JUNIPer | Engineering
NETWORKS | Simplicity

JUNOS

Junos® OS

Identity Aware Firewall User Guide

Published
2024-12-19

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 Identity Aware Firewall User Guide
Copyright © 2024 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

3

Configuration Statements and Operational Commands

About This Guide

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.

1
CHAPTER

Overview

Identity-Aware Firewall

SUMMARY

Learn about identity-aware firewall and its component authentication table.

IN THIS SECTION

Identity is the foundation for securing any network, application, device, and user access. User identity (credentials, group information, IP address) and device information, when collected through various sources helps secure your network resources.

Security Operations Centre (SOC) and Network Operations Centre (NOC) teams can use identity parameters to configure security policies. Security policies provide the right level of access to the authenticated users.

Identity helps your organization to mitigate major security threats by securing access to your resources (network components, applications) based on username, roles, and groups.

Figure 1: Identity Sources Deployment at Juniper SRX Series Firewall level
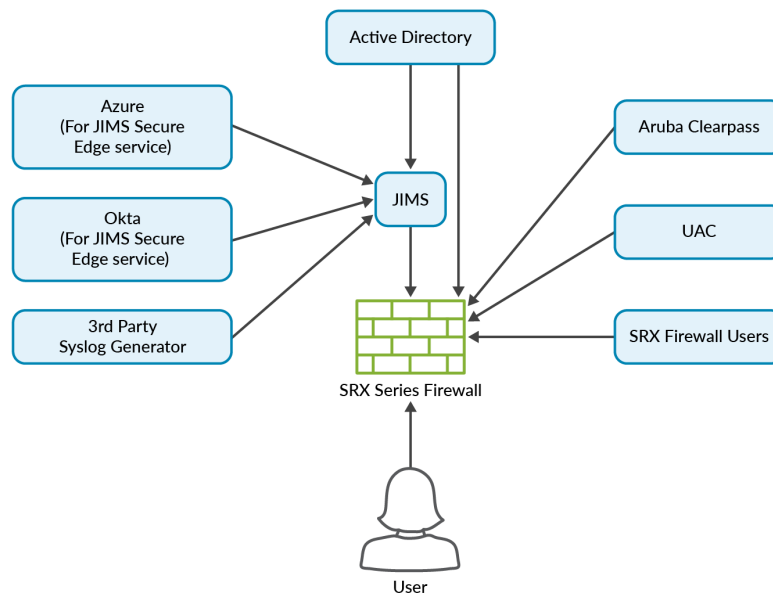
Benefits of Identity-Aware Firewall

An identity-aware firewall:

- Adds an additional layer of security to your network.

- Helps identify the source (user or device) of the traffic in the network.

- Provides visibility on users and devices that generate alerts, alarms, and cause security incidents.

- Optimizes user experience by providing users with a streamlined and smooth access to the appropriate resources without compromising on network and application security.

Deploy Identity Source at SRX Series Firewall Level

IN THIS SECTION

How is Identity Source deployed at Juniper SRX Series Firewall level?

A *firewall* obtain user information from various identity sources such as Active Directory, Juniper® Identity Management Service (JIMS), Aruba ClearPass, and Unified Access Control (UAC). After obtaining user information, the network administrator can deploy the device to receive data from the identity sources.

> **TIP**: In this guide, we use *firewall* to refer to a Juniper Networks® SRX Series Firewall or a Juniper Networks® vSRX Virtual Firewall (vSRX3.0), or a Juniper Networks® cSRX Container Firewall or a Juniper Networks® NFX Series Network Services Platform (NFX Series) devices.

SRX Series Firewalls can create, manage, and redefine firewall rules that are based on user identity rather than an IP address. The firewalls can query JIMS, obtain the correct user identity information, and then enforce the appropriate security policy decisions to permit or deny access to protected corporate resources and the Internet.

Each component involved in the identity source process has its own security infrastructure where the authorization policies governing access to protected resources are administratively defined. The following tables describe the roles of these components and how the components communicate with each other.

Table 1: Identity Deployment at Juniper SRX Series Firewall level

Deployment Mode	Identity Source	Deployment Mode Details
Active Directory as Identity Source	Active Directory	Simple configuration and any identity manager outside of SRX Series Firewalls is not required.
Juniper® Identity Management Service (JIMS) as Identity Manager	Active Directory	Higher scalability and deployment flexibility.
	Microsoft Azure	SRX Series Firewalls can support user identities that are configured in Azure.
	Okta	SRX Series Firewalls can support user identities that are configured in Okta using JIMS.
	Third-party Syslog generator	JIMS can gather syslog information that is generated through various third-party sources for obtaining user information.
Aruba ClearPass as Identity Source	Aruba ClearPass	SRX Series Firewall can obtain user information from Aruba ClearPass.
Unified Access Control as Identity Source	UAC	SRX Series Firewalls can obtain user information that is configured on the UAC.

Table 1: Identity Deployment at Juniper SRX Series Firewall level *(Continued)*

Deployment Mode	Identity Source	Deployment Mode Details
SRX Firewall Users	Firewall users	The user information can be configured on a firewall authentication and can be maintained on a local server, or on an external server such as LDAP or RADIUS server.

Role of Identity Source and Identity Manager

Identity Source An identity source can manage user or device information, roles, and maintain user events.

Identity Manager JIMS is an advanced user identity management system.

JIMS connects and communicates to various identity sources that are shown in Figure 1 on behalf of the SRX Series Firewall.

Table 2: Description of Identity Aware Firewall Components

Deployment Mode	Description	Benefits of Deployment
Active Directory as Identity Source	Active Directory as an Identity Source gathers user and group information for authentication by reading domain controller event logs, probing domain PCs, and querying Lightweight Directory Access Protocol (LDAP) services within the configured Windows domain.	**Centralized management:** Centralizes user and group management within an organization, which simplifies administration. **Effective authentication:** Verifies user and computer identities, enhancing network security. **Policy enforcement:** Allows administrators to enforce security policies using Group Policy Objects (GPOs). **Dependencies:** Relies on Active Directory infrastructure. Some organizations might not prefer this infrastructure.

Table 2: Description of Identity Aware Firewall Components *(Continued)*

Deployment Mode	Description	Benefits of Deployment
Juniper® Identity Management Service (JIMS)	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 Identity Source domains or syslog sources. For more information, see JIMS with SRX Series Firewall.	**Efficient management:** Simplifies end-user experience by automating the correlation between usernames, devices, and IP addresses. **Load reduction:** Reduces load on the identity management system by acting as middleware between identity management and firewalls. **Access control:** Allows policy control that is based on group memberships, enhancing security, and access restrictions. **Dependencies:** Relies on a standalone Windows service, potentially adding a point of failure or dependency on the Windows environment.
Aruba ClearPass as Identity Source	A *firewall* and Aruba ClearPass collaborate to protect your network resources. These devices control user access to the Internet and enforce security at the user identity level that is based on the usernames or by the groups that the network resources belong to.	**Policy management:** Facilitates policy management for onboarding new devices and controlling access based on roles and device types. **Granular access control:** Grants access levels based on user roles and enhances security and compliance. **Dependencies:** The identity source must be integrated with Aruba ClearPass, which might require additional configuration and setup.

Table 2: Description of Identity Aware Firewall Components *(Continued)*

Deployment Mode	Description	Benefits of Deployment
Unified Access Control (UAC) as Identity Source	A Unified Access Control (UAC) uses IC Series UAC Appliances, intranet Enforcers, and Infranet Agents to protect your network by ensuring only valid users can access the resources. An IC Series appliance is a policy decision point in the network that uses authentication information and policy rules to determine whether or not to provide access to individual resources on the network.	**Simplified configuration:** Simplifies configuration by creating user information, groups, and policy rules in a centralized location. **Enforced Security:** Ensures that only valid users can access network resources through IP-based policies. **Dependencies:** Requires the deployment of IC Series UAC Appliances, intranet Enforcers, and Infranet Agents. This might make the network more complex.

Choose Identity-Aware Firewall Components

Customers typically choose identity-aware firewall components that are based on their specific organizational needs, existing infrastructure, and security requirements:

Source of Identity

You must connect the SRX Series firewall with one of the identity sources or identity manager, described in Table 1 on page 4.

Scaling

For the deployments with higher scaling requirements, Juniper Identity Management Service is recommended.

Before you choose the components of identity-aware firewall, evaluate the complexity of integration and maintenance requirements to ensure that they align with your organizational goals and security requirements.

Authentication Table

What is Authentication Table?

The authentication table contains the IP address, username, and group mapping information that serves as the authentication source.

How Authentication Table is implemented?

The user and group mapping information in the authentication table is obtained by user identity information. When JIMS is deployed, the authentication table is obtained by using IP query or batch-query.

The obtained information in the table is generated on the Routing Engine of the device, that push 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.

You must configure Active-Directory authentication table to enable Active Directory as Identity Source information retrieval in the Windows Active Directory environment. See "Configure Active Directory as Identity Source on SRX Series Firewall" on page 54.

The `priority` option specifies the sequence in which user information tables are checked. Using the lowest setting for the identity source specifies the highest priority, meaning that the Active Directory authentication source is searched first. For more information, see "Active Directory as Identity Source Authentication Table" on page 19 and "ClearPass Authentication Table" on page 240.

How Authentication Tables are managed?

Windows domain environments are constantly changing as users log in and log out of the network and as network administrators modify user group information. The identity source manages changes in the

Windows domain and updates periodically. The authentication table is also updated to reflect the up-to-date relevant group information for all listed users.

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.

See .

State Information for Identity Source Authentication Table Entries

Identity source 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.

Timeout Setting

When a user is no longer active, a timer is started for that user's entry in the 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.

We recommend that you disable timeouts when disabling on-demand probing in order to prevent someone from accessing the Internet without logging in again.

For more information, see and .

2

CHAPTER

Identity Sources

Active Directory as Identity Source

SUMMARY

Learn about Active Directory as an identity source, its benefits, and how an Active Directory works as an identity source.

An identity source stores user information in a database. You can use this information for user authentication. Active Directory as an identity source defines integration of *firewall* with Microsoft Windows Active Directory.

