sets on page 16
- Example: Configuring AACL Rules on page 19

CHAPTER 3

AACL Example

Example: Configuring AACL Rules

The following example shows an AACL configuration containing a rule with three terms using a variety of match conditions and actions:

```
[edit services aacl]
rule aacl-test {
    match-direction input;
    term term1 {
        from {
            source-address 10.0.1.1
            application test1;
        }
        then {
            accept;
        }
    }
    term term2 {
        from {
            source-address {
                any-unicast;
            }
            application test1;
        }
        then {
            discard;
        }
    }
    term term3 {
        from {
            source-address {
                any-unicast;
            }
            application test1 test2;
        }
        then {
            accept;
            count application;
        }
```

```
    }
  }
```

Related Documentation

- AACL Overview on page 3
- Configuring AACL Rules on page 11

CHAPTER 4

AACL Configuration Statements

[edit services aacl] Hierarchy List

To configure application-aware access list (AACL) services, include the **aacl** statements at the **[edit services]** hierarchy level:

```
[edit services]
aacl {
  rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
      from {
        application-group-any;
        application-groups [ application-group-names ];
        application-unknown;
        applications [ application-names ];
        destination-address address <any-unicast>;
        destination-address-range low minimum-value high maximum-value;
        destination-prefix-list list-name;
        source-address address <any-unicast>;
        source-address-range low minimum-value high maximum-value;
        source-prefix-list list-name;
      }
      then {
        (accept | discard);
        count (application | application-group | application-group-any | none);
        forwarding-class class-name;
        policer policer-name;
      }
    }
  }
  rule-set rule-set-name {
    [ rule rule-names ];
  }
}
```

Related Documentation

- Configuring AACL Rules on page 11
- Configuring AACL Rule Sets on page 16
- Configuring Logging of AACL Flows on page 16

applications (Services AACL)

Syntax applications [*application-names*];

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.

Description Identify one or more applications defined in the application identification configuration for inclusion as a match condition.

Options ***application-names***—Identifiers of the applications.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

application-groups (Services AACL)

Syntax application-groups [*application-group-names*];

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.

Description Identify one or more application groups defined in the application identification configuration for inclusion as a match condition.

Options ***application-group-names***—Identifiers of the application groups.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

application-group-any

Syntax application-group-any;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.

Description Indicates that any application group defined in the database is considered a match.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

application-unknown

Syntax application-unknown

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 11.2.

Description Enable AACL logging of flows for unknown applications.

Related Documentation
- See Configuring Logging of AACL Flows on page 16.

destination-address

Syntax destination-address *address*;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.
IPv6 support introduced in Junos OS Release 12.2

Description Specify the destination address for rule matching.

Options ***address***—Destination IPv4 or IPv6 address or prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

destination-address-range

Syntax destination-address-range low *minimum-value* high *maximum-value*;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.
IPv6 support introduced in Junos OS Release 12.2

Description Specify the destination address range for rule matching.

Options ***minimum-value***—Lower boundary for the IPv4 or IPv6 address range.

maximum-value—Upper boundary for the IPv4 or IPv6 address range.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

destination-prefix-list

Syntax destination-prefix-list *list-name*;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.

Description Specify the destination prefix list for rule matching. You configure the prefix list by including the **prefix-list** statement at the **[edit policy-options]** hierarchy level.

Options ***list-name***—Destination prefix list.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

from

Syntax

```
from {
  application-group-any;
  application-groups [ application-group-names ];
  application-unknown;
  applications [ application-names ];
  destination-address address <any-unicast>;
  destination-address-range low minimum-value high maximum-value;
  destination-prefix-list list-name;
  nested-application-unknown;
  source-address address <any-unicast>;
  source-address-range low minimum-value high maximum-value;
  source-prefix-list list-name;
}
```

Hierarchy Level [edit services aacl rule *rule-name* term *term-name*]

Release Information Statement introduced before Junos OS Release 9.5.

Description Specify match conditions for the AACL term.

Options For information on match conditions, see the description of firewall filter match conditions in the *Routing Policy Feature Guide for Routing Devices*.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring AACL Rules on page 11

log (aacl)

Syntax log *event-type*

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 11.2.

Description Enable AACL logging of flows for known or unknown applications.

Options ***event-type***—Enable logging of the specified ***event-type***:

- **session-start**
- **session-end**
- **session-start-end-no-stats**
- **session-start-interim-end**
- **session-interim end**
- **session-end**

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- See Configuring Logging of AACL Flows on page 16.

match-direction

Syntax match-direction (input | output | input-output);

Hierarchy Level [edit services aacl rule *rule-name*]

Release Information Statement introduced in Junos OS Release 9.5.

Description Specify the direction in which the rule match is applied.

Options **input**—Apply the rule match on the input side of the interface.

output—Apply the rule match on the output side of the interface.

input-output—Apply the rule match bidirectionally.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Match Direction for AACL Rules on page 12

rule

Syntax

```
rule rule-name {
  match-direction (input | output | input-output);
  term term-name {
    from {
      application-group-any;
      application-groups [ application-group-names ];
      application-unknown;
      applications [ application-names ];
      destination-address address <any-unicast>;
      destination-address-range low minimum-value high maximum-value;
      destination-prefix-list list-name;
      nested-application-unknown;
      source-address address <any-unicast>;
      source-address-range low minimum-value high maximum-value;
      source-prefix-list list-name;
    }
    then {
      (accept | discard);
      count (application | application-group | application-group-any | nested-application |
        none);
      forwarding-class class-name;
      policer policer-name;
    }
  }
}
```

Hierarchy Level

[edit services aacl],
[edit services aacl rule-set *rule-set-name*]

Release Information

Statement introduced in Junos OS Release 9.5.

Description

Specify the rule the router uses when applying this service.

Options

rule-name—Identifier for the collection of terms that constitute this rule.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring AACL Rules on page 11

rule-set (Services AACL)

Syntax

```
rule-set rule-set-name {
    [rule rule-names ];
}
```

Hierarchy Level [edit services aacl]

Release Information Statement introduced in Junos OS Release 9.5.

Description Specify the rule set the router uses when applying this service.

Options ***rule-set-name***—Identifier for the collection of rules that constitute this rule set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring AACL Rule Sets on page 16

services (AACL)

Syntax services aacl { ... }

Hierarchy Level [edit]

Release Information **aacl** statement introduced in Junos OS Release 9.5.

Description Define the services to be applied to traffic.

Options **aacl**—The values configured for application-aware-access-list matching rules.

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Application-Aware Access List*

source-address (AACL)

Syntax source-address *address*;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.
IPv6 support introduced in Junos OS Release 12.2

Description Specify the source address for rule matching.

Options ***address***—Source IPv4 or IPv6 address or prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

source-address-range

Syntax source-address-range low *minimum-value* high *maximum-value*;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.
IPv6 support introduced in Junos OS Release 12.2

Description Specify the source address range for rule matching.

Options ***minimum-value***—Lower boundary for the IPv4 or IPv6 address range.

maximum-value—Upper boundary for the IPv4 or IPv6 address range.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in AACL Rules on page 12

source-prefix-list

Syntax source-prefix-list *list-name*;

Hierarchy Level [edit services aacl rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 9.5.

Description Specify the source prefix list for rule matching. You configure the prefix list by including the **prefix-list** statement at the **[edit policy-options]** hierarchy level.

Options ***list-name***—Source prefix list.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Match Conditions in AACL Rules on page 12

term

Syntax

```
term term-name {
  from {
    application-group-any;
    application-groups [ application-group-names ];
    application-unknown;
    applications [ application-names ];
    destination-address address <any-unicast>;
    destination-address-range low minimum-value high maximum-value;
    destination-prefix-list list-name;
    nested-application-unknown;
    source-address address <any-unicast>;
    source-address-range low minimum-value high maximum-value;
    source-prefix-list list-name;
  }
  then {
    (accept | discard);
    count (application | application-group | application-group-any | nested-application |
      none);
    forwarding-class class-name;
    policer policer-name;
  }
}
```

Hierarchy Level [edit services aacl rule *rule-name*]

Release Information Statement introduced in Junos OS Release 9.5.

Description Define the AACL term properties.

Options ***term-name***—Identifier for the term.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring AACL Rules on page 11

then

Syntax

```
then {
  (accept | discard);
  count (application | application-group | application-group-any | nested-application | none);
  forwarding-class class-name;
  log event-type;
  policer policer-name;
}
```

Hierarchy Level [edit services aacl rule *rule-name* term *term-name*]

Release Information Statement introduced in Junos OS Release 9.5.
policer statement added in Junos OS Release 9.6.
The **nested-application** option for the **count** statement introduced in Junos OS Release 11.1.