> 💡 **TIP**: Active Directory as identity source was previously known as Integrated User Firewall.

Overview of Active Directory as Identity Source

Figure 2 on page 12 illustrates a typical scenario where the Active Directory as identity source is deployed. Users inside and outside the Active Directory domain need to access the Internet through a device.

Benefits

- Simplifies configuration steps and provides best-effort security to access the device.

- Ideal for medium businesses and up to medium-scale deployments.

- Supports high availability (HA).

- Configuration of captive portal is optional for users who do not authenticate.

How does Active Directory as Identity Source Work?

Figure 2: Active Directory as Identity Source Deployment
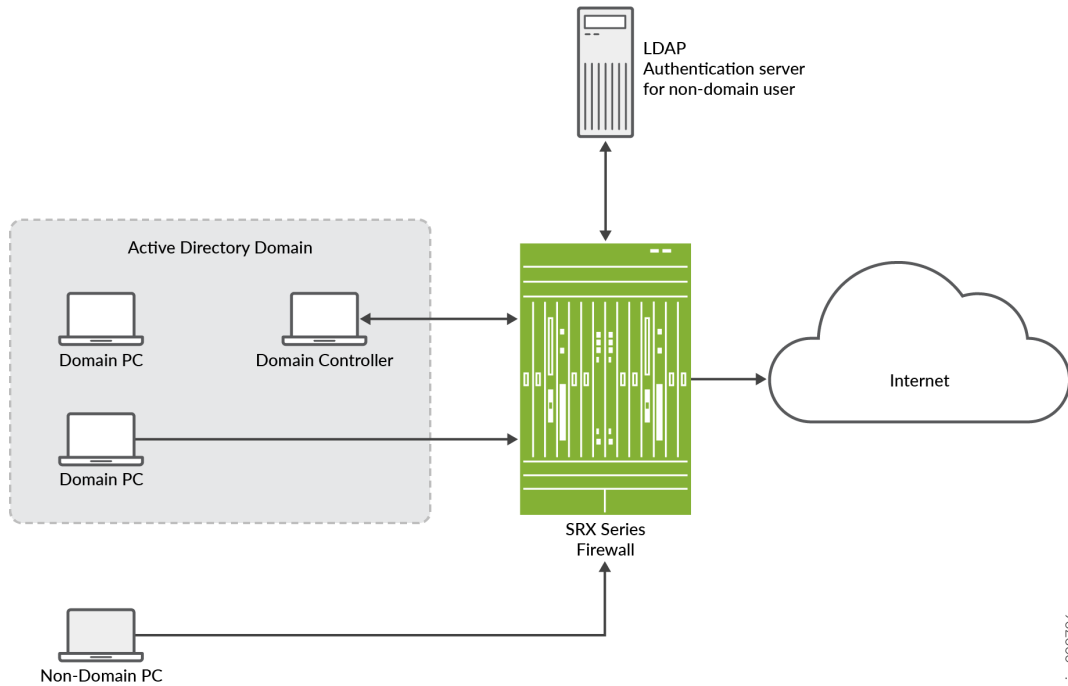
Table 3: Components of Active Directory as Identity Source

Components	Description
Domain Controller	Domain controller helps to apply security policies for request access to user authentication resources. The domain controller might also act as the LDAP server.
Domain PC	Connected windows computers group that share user account information and a security policy.

Table 3: Components of Active Directory as Identity Source *(Continued)*

Components	Description
Non-domain PC	Users can access Windows desktop enviroment from any location using a device with internet connectivity.
LDAP authentication server for non-domain user	LDAP protocol helps identify the groups to which users belong. The username and group information are queried from the LDAP service in the Active Directory controller.

1. The device reads and analyzes the Windows event log of the active directory controller and generates an authentication table.

2. The Active Directory as identity source is aware of any domain user via the authentication table.

3. The device configures a policy that enforces the required user-based or group-based access control.

4. For any non-domain user or domain user on a non-domain machine, the administrator specifies a captive portal to force the user to authenticate (if the device supports captive portal for the traffic type).

5. After users enter their names and passwords and authenticate, the device gets user/group information and enforces the policy.

6. In addition to captive portal, if the IP address or user information is not available from the event log, users can again log in to the Windows PC to generate an event log entry. Then, the system generates the user authentication entry.

Windows Management Instrumentation Client (WMIC)

What is Windows Management Instrumentation Client (WMIC)?

When you configure the Active Directory as identity source, the device establishes a Windows Management Instrumentation (WMI)/Distributed Component Object Module (DCOM) connection with the domain controller. The device acts as a WMI client (WMIC). The device 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.

How WMIC reads the Event Log on the Domain Controller?

WMIC reads the event log on the domain controller in following manner:

1. The device monitors the event log at a configurable interval, which defaults to 10 seconds.

2. The device reads the event log for a certain timespan, which you can configure. The default timespan is one hour.

3. Each time at WMIC starts up, the device checks the last timestamp and the current timespan. If the last timestamp is older than the current timespan, then the current timespan takes effect.

4. The device can read the event log to obtain both IPv4 and IPv6 addresses.

5. When WMIC starts up, the device has a maximum count of events that it can read from the event log. You cannot configure the maximum count of events.

   - On the SRX300, Juniper Networks® SRX320 Firewall, Juniper Networks® SRX340 Firewall, Juniper Networks® SRX345 Firewall, and Juniper Networks® SRX380 Firewall, the maximum count is 100,000.

   - On the Juniper Networks® SRX5400 Firewall, Juniper Networks® SRX5600 Firewall, and Juniper Networks® SRX5800 Firewall, the maximum count is 200,000.

   When WMIC starts up, the WMIC uses the maximum count with the timespan setting. When the limit is reached, the WMIC stops reading the event log.

6. After a failover, the device reads the event log from the latest event log timestamp.

Specify IP Filters to Limit IP-to-User Mapping

You can specify the 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.

Windows 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.

Firewall Authentication as backup to WMIC

The primary method for the Active Directory as identity source to get IP address-to-user mapping information is for the device to act as a WMI client (WMIC). However, the event-log-reading and PC probe functions both use WMI, and using a global policy to disable the WMI-to-PC probe affects event log reading. These might result in the failure of the PC probe, and a backup method for getting IP address-to-user mappings is needed. That method is to use firewall authentication to identify users.

See .

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.

Domain PC Probing

What is Domain PC Probing?

Domain 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 Active Directory as identity source 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.

Domain PC Probing User Information

The Active Directory as identity source 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.

> *i* **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.

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.

If the device cannot access a domain PC for some reason, such as a network configuration or Windows firewall issue, the probe fails.

For more probe responses and corresponding authentication table actions, see .

Table 4: 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 (The configured username and password have no privilege to probe the domain PC.)	Update IP-related entry as invalid.

How to Configure Probe?

On-demand probing is enabled by default. To disable on-demand probing, you can 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.

To initiate a manual probe, you can 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.

The probe timeout value is configurable. 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.

See "Configure Active Directory as Identity Source on SRX Series Firewall" on page 54.

Probe Rate and Statistics

The maximum probe rate for the Active Directory as identity source is set by default and cannot be changed. For SRX 5400, SRX 5600, and SRX 5800 Series Firewalls, the probe rate is 600 times per minute. For branch SRX Series Firewalls, 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.

LDAP for Active Directory as Identity Source

What is the use of LDAP for Active Directory as Identity Source?

The use of LDAP in this section applies specifically to LDAP functionality within the Active Directory as identity source.

In order to get the user and group information necessary to implement the Active Directory as identity source, the device 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 Active Directory as identity source, 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.

How LDAP for Active Directory as Identity Source Work?

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.

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.

Active Directory as Identity Source Authentication Table

The Active Directory as identity source 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 Active Directory as identity source. When an authentication entry in an authentication table is deleted, Active Directory as identity source only closes sessions that are associated with that entry. It will not close sessions that it does not manage.

Table 5 on page 19 lists Active Directory as identity source authentication table device support that depends on the Junos OS release in your installation.

Table 5: Active Directory as Identity Source Authentication Table Device Support

Devices	Identity Source Authentication Table Entries	Domains	Controllers
SRX300, SRX320	500	1	5
SRX340, SRX345, SRX380	1000	1	5
SRX1500, SRX1600, SRX2300	20,000	2	10
SRX4000 line	50,000	2	10

Table 5: Active Directory as Identity Source Authentication Table Device Support *(Continued)*

Devices	Identity Source Authentication Table Entries	Domains	Controllers
SRX5000 line	The user entries are as follows: • 100000—For users without JIMS • 256000—For users with JIMS	2	10
vSRX Virtual Firewall (2 vCPUs and 4 GB vRAM, 5 vCPUs and 8 GB vRAM)	5000	2	10
vSRX Virtual Firewall (9 vCPUs and 16 GB vRAM, 17 vCPUs and 32 GB vRAM)	10,000	2	10
NFX150	500	1	5

Active Directory as Identity Source Timeout Setting

Here timeout setting refers to firewall authentication forced timeout 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 *firewall* 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 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 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. 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 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 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 timeout configuration is deleted.

  Timeout has no effect on new authentication entries. Timeout remains enforced for existing authentication entries to which it applied before it was deleted. That is, for those authentication entries, the original timeout setting remains in effect.

- The timeout configuration setting is changed.

  The new time-out setting is applied to new incoming authentication entries. Existing entries keep the original, former setting.

- The timeout is set to 0, disabling it.

  If the timeout is set to a new value, that value is assigned to all incoming authentication entries. There is no timeout setting for existing authentication entries.

- The timeout value is not configured.

  - The *firewall* 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.

Configure Active Directory as Identity Source

Example: Configure Active Directory as Identity Source on SRX Series Firewall

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 Firewall

- Junos OS Release 12.1X47-D10 or later for SRX Series Firewalls

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 Firewall. The SRX Series Firewall reads and analyzes the event log of the domain controllers configured in the domain. Thus, the SRX Series Firewall 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 Firewall 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 Firewall 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 Firewall 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.

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

Procedure

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 Firewall 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 Firewall checks the policy configured in Step 4 for further action. If the user firewall does not hit any authentication entry, the SRX Series Firewall 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
```

When the no-tls-certificate-check option is configured, the SRX Series Firewall 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
```

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
```

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).

A unique priority value is assigned to each authentication table. Lower the value, higher is the priority. For example, a table with priority 120 is searched before a table with priority 200. Setting

the priority value of a table to 0 disables the table and eliminates the priority value from the search sequence.

For more details, see Understanding Active Directory Authentication Tables .

(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.

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 Firewall supports captive portal for the traffic type). After the user enters a name and password and passes firewall authentication, the SRX Series Firewall 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 Firewall 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 Firewall 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

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

show services user-identification active-directory-access active-directory-authentication-table

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

show services user-identification active-directory-access statistics

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

Example: Configure Active Directory as Identity Source on SRX Series Firewalls to Use Web-Redirect for Unauthenticated and Unknown Users

IN THIS SECTION

- Requirements | 35
- Overview | 35

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 Firewall

- Junos OS Release 15.1X49-D70 or later for SRX Series Firewalls

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
```

Procedure

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
```

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

Example: Configuring Integrated User Firewall on SRX Series

Example: Configure Active Directory as Identity Source on SRX Series Firewalls to Use Web-Redirect-to-HTTPS to Authenticate Unauthenticated and Unknown Users

IN THIS SECTION

- Requirements | **40**
- Overview | **41**
- Configuration | **42**

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 Firewall'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 Firewall

- Junos OS Release 15.1X49-D70 or later for SRX Series Firewalls

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.
```

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.

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

Procedure

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
```

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.

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

LDAP Functionality in Integrated User Firewall

Example: Configure User Identity Information to Session Log Based On Source Zone

IN THIS SECTION

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 Firewalls 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-close) 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 Firewall previously obtained the user identity information by reading the domain controller event log. The SRX Series Firewall 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

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 | **53**

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. calls out the parts of the log that are specific to the source identity log function configuration:

Table 6: Session Log Components Specific to the Source Identity Log Function

session create This is the session initiation which begins the first section of the log that records the session setup information. 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" source-port="635" destination-address="198.51.100.10/24" destination-port="51"
application service 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	source-zone-name="trust" destination-zone-name="untrust"
Further down the log are the zone specifications which show trust as the source zone and untrust as the destination zone as defined.	
session close	user1 RT_FLOW - RT_FLOW_SESSION_CLOSE
This is the session close initiation, which begins the second part of the log record that covers session tear-down and close.	
The user's name, user1, is displayed at the beginning of the session close record.	

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

command-name

```
<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)

Configure Active Directory as Identity Source on SRX Series Firewall

Table 2 on page 55 describes the steps to configure Active Directory as Identity Source on your SRX Series firewall, vSRX virtual firewall or cSRX container firewall or NFX Series devices.

Table 7: Configure Active Directory as Identity Source

Configuration Step	Command

Step 1: Configure authentication-table
 	
You can configure active directory authentication table.
 	
You can configure priority option.
	Authentication table
	
	`[edit security user-identification authentication source]`
	
	`user@host# set active-directory-authentication-table`
	
	Authentication table priority
	
	`[edit security user-identification authentication source active-directory-authentication-table]`
	
	`user@host# set priority`

Step 2: Configure timeout
 	
You can configure valid authentication entry and invalid authentication entry timeout for entries in the authentication table. The default `authentication-entry-timeout` interval is 30 minutes. To disable timeouts, set the interval to 0.
 	
You can view timeout information for authentication table entries.
	Valid authentication entries
	
	`[edit services user-identification active-directory-access]`
	
	`user@host# set authentication-entry-timeout minutes`
	
	Invalid authentication entries
	
	`[edit services user-identification active-directory-access]`
	
	`user@host# set invalid-authentication-entry-timeout minutes`
	
	View timeout information
	
	`[edit show services user-identification active-directory-access active-directory-authentication-table]`
	
	`user@host# set all extensive`

Table 7: Configure Active Directory as Identity Source *(Continued)*

Configuration Step	Command

Step 3: Configure Windows Event Log Verification and Statistics
 	
You can verify that the authentication table is getting IP address and user information.
 	
You can see statistics about reading the event log.
 	
You can configure firewall authentication as backup to WMIC
	Windows Event Log Verification
	
	`[edit show services user-identification active-directory-access active-directory-authentication-table]`
	
	`user@host# set all`
	
	Windows Event Log Statistics
	
	`[edit show services user-identification active-directory-access ip-user-mapping]`
	
	`user@host# set statistics domain`
	
	Firewall authentication as backup to WMIC
	
	`[edit security policies from-zone trust to-zone untrust policy <policy-name> then permit`
	
	`user@host# set firewall-authentication user-firewall domain <domain-name>`

Table 7: Configure Active Directory as Identity Source *(Continued)*

Configuration Step	Command

Step 4: Configure domain PC probing
 	
On-demand probing is enabled by default. You can disable on-demand probing. When on-demand probing is disabled, manual probing is available.
 	
You can configure probe timeout value. The default timeout is 10 seconds.
 	
You can display probe statistics.
	Disable on-demand probing
	
	`[edit services user-identification active-directory-access]`
	
	`user@host# set no-on-demand-probe`
	
	Enable manual probing
	
	`[edit services user-identification active-directory-access ip-user-probe address ip-address address]`
	
	`user@host# set domain domain-name`
	
	Probe timeout value
	
	`[edit services user-identification active-directory-access]`
	
	`user@host# set wmi-timeout seconds`
	
	Display probe statistics
	
	`[edit show services user-identification active-directory-access]`
	
	`user@host# set statistics ip-user-probe`

Step 5: Configure LDAP Server Status and Statistics
 	
You can verify the LDAP connection status.
 	
You can see counts of queries made to the LDAP server.
	LDAP server status
	
	`[edit show services user-identification active-directory-access]`
	
	`user@host# set user-group-mapping status`
	
	LDAP server statistics
	
	`[edit show services user-identification active-directory-access]`
	
	`user@host# set statistics user-group-mapping`

Configure Active Directory as Identity Source 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.

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
```

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
```

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

SRX Firewall Users

SUMMARY

Learn about the authentication methods: pass-through authentication, captive portal authentication, or pass-through with web-redirect authentication.

A SRX 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. You can store these user group on the local database or on a RADIUS, or LDAP 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.

When you define firewall users, you can create a policy that requires the users to authenticate themselves through one of the authentication methods: pass-through authentication, captive Portal authentication, or pass-through with web-redirect authentication.

Pass-Through Authentication

What is pass-through authentication?

Pass-through authentication helps to authenticate yourself automatically using your currently logged in windows system username and password. You would not need to manually enter your windows credential to log-in.

How a pass-through authentication works?

A user from one zone tries to access resources on another zone. 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 IP address of the protected resource.

The device uses FTP, Telnet, HTTP, or HTTPS to collect username and password information. The device then intercepts the request and prompts the user to enter a username and password. The device validates the username and password by checking them against those stored in the local database or on an external authentication server.

Figure 3: Policy Lookup for a User
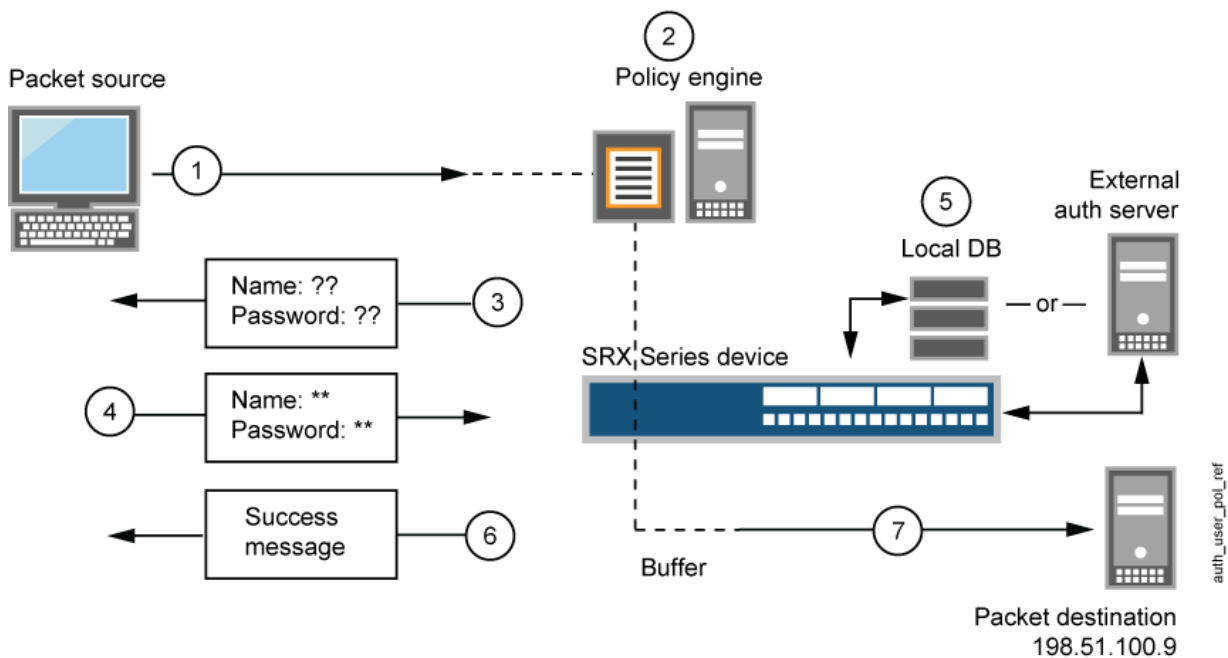
1. A client user sends an FTP, an HTTP, an HTTPS, or a Telnet packet to 198.51.100.9.

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.

Benefits

- Better security.

- Easier deployment.

- Highly available.

- Seamless user experience.

Captive Portal Authentication

What is captive portal authentication?

A captive portal is a webpage that users of a public network view and interact with before they can access the network. The web page also prompts the users to authenticate or accept the usage policy and terms. The captive portal web login page is hosted by an internal or external server.

How a captive portal authentication works?