Description Define the AACL term actions. You can configure the router to accept or discard the targeted traffic. The action modifiers (count and forwarding-class) are optional.

Options You can configure one of the following actions:

- **accept**—Accept the packets and all subsequent packets in flows that match the rules.
- **discard**—Discard the packet and all subsequent packets in flows that match the rules.

When you select **accept** as the action, you can optionally configure one or both of the following action modifiers. No action modifiers are allowed with the **discard** action.

- **count (application | application-group | application-group-any | nested-application | none)**—For all accepted packets that match the rules, record a packet count using AACL statistics practices. You can specify one of the following options; there is no default setting:
  - **application**—Count the application that matched in the **from** clause.
  - **application-group**—Count the application group that matched in the **from** clause.
  - **application-group-any**—Count all application groups that match **from application-group-any** under the **any** group name.
  - **nested-application**—Count all nested applications that matched in the **from** clause.
  - **none**—Same as not specifying **count** as an action.
- **forwarding-class *class-name***—Specify the packets' forwarding-class name.

policer *policer-name*—Apply rate-limiting properties to the traffic as configured at the **[edit firewall policer *policer-name*]** hierarchy level. This configuration allows bit-rate and burst-size attributes to be applied to the traffic that are not supported by AACL rules. When you include a policer, the only allowed action is **discard**. For more information on policers, see the *Routing Policy Feature Guide for Routing Devices*.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring AACL Rules on page 11
- *Routing Policy Feature Guide for Routing Devices*

PART 3

Administration

CHAPTER 5

AACL Operational Mode Commands

clear services application-aware-access-list statistics

Syntax clear services application-aware-access-list statistics

Release Information Command introduced in Junos OS Release 9.5.

Description Clear application aware access list (AACL) statistics.

Options This command has no options.

Required Privilege Level clear

Related Documentation
- show services application-aware-access-list statistics on page 39

show services application-aware-access-list statistics

Syntax **show services application-aware-access-list statistics**
<interface *interface-name*>
<subscriber *subscriber-name*>

Release Information Command introduced in Junos OS Release 9.5.

Description Display application-aware-access-list (AACL) statistics.

Options **interface *interface-name***—(Optional) Dsiplays AACL statistics for the specified interface(s) only.

subscriber*subscriber-name*—(Optional) Dsiplays AACL statistics for the specified subscriber(s) only.

Required Privilege Level view

List of Sample Output show services application-aware-access-list statistics by interface on page 40
show services application-aware-access-liststatistics by subscriber on page 40

Output Fields Table 3 on page 39 lists the output fields for the **show services application-aware-access-list statistics** command. Output fields are listed in the approximate order in which they appear.

Table 3: show services application-aware-access-list statistics Output Fields

Field Name	Field Description	Level of Output
Interface	Interface name.	Subscriber option
Subscriber	Subscriber identifier.	Interface option
Service-set-interface	Service set interface name.	All levels
Service set	Service set name.	All levels
Application group	Application group identifier.	All levels
Packets in	Number of ingress packets.	All levels
Bytes in	Number of ingress bytes.	All levels
Packets out	Number of egress packets.	All levels
Bytes out	Number of egress bytes.	All levels

Sample Output

show services application-aware-access-list statistics by interface

```
user@host> show services application-aware-access-list statistics interface ge-0/0/0.100
Subscriber: user@juniper.net

service-set: IDP
service-set interface: ms-2/0/0

Application group     Application           Packets in            Bytes in
        Packets out           Bytes out

                      junos:ftp [63]        5                     334
        6                     346
```

show services application-aware-access-list statistics by subscriber

```
user@host> show services application-aware-access-list statistics subscriber user@juniper.net
Interface: ge-1/1/0.0

Service-set-interface: ms-1/3/0
Service set: aacl-svc-set

Application-aware-access-list statistics

Application group       Packets in       Bytes in         Packets out       Bytes
 out
P2P                                  400                    32025            200
                    16284
FTP                                  20000               5231000       100
               8700
```

show services application-aware-access-list flows

Syntax **show services application-aware-access-list flows**
<interface *interface-name*>
<subscriber *subscriber-name*>

Release Information Command introduced in Junos OS Release 10.1.
Offload status for flows using Juniper Forwarding Mechanism (JFM) added in Junos OS Release 12.1.