Users try to connect, using HTTP or HTTPS, to an IP address on the device that is enabled for captive portal authentication. Users don't get to the IP address of the protected resource. This initiates an HTTP session to the IP address hosting the captive portal authentication feature on the device. The device then prompts you for your username and password and caches the result in the device. Subsequent traffic from the user to the protected resource is allowed or denied based on the result of this authentication.

Figure 4: Captive Portal Authentication

1 - 5. The auth user connects to the WebAuth IP Address 198.51.100.63/24 and successfully logs on. (JUNOS checks the auth user account in the specified database - local database or external auth server.)

WebAuth 198.51.100.63/24

HTTP to 198.51.100.63/24

Name: ??
Password: ??

Name: **
Password: **

Success message

External auth server

Local DB

— or —

SRX Series device

203.0.113.19/24

6. The auth user initiates a connection to 203.0.113.19/24.

Policy engine

7. The policy engine notes that the policy applying to this traffic requires authentication via WebAuth. It checks its list of authenticated users, discovers that the user is currently authenticated via WebAuth, and passes the traffic to its destination.

webauth_prepol_check

1. The default captive portal 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.

2. The captive portal 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 captive portal authentication through ethernet3, which has IP address 203.0.113.1/24, then you can assign captive portal authentication an IP address in the 203.0.113.0/24 subnet.

3. You can put a captive portal authentication address in the same subnet as the IP address of any physical interface or virtual security interface (VSI).

4. You can put captive portal authentication addresses on multiple interfaces.

5. After the device authenticates a user at a particular source IP address, it subsequently permits traffic as specified in the policy. This requires 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.

6. With captive portal authentication enabled, any HTTP traffic to the IP address will get the captive portal 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.

7. We recommend that you have a separate primary or preferred IP address, if an address is used for captive portal authentication.

Benefits

- Simpler way to access the internet.

- Adds additional layer of security.

- Marketing and business recognition.

Pass-Through with Web-Redirect Authentication

What is pass-through with web-redirect authentication?

Pass-through with web-redirect authentication 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.

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, browser launches the user's original destination URL without their needing to retype the URL.

Benefits

- Richer user login experience—Instead of a popup prompt asking the user to enter their username and password, users are presented with the login page in a browser.

- Seamless authentication experience—The user doesn't need to know the IP address of the captive portal authentication source but only the IP address of the resource they are attempting to access.

Captive Portal for Unauthenticated Browsers

The SRX Series Firewall 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. This 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.

The service discarded the page without informing the browser, and the browser user was never presented with the authentication portal. The SRX Series Firewall did not support simultaneous authentication from the same source (IP address) on different SPUs.

The SRX Series Firewall 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 Firewall 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 Firewall distinguishes HTTP/HTTPS browser traffic from other HTTP/HTTPS traffic. The SRX Series Firewall 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 Firewall 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.

For more information on how to configure captive portal for unauthenticated browsers, see "Configure Captive Portal for Unauthenticated Browsers" on page 103.

Unified Policy

Unified policy enables you to authenticate users before users can access network resources behind a firewall. When you enable SRX firewall users with unified policy, a user must provide a username and password for authentication when initiating a connection across the firewall.

For more information on how to configure unified policy, see "Example: Configure Unified Policy" on page 106.

Table 8: Unified Policy Workflow

SRX Firewall Users with Unified Policy	Workflow
Pass-through Authentication with a traditional security Policy and an unified policy	• Traditional security policy triggers firewall authentication when FTP, Telnet, HTTP, or HTTPS traffic matches the security policy match criteria. • After successful authentication, the unified policy permits or blocks subsequent traffic that matches the unified policy rules.
Pass-Through Authentication with a Traditional Security Policy and a Unified Policy with Dynamic Application as "any"	• The unified policy enforces firewall authentication based on the pre-defined application such as FTP, Telnet, HTTP, or HTTPS service port as per the dynamic-application configured as "any" in the policy. In case a user sends traffic with other service port, and eventually the traffic could be identified as dynamic-application junos:HTTP, this traffic does not trigger the firewall authentication. • After successful authentication, the unified policy permits or blocks subsequent traffic that matches the unified policy rules.
Captive Portal Authentication with a Unified Policy	• The unified policy enforces firewall authentication when a user opens a browser and enters the IP address of the interface. The interface that users access must be enabled for the captive portal authentication. • After successful authentication, the unified policy permits or blocks subsequent traffic that matches the unified policy rules.

External Authentication Servers

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

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 SRX firewall user authentication, which supports the following types of servers:

- Local authentication and authorization.

- RADIUS authentication and authorization.

- LDAP authentication.

Client Groups

To manage a number of SRX 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 group name 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.

Customize Banner

A banner is a custom message that you can create to indicate a user whether the authentication is successful or failed. A banner is a message that appears on a monitor in different places depending on the type of login.

Figure 5: Custom Banner

- At the top of a browser screen after a user has successfully logged into a captive portal authentication address as shown in Figure 3.

- 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.

Configure Authentication Methods For SRX Firewall Users

SUMMARY

Learn how to configure pass-through and captive portal authentication.

IN THIS SECTION

Example: Configure Pass-Through Authentication

IN THIS SECTION

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 Firewall

- 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 Firewall 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 Firewall if the traffic matches the security policy tuples.

Figure 6 on page 70 shows the topology used in this example.

Figure 6: Configuring Pass-Through Firewall Authentication
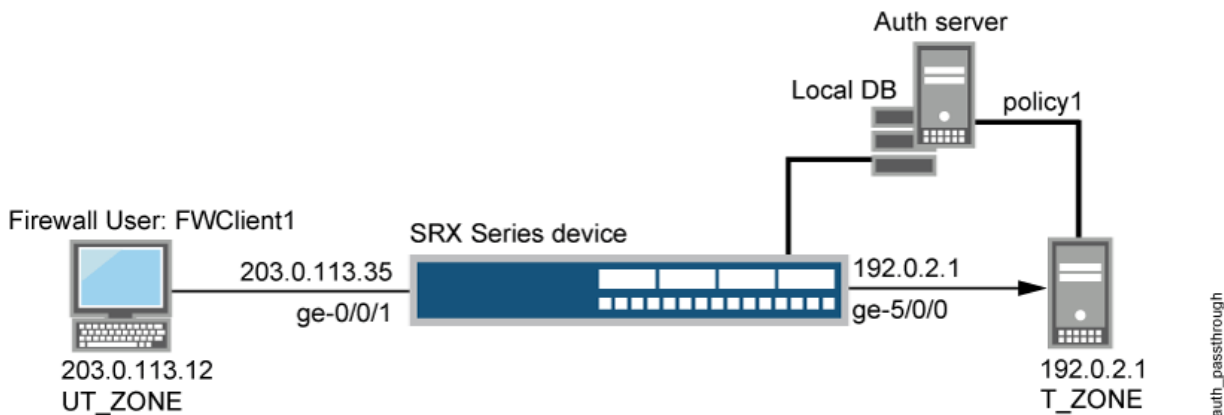
> **NOTE**: Although the topology shows use of an external server, it is not covered in the configuration. It is outside the scope of this example.

Configuration

Procedure

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

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: Configure HTTPS Traffic to Trigger Pass-Through Authentication

IN THIS SECTION

- Requirements | 78
- Overview | 79

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 Firewall

- 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 Virtual Firewall, SRX300, SRX320, SRX340, SRX345, SRX380, 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 Virtual Firewall, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550M, and SRX1500 Services Gateways.

Before you begin:

An SRX Series Firewall 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 Firewall. 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 Firewall, 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 Firewall.

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 7 on page 80 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 Firewall 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 7: Pass-Through Authentication Using HTTPS Traffic
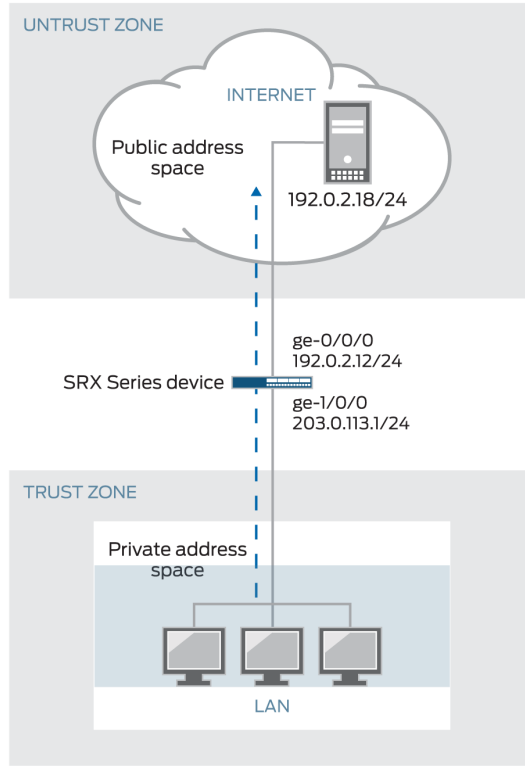
| Original Source IP 203.0.113.1/24 | Translated Source IP 10.1.1.1 - 10.1.1.14 |
| Original Destination IP 1.1.1.100/32 | Translated Destination IP 10.1.1.200/32 |

Configuration

IN THIS SECTION

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
```

Procedure

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.

Example: Configure Captive Portal Authentication

IN THIS SECTION

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

Requirements

Before you begin:

- Define firewall users. See Firewall User Authentication Overview.

- 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 8 on page 88.) In this example, FWClient1 has already authenticated through the Web authentication login page.

The FWClient1 firewall user does the following to get authenticated:

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

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

Figure 8: Web Authentication Example
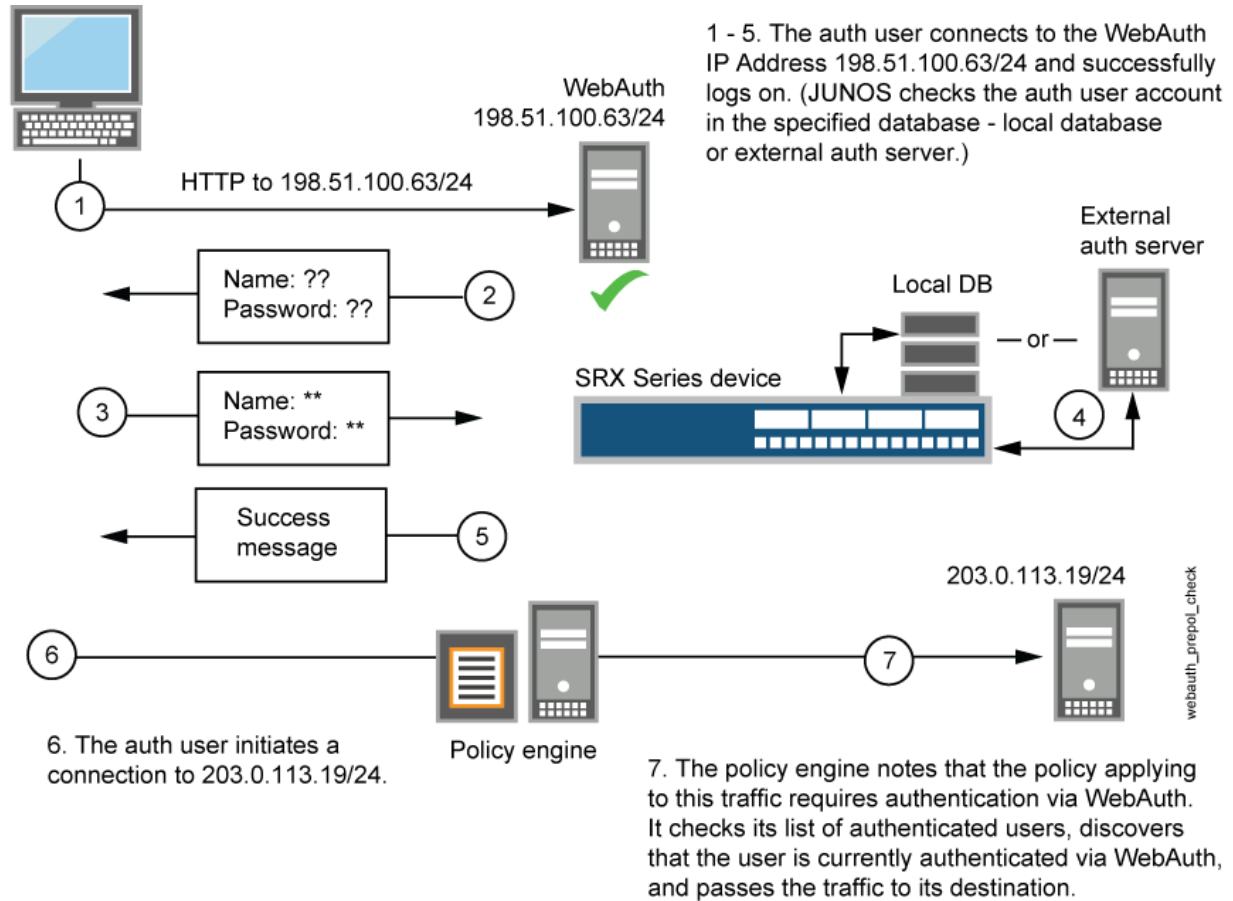
When you configure the device as described in these instructions and the user successfully authenticates, the screen illustrated in appears.

Figure 9: Web Authentication Success Banner
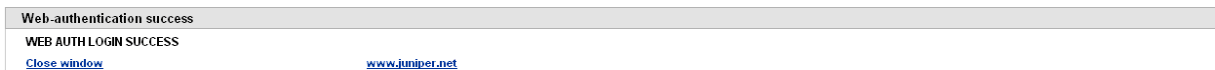
Configuration

Procedure

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

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

Security Zones Overview

Example: Configure HTTPS Traffic to Trigger Captive Portal Authentication

IN THIS SECTION

- Requirements | **96**
- Overview | **97**
- Configuration | **99**
- Verification | **102**

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

Requirements

Before you begin:

This example uses the following hardware and software components:

- SRX Series Firewall

- 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 Virtual Firewall, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550M, and SRX1500 Services Gateways.

An SRX Series Firewall 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 Firewall. 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 Firewall, 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 Firewall.

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 10 on page 98 shows an example of Web authentication using HTTPS traffic.

Figure 10: Web Authentication Using HTTPS Traffic
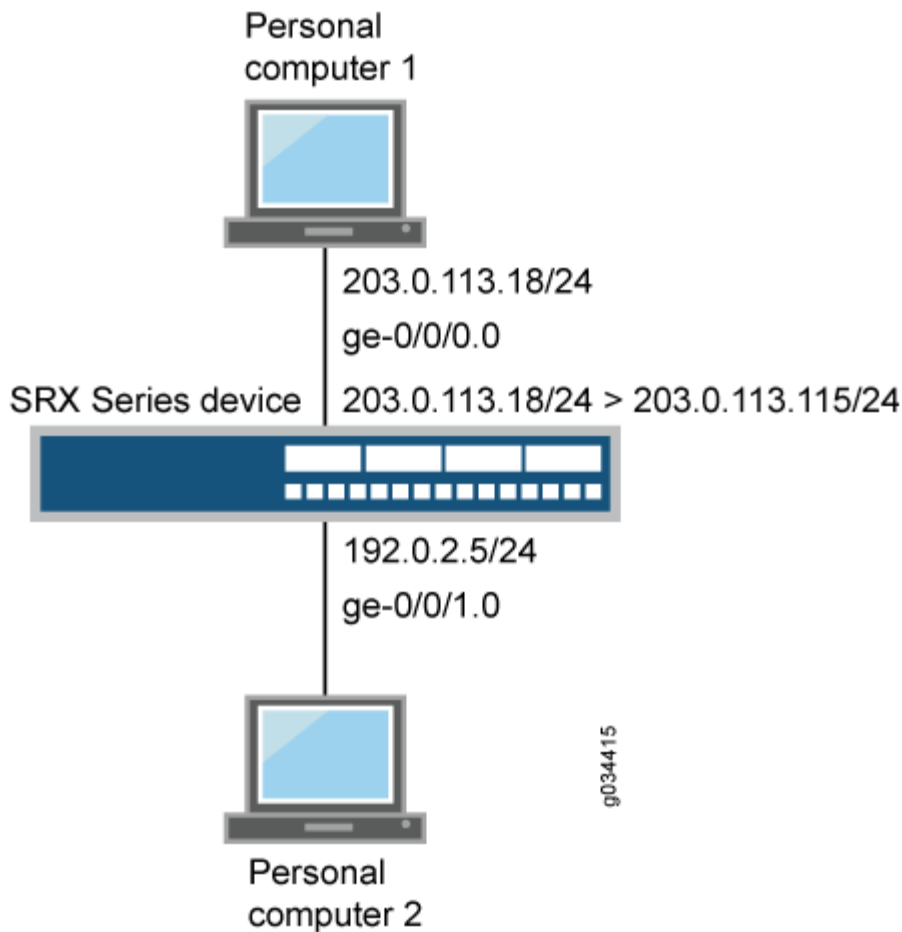
Personal
computer 1

203.0.113.18/24
ge-0/0/0.0

SRX Series device | 203.0.113.18/24 > 203.0.113.115/24

192.0.2.5/24
ge-0/0/1.0

Personal
computer 2

g034415

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
```

Procedure

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

IN THIS SECTION

- Verifying the Configuration | **102**

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.

Configure Captive Portal for Unauthenticated Browsers

SUMMARY

Learn how to configure captive portal for unauthenticated browsers.

Here are some examples of how you can 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

auth-user-agent

Example: Configure Unified Policy

SUMMARY

Read this example to understand how to configure pass-through authentication and captive portal authentication in a unified policy to restrict or permit users to access network resources.

Overview

Firewall user authentication enables you to authenticate users before users can access network resources behind a firewall. When you've enabled firewall user authentication, a user must provide a username and password for authentication when initiating a connection across the firewall.

Starting in Junos OS Release 21.2R1, we support firewall user authentication with unified policies. Support is available for both pass-through authentication and captive portal authentication.

Topology

Figure 6 shows the topology used in this example.

Figure 11: Topology: Configuring Firewall User Authentication with Unified Policy
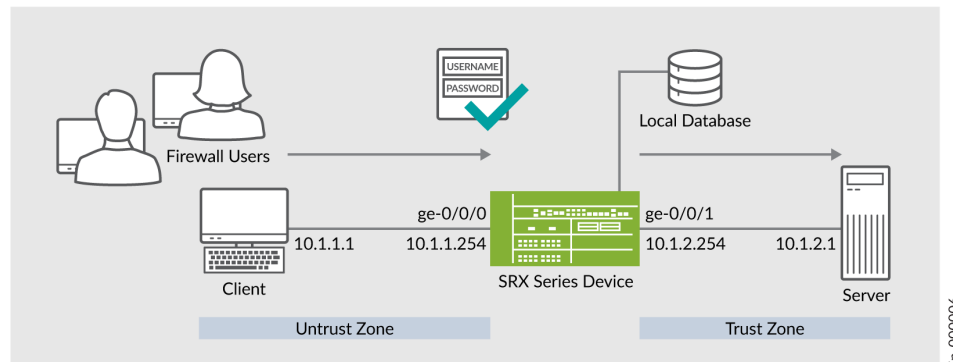
As shown in the topology, firewall users in the untrust zone need to access an external server (IP address 10.1.2.1) in the trust zone. The user authenticates with the security device before accessing the server. The device queries a local database to determine the authentication result. After successful authentication, the security device allows subsequent traffic from the same source IP address until the user's session times out and closes.

In this example, you'll configure the following functionality on the SRX Series Firewall:

1. Configure a user database that is local to the security device in an access profile. Add one or more clients within the profile, representing end users. The client-name represents the username. Enter the password for each user in plain-text format.

2. Associate access profile with pass-through or Web firewall authentication methods. Set a customized banner for display to the end user.

3. Configure security policy to allow or restrict traffic and apply firewall user authentication for the allowed traffic.

Requirements

This example uses the following hardware and software components:

- An SRX Series Firewall or vSRX Virtual Firewall

- Junos OS Release 21.2R1

Before You Begin:

- Install a valid application identification feature license on your SRX Series Firewall. See Installing and Verifying Licenses for an Application Signature Package.

- Install application signature database on the SRX Series Firewall. See Downloading and Installing the Junos OS Application Signature Package.

Configuration of SRX Firewall Users with Traditional Policy and Unified Policy

IN THIS SECTION

- CLI Quick Configuration | **111**
- Step-by-Step Procedure | **112**
- Results | **115**

In this example, we'll configure pass-through authentication with both the traditional security policy and the unified policy. The configuration includes setting up security zones and interfaces, creating access profiles, and defining security policies as shown in the following table:

Table 9: Security Policies Details

Scenarios	Policies	Workflow When User Initiates a Session	Result
Authentication with traditional security policy and unknown user	Policy P1 • **Match criteria:** source-identity - unknown/ unauthenticated users	1. Device searches for the user source identity in the user identification table (UIT). 2. Policy considers the user as an unauthenticated-user if the source identity not available. 3. Policy intercepts HTTP or HTTPS traffic from the user and triggers a firewall authentication prompt. 4. After successful authentication, the policy permits or rejects the traffic based on the configured policy rules. 5. Device creates an authentication entry in the user identification table by including IP address and username.	Permits an unauthenticated user after a successful firewall user authentication.

Table 9: Security Policies Details *(Continued)*

Scenarios	Policies	Workflow When User Initiates a Session	Result
Authentication with unified policy and an authenticated user	Policy P2 • **Match criteria:** source-identity - authenticated-users • dynamic-application - junos:GOOGLE	1. Device retrieves user and role information from the user identification table (UIT) if available. 2. Security policy classifies the user as an authenticated user. 3. After successful authentication, the policy permits or rejects the traffic based on the configured policy rules.	Permits an authenticated user without firewall user authentication.
Authentication with unified policy	Policy P3 • dynamic-application - junos:YAHOO	1. Device searches the authentication profile PROFILE-1 to determine authentication result. 2. After successful authentication, the policy permits or rejects the traffic based on the configured policy rules.	Permits traffic with firewall user authentication.

To redirect the traffic from an unauthenticated-user to a UAC captive portal for authentication, see Example: Configuring a User Role Firewall on an SRX Series Device.

CLI Quick Configuration

To quickly configure this example on your SRX Series Firewall, 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 ssl termination profile ssl-a server-certificate SERVER-CERTIFICATE-1
set security policies from-zone untrust to-zone trust policy p1 match source-address any
set security policies from-zone untrust to-zone trust policy p1 match destination-address any
set security policies from-zone untrust to-zone trust policy p1 match application junos-http
set security policies from-zone untrust to-zone trust policy p1 match application junos-https
set security policies from-zone untrust to-zone trust policy p1 match source-identity
unauthenticated-user
set security policies from-zone untrust to-zone trust policy p1 match source-identity unknown-
user
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall access-profile PROFILE-1
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall ssl-termination-profile ssl-a
set security policies from-zone untrust to-zone trust policy p1 then log session-init
set security policies from-zone untrust to-zone trust policy p1 then log session-close
set security policies from-zone untrust to-zone trust policy p2 match source-address any
set security policies from-zone untrust to-zone trust policy p2 match destination-address any
set security policies from-zone untrust to-zone trust policy p2 match application any
set security policies from-zone untrust to-zone trust policy p2 match source-identity
authenticated-user
set security policies from-zone untrust to-zone trust policy p2 match dynamic-application
junos:GOOGLE
set security policies from-zone untrust to-zone trust policy p2 then permit
set security policies from-zone untrust to-zone trust policy p3 match source-address any
set security policies from-zone untrust to-zone trust policy p3 match destination-address any
set security policies from-zone untrust to-zone trust policy p3 match application any
set security policies from-zone untrust to-zone trust policy p3 match dynamic-application
junos:YAHOO
set security policies from-zone untrust to-zone trust policy p3 then permit firewall-
authentication user-firewall access-profile PROFILE-1
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic system-
services all
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-
services all
```

```
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic protocols all
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.254/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.254/24
set access profile PROFILE-1 client CLIENT-1 client-group GROUP-1
set access profile PROFILE-1 client CLIENT-1 firewall-user password "$ABC123"
set access profile PROFILE-1 client CLIENT-2 client-group GROUP-1
set access profile PROFILE-1 client CLIENT-2 firewall-user password "$ABC123"
set access profile PROFILE-1 session-options client-idle-timeout 10
set access firewall-authentication pass-through default-profile PROFILE-1
set access firewall-authentication web-authentication default-profile PROFILE-1
```

Step-by-Step Procedure

1. Configure interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.254/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.254/24
```

2. Create security zones and assign the interfaces.

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

3. Set up access profile and add user details.

```
[edit]
user@host# set access profile PROFILE-1 client CLIENT-1 client-group GROUP-1
user@host# set access profile PROFILE-1 client CLIENT-1 firewall-user password
"$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"
user@host# set access profile PROFILE-1 client CLIENT-2 client-group GROUP-1
user@host# set access profile PROFILE-1 client CLIENT-2 firewall-user password "$9$/
```

```
Bv59pBIRSleWB17-ws4o"
user@host# set access profile PROFILE-1 session-options client-idle-timeout 10
```

We've added two users CLIENT-1 and CLIENT-2 with passwords and assigned these users to client-group GROUP-1.

4. Configure authentication methods and assign the access profile.

```
[edit]
user@host# set access firewall-authentication pass-through default-profile PROFILE-1
user@host# set access firewall-authentication web-authentication default-profile PROFILE-1
```

5. Configure an SSL termination profile.

```
[edit]
user@host# set services ssl termination profile ssl-a server-certificate SERVER-CERTIFICATE-1
```

6. Configure a security policy to permit unauthenticated users with firewall user authentication.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy p1 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy p1 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy p1 match application
junos-http
user@host# set security policies from-zone untrust to-zone trust policy p1 match application
junos-https
user@host# set security policies from-zone untrust to-zone trust policy p1 match source-
identity unauthenticated-user
user@host# set security policies from-zone untrust to-zone trust policy p1 match source-
identity unknown-user
user@host# set security policies from-zone untrust to-zone trust policy p1 match source-
identity unknown-user
user@host# set security policies from-zone untrust to-zone trust policy p1 then permit
firewall-authentication user-firewall access-profile PROFILE-1
user@host# set security policies from-zone untrust to-zone trust policy p1 then permit
firewall-authentication user-firewall ssl-termination-profile ssl-a
user@host# set security policies from-zone untrust to-zone trust policy p1 then log session-
init
```

```
user@host# set security policies from-zone untrust to-zone trust policy p1 then log session-
close
```

7. Configure a security policy to permit authenticated users without firewall user authentication.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy p2 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy p2 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy p2 match application
any
user@host# set security policies from-zone untrust to-zone trust policy p2 match source-
identity authenticated-user
user@host# set security policies from-zone untrust to-zone trust policy p2 match dynamic-
application junos:GOOGLE
user@host# set security policies from-zone untrust to-zone trust policy p2 then permit
```

8. Configure a security policy to permit the traffic with firewall user authentication.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy p3 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy p3 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy p3 match application
any
user@host# set security policies from-zone untrust to-zone trust policy p3 match dynamic-
application junos:YAHOO
user@host# set security policies from-zone untrust to-zone trust policy p3 then permit
firewall-authentication user-firewall access-profile PROFILE-1
user@host#
```

9. Add an entry to a local authentication table. Note that each entry must include an IP address.

```
user@host> request security user-identification local-authentication-table add user-name
CLIENT-1 ip-address 10.1.1.1
```

Results

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

[edit]

```
user@host# show security policies
from-zone untrust to-zone trust {
    policy p1 {
        match {
            source-address any;
            destination-address any;
            application [ junos-http junos-https ];
            source-identity [ unauthenticated-user unknown-userset unknown-user ];
        }
        then {
            permit {
                firewall-authentication {
                    user-firewall {
                        access-profile PROFILE-1;
                        ssl-termination-profile ssl-a;
                    }
                }
            }
            log {
                session-init;
                session-close;
            }
        }
    }
     policy p2 {
        match {
            source-address any;
            destination-address any;
            application any;
            source-identity authenticated-user;
            dynamic-application junos:GOOGLE;
        }
        then {
            permit;
        }
```

```
        }

    policy p3 {
        match {
            source-address any;
            destination-address any;
            application any;
            dynamic-application junos:YAHOO;
        }
        then {
            permit {
                firewall-authentication {
                    user-firewall {
                        access-profile PROFILE-1;
                    }
                }
            }
        }
    }
}
```

[edit]

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

```
                host-inbound-traffic {
                    system-services {
                        all;
                    }
                    protocols {
                        all;
                    }
                }
            }
        }
    }
```

[edit]

```
user@host# show interfaces
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.1.254/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.2.254/24;
            }
        }
    }
```

[edit]

```
user@host# show access
profile PROFILE-1 {
    client CLIENT-1 {
        client-group GROUP-1;
        firewall-user {
            password "$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"; ## SECRET-DATA
        }
    }
```

```
    client CLIENT-2 {
        client-group GROUP-1;
        firewall-user {
            password "$9$/Bv59pBIRSleWB17-ws4o"; ## SECRET-DATA
        }
    }

    session-options {
        client-idle-timeout 10;
    }
}
firewall-authentication {
    pass-through {
        default-profile PROFILE-1;

    web-authentication {
        default-profile PROFILE-1;
    }
}
```

If you are done configuring the feature on your device, enter `commit` from configuration mode.

Verifying Firewall User Authentication Is Working

To verify that the firewall user authentication is working, open a Web browser on the client machine. Access the server by entering the server IP address 10.1.2.1. The system prompts for the login and password details as shown in .

Figure 12: Pass-Through Authentication Prompt

After successfully entering the credentials, you can access the server.

Configuration of Pass-Through Authentication with Unified Policy

IN THIS SECTION

In this example, we'll configure pass-through authentication with a unified policy. The configuration includes setting up security zones and interfaces, creating access profiles, and defining a unified policy. In the unified policy, we define the match criteria dynamic application as any.

CLI Quick Configuration

To quickly configure this example on your SRX Series Firewall, 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 ssl termination profile ssl-a server-certificate SERVER-CERTIFICATE-1
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.254/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.254/24
set security policies from-zone untrust to-zone trust policy p1 match source-address any
set security policies from-zone untrust to-zone trust policy p1 match destination-address any
set security policies from-zone untrust to-zone trust policy p1 match application any
set security policies from-zone untrust to-zone trust policy p1 match dynamic-application any
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication pass-through access-profile PROFILE-1
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication pass-through ssl-termination-profile ssl-a
set security policies from-zone untrust to-zone trust policy p1 then log session-init
set security policies from-zone untrust to-zone trust policy p1 then log session-close
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic system-
services all
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-
services all
```

```
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic protocols all
set access profile PROFILE-1 client CLIENT-1 client-group GROUP-1
set access profile PROFILE-1 client CLIENT-1 firewall-user password
"$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"
set access profile PROFILE-1 client CLIENT-2 client-group GROUP-1
set access profile PROFILE-1 client CLIENT-2 firewall-user password "$9$/Bv59pBIRSleWB17-ws4o"
set access profile PROFILE-1 session-options client-idle-timeout 10
set access firewall-authentication pass-through default-profile PROFILE-1
set access firewall-authentication web-authentication default-profile PROFILE-1
```

Step-by-Step Procedure

1. Configure interfaces.

   ```
   [edit]
   user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.254/24
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.254/24
   ```

2. Define security zones and assign interfaces.

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

3. Set up access profile and add user details.

   ```
   [edit]
   user@host# set access profile PROFILE-1 client CLIENT-1 client-group GROUP-1
   user@host# set access profile PROFILE-1 client CLIENT-1 firewall-user password
   "$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"
   user@host# set access profile PROFILE-1 client CLIENT-2 client-group GROUP-1
   user@host# set access profile PROFILE-1 client CLIENT-2 firewall-user password "$9$/
   ```

```
Bv59pBIRSleWB17-ws4o"
user@host# set access profile PROFILE-1 session-options client-idle-timeout 10
```

We've added two users CLIENT-1 and CLIENT-2 with passwords and assigned the users to client-group GROUP-1.

4. Configure authentication methods and assign the access profile.

```
[edit]
user@host# set access firewall-authentication pass-through default-profile PROFILE-1
user@host# set access firewall-authentication web-authentication default-profile PROFILE-1
```

5. Configure an SSL termination profile.

```
[edit]
user@host# set services ssl termination profile ssl-a server-certificate SERVER-CERTIFICATE-1
```

6. Configure a security policy with dynamic application as `any`.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy p1 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy p1 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy p1 match application
any
user@host# set security policies from-zone untrust to-zone trust policy p1 match dynamic-
application any
user@host# set security policies from-zone untrust to-zone trust policy p1 then permit
firewall-authentication pass-through access-profile PROFILE-1
user@host# set security policies from-zone untrust to-zone trust policy p1 then permit
firewall-authentication pass-through ssl-termination-profile ssl-a
user@host# set security policies from-zone untrust to-zone trust policy p1 then log session-
init
user@host# set security policies from-zone untrust to-zone trust policy p1 then log session-
close
```

Results

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

[edit]

```
user@host# show security policies]
from-zone untrust to-zone trust {
    policy p1 {
        match {
            source-address any;
            destination-address any;
            application any;
            dynamic-application any;
        }
        then {
            permit {
                firewall-authentication {
                    pass-through {
                        access-profile PROFILE-1;
                        ssl-termination-profile ssl-a;
                    }
                }
            }
            log {
                session-init;
                session-close;
            }
        }
    }
}
```

[edit]

```
user@host# show security zones
security-zone trust {
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                system-services {
```

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

[edit]

```
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.1.254/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.2.254/24;
        }
    }
}
```

[edit]

```
user@host# show access
profile PROFILE-1 {
    client CLIENT-1 {
        client-group GROUP-1;
        firewall-user {
            password "$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"; ## SECRET-DATA
        }
    }
    client CLIENT-2 {
        client-group GROUP-1;
        firewall-user {
            password "$9$/Bv59pBIRSleWB17-ws4o"; ## SECRET-DATA
        }
    }
    session-options {
        client-idle-timeout 10;
    }
}
firewall-authentication {
    pass-through {
        default-profile PROFILE-1;
    }
    web-authentication {
        default-profile PROFILE-1;
    }
}
```

If you are done configuring the feature on your device, enter `commit` from configuration mode.

Verifying Pass-Through Authentication Is Working

To verify that firewall user authentication is working, open a Web browser on the client machine. Access the server by entering server IP address 10.1.2.1. The system prompts for login and password details as shown in .

Figure 13: Pass-Through Authentication Prompt

After successfully entering the credentials, you can access the server.

Configuration of Captive Portal Authentication with Unified Policy

IN THIS SECTION

In this example, we'll configure Captive Portal authentication with a unified policy. The configuration includes setting up security zones and interfaces, creating access profiles, and defining a unified policy. For Captive Portal authentication, we'll define a success banner for HTTP sessions.

CLI Quick Configuration

To quickly configure this example on your SRX Series Firewall, 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 system services web-management http interface ge-0/0/0.0
set system services web-management https system-generated-certificate
set system services web-management https interface ge-0/0/0.0
set security policies from-zone untrust to-zone trust policy p1 match source-address any
set security policies from-zone untrust to-zone trust policy p1 match destination-address any
set security policies from-zone untrust to-zone trust policy p1 match application junos-http
```

```
set security policies from-zone untrust to-zone trust policy p1 match application junos-https
set security policies from-zone untrust to-zone trust policy p1 match dynamic-application
junos:HTTP
set security policies from-zone untrust to-zone trust policy p1 match dynamic-application
junos:SSH
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication web-authentication
set security policies from-zone untrust to-zone trust policy p1 then log session-init
set security policies from-zone untrust to-zone trust policy p1 then log session-close
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic system-
services all
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-
services all
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic protocols all
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.254/24
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.253/24 web-authentication http
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.253/24 web-authentication https
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.254/24
set access profile PROFILE-1 client CLIENT-1 client-group GROUP-1
set access profile PROFILE-1 client CLIENT-1 firewall-user password
"$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"
set access profile PROFILE-1 client CLIENT-2 client-group GROUP-1
set access profile PROFILE-1 client CLIENT-2 firewall-user password "$9$/Bv59pBIRSleWB17-ws4o"
set access profile PROFILE-1 session-options client-idle-timeout 10
set access firewall-authentication pass-through default-profile PROFILE-1
set access firewall-authentication web-authentication default-profile PROFILE-1
set access firewall-authentication web-authentication banner success "WELCOME to JUNIPER HTTP
SESSION"
```

Step-by-Step Procedure

1. Create interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.254/24
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.253/24 web-
authentication http
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.253/24 web-
authentication https
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.254/24
```

Use a secondary IP address for the Web authentication. In this example, we're using 10.1.1.253/24 for web authentication. Note that the secondary IP address must use the same subnet as primary IP address.

2. Create security zones and assign interfaces.

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

3. Enable the interface for the Web authentication.

```
[edit]
user@host# set system services web-management http interface ge-0/0/0.0
user@host# set system services web-management https system-generated-certificate
```

4. Set up access profile and add user details.

```
[edit]
user@host# set access profile PROFILE-1 client CLIENT-1 client-group GROUP-1
user@host# set access profile PROFILE-1 client CLIENT-1 firewall-user password
"$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"
user@host# set access profile PROFILE-1 client CLIENT-2 client-group GROUP-1
user@host# set access profile PROFILE-1 client CLIENT-2 firewall-user password "$9$/
Bv59pBIRSleWB17-ws4o"
user@host# set access profile PROFILE-1 session-options client-idle-timeout 10
```

We've added two users CLIENT-1 and CLIENT-2 with passwords and assigned the users to client-group GROUP-1.

5. Configure Web authentication properties

```
[edit]
user@host# set access firewall-authentication web-authentication default-profile PROFILE-1
```

```
user@host# set access firewall-authentication web-authentication banner success "WELCOME to
JUNIPER HTTP SESSION"
```

6. Create a security policy with dynamic-application.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy p1 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy p1 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy p1 match application
junos-http
user@host# set security policies from-zone untrust to-zone trust policy p1 match application
junos-https
user@host# set security policies from-zone untrust to-zone trust policy p1 match dynamic-
application junos:HTTP
user@host# set security policies from-zone untrust to-zone trust policy p1 then permit
firewall-authentication web-authentication
user@host# set security policies from-zone untrust to-zone trust policy p1 then log session-
init
user@host# set security policies from-zone untrust to-zone trust policy p1 then log session-
close
```

Results

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

[edit]

```
user@host# show security policies
from-zone untrust to-zone trust {
    policy p1 {
        match {
            source-address any;
            destination-address any;
            application [ junos-http junos-https ];
            dynamic-application [ junos:HTTP junos:SSH ];
        }
        then {
```

```
            permit {
                firewall-authentication {
                    web-authentication;
                }
            }
            log {
                session-init;
                session-close;
            }
        }
    }
}
```

[edit]

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

```
        }
    }
```

[edit]

```
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.1.254/24;
            address 10.1.1.253/24 {
                web-authentication {
                    http;
                    https;
                }
            }
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.2.254/24;
        }
    }
}
```

[edit]

```
user@host# show access
profile PROFILE-1 {
    client CLIENT-1 {
        client-group GROUP-1;
        firewall-user {
            password "$9$2ngZjHkPQ39.PhrvLVb.P5Tz6"; ## SECRET-DATA
        }
    }
    client CLIENT-2 {
        client-group GROUP-1;
        firewall-user {
            password "$9$/Bv59pBIRSleWB17-ws4o"; ## SECRET-DATA
```

```
            }
        }
        session-options {
            client-idle-timeout 10;
        }
    }
    firewall-authentication {
        pass-through {
            default-profile PROFILE-1;
            }
        }
        web-authentication {
            default-profile PROFILE-1;
            banner {
                success "WELCOME to JUNIPER HTTP SESSION";
            }
        }
    }
```

[edit]

```
user@host# show system services
ssh {
    root-login allow;
}
web-management {
    http {
        interface [ fxp0.0 ge-0/0/0.0 ];
    }
    https {
        system-generated-certificate;
        interface [ fxp0.0 ge-0/0/0.0 ];
    }
}
```
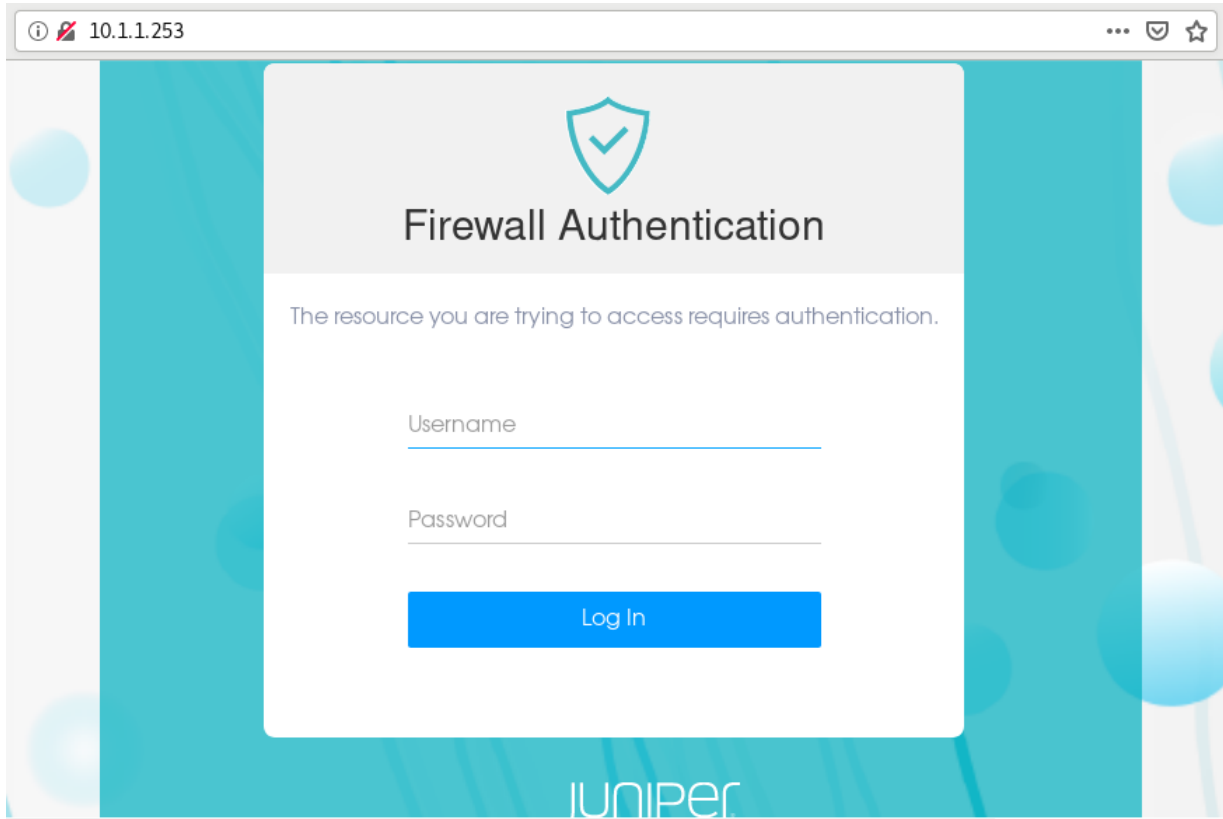
If you are done configuring the feature on your device, enter `commit` from configuration mode.

Verifying Web Authentication Is Working

To verify that Web authentication is working, open a Web browser on the client machine. First, access the security device using a Web browser. Use the IP address 10.1.1.253 which we've configured for

Web authentication. The device prompts for a username and password as shown in .

Figure 14: Web Authentication Prompt

After successful authentication, the system displays the configured banner as shown in , and you can get access to the server.

Figure 15: Web Authentication Banner
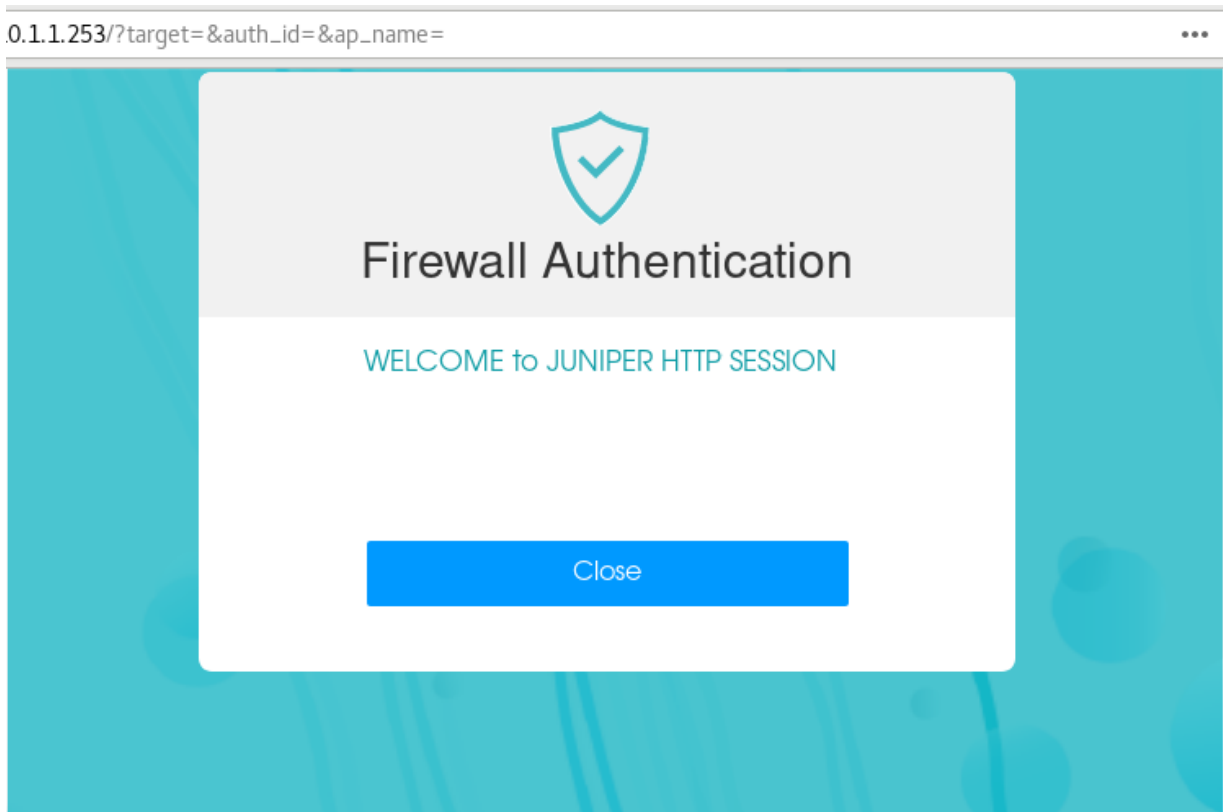
Verification

Monitoring Firewall Users

Purpose

Display firewall authentication user history to verify the firewall users details.

Action

From operational mode, enter these show commands:

```
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
        15 10.1.1.1                       N/A      N/A      PROFILE-    1 Success  CLIENT-2
```

user@host> show security firewall-authentication users identifier 16
Username: CLIENT-2
Source IP: 10.1.1.1
Authentication state: Success
Authentication method: User-firewall using HTTP
Age: 1
Access time remaining: 9
Lsys: root-logical-system
Source zone: N/A
Destination zone: N/A
Access profile: PROFILE-1
Interface Name: ge-0/0/0.0
Bytes sent by this user: 56986
Bytes received by this user: 436401
Client-groups: GROUP-1

lab@vSRX-01> **show security firewall-authentication users identifier 15**
Username: CLIENT-2
Source IP: 10.1.1.1
Authentication state: Success
Authentication method: Web-authentication using HTTP
Age: 2
Access time remaining: 8
Lsys: root-logical-system
Source zone: N/A
Destination zone: N/A
Access profile: PROFILE-1
Interface Name: ge-0/0/0.0
Bytes sent by this user: 0
Bytes received by this user: 0
Client-groups: GROUP-1

user@host> show security firewall-authentication history
History of firewall authentication data:
  Authentications: 2
        Id Source Ip   Date   Time   Duration Status   User

```
        0 10.1.1.1                                2021-05-12 06:44:26 0:00:59    Failed
       14 10.1.1.1                                2021-05-12 07:33:43 0:10:00    Success  CLIENT-2
```

Meaning

Command output provides details such as logged in users, authentication method used, profile applied, login attempts and so on.

Verifying Security Policy Utilization Details

Purpose

Display the utility rate of security policies according to the number of hits received.

Action

From operational mode, enter these show commands:

```
user@host> show security policies hit-count
Logical system: root-logical-system
Index   From zone       To zone         Name            Policy count  Action
1       untrust         trust           p2              2             Permit
```

Meaning

Command output provides details on the security policies applied on the traffic.

Example: Configure External Authentication Servers

IN THIS SECTION

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.

> (i) **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

IN THIS SECTION

- Procedure | **136**

Procedure

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 | **140**

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.

Example: Configure Client Groups

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

Procedure

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 | **143**

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.

Example: Customize Banner

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

Procedure

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.

Example: Configure Mutual-TLS (mTLS) Authentication

SUMMARY

Learn how to configure mutual-TLS (mTLS) authentication.

IN THIS SECTION

Use this example to configure and verify mutual-Transport Layer Security (mTLS) authentication on your firewall. In this example, we use *firewall* to refer to a Juniper Networks® SRX Series Firewall or a Juniper Networks® vSRX Virtual Firewall (vSRX3.0). With this configuration, an user can authenticate without a password. User authentication happens