Description Display application-aware-access-list (AACL) flows. Offloading using JFM is supported only on MX Series routers with Modular Port Concentrators (MPCs).

Options **interface *interface-name***—Displays AACL flows for the specified interface(s) only. The keyword, interface, must be appended to the command.

subscriber *subscriber-name*—Displays AACL flows for the specified subscriber(s) only. The keyword, subscriber, must be appended to the command.

Required Privilege Level view

Related Documentation
- *Application-Aware Access List*

List of Sample Output
show services application-aware-access-list flows by interface on page 42
show services application-aware-access-list flows by subscriber on page 42
show services application-aware-access-list flows by subscriber for offloading using JFM on page 43

Output Fields Table 4 on page 41 lists the output fields for the **show services application-aware-access-list flows** command. Output fields are listed in the approximate order in which they appear.

Table 4: show services application-aware-access-list flows Output Fields

Field Name	Field Description	Level of Output
5-tuple	This field comprises five components of the given flow. The components are: • Src IP • Dest IP • Src Port • Dest Port • Protocol	All levels
Application-ID	The identification number associated with the application.	All levels
Dir	The direction in terms of input or output. • Input (I) • Output (O)	All levels

Table 4: show services application-aware-access-list flows Output Fields *(continued)*

Field Name	Field Description	Level of Output
Off	The status of offload to Packet Forwarding Engine. The various options are: • Not Offloaded (-) • Policer Offloaded, Flow Not Offloaded (P) • Policer Not Offloaded, Flow Offloaded (F) • Policer and Offloaded (P+F)	All levels
Off	The status of offload to Packet Forwarding Engine using JFM. The various options are: • Not Offloaded (-) • Offload requested but not completed (R) • Offload requested and completed (O)	All levels
Actions	The types of actions displayed are: • discard: (D) • accept : A • accept, count [T]: C-A or C-G or C-T • accept, fwd-class [C]: FC • accept, policer [P]: P • accept, count [T], fwd-class [C]: C-T+FC • accept, count [T], policer [P]: C-T+P • accept, fwd-class [C], policer [P]: FC+P • accept, count[T],fwd-class[C],policer[P]: C-T+FC+P	All levels

Sample Output

show services application-aware-access-list flows by interface

```
user@host>show services application-aware-access-list flows interface ge-1/0/5.0
Interface: ge-1/0/5.0
service-set: aacl-countApps
service-set interface: ms-0/0/0
Currently active flows: 2
High watermark flows: 2

5-tuple                                          Application-ID
Dir Off Action
------------------------------------------------ ------------------------------
--- --- --------
        1.0.5.2:47072->  10.10.254.116:80   ,6   junos:http [64]
I   -   C-T
  10.10.254.116:80   ->        1.0.5.2:47072,6   junos:http [64]
O   -   C-T
```

show services application-aware-access-list flows by subscriber

```
user@host>show services application-aware-access-list flows subscriber user@juniper.net
Subscriber: user@juniper.net

Service-set: ss1
```

```
Service-set interface: ms-2/0/0
Currently active flows: 4
High watermark flows: 40


5-tuple              Application-ID   Dir Off Action

150.100.100.100:20109->160.200.200.200:80,17  junos:http [64]  I  -  C-T+FC+P
160.200.200.200:80->150.100.100.100:20109,17  junos:http [64]  O  -  C-T+FC+P
150.100.100.100:20108->160.100.100.100:80,17  junos:http [64]  I  P+F C-T+FC+P
160.100.100.100:80->150.100.100.100:20108,17  junos:http [64]  O  P+F C-T+FC+P
```

show services application-aware-access-list flows by subscriber for offloading using JFM

```
user@host>show services application-aware-access-list flows subscriber user@juniper.net
Subscriber: user@juniper.net

Service-set: ss1
Service-set interface: ms-2/0/0
Currently active flows: 4
High watermark flows: 40


5-tuple              Application-ID   Dir Off Action

150.100.100.100:20109->160.200.200.200:80,17     junos:http [64]              I
   -   C-T+FC+P
160.200.200.200:80   ->150.100.100.100:20109,17  junos:http [64]              O
   -   C-T+FC+P
150.100.100.100:20108->160.100.100.100:80,17     junos:http [64]              I
   R   C-T+FC+P
160.100.100.100:80   ->150.100.100.100:20108,17  junos:http [64]              O
   O   C-T+FC+P
```

PART 4

Index

Index

R

S

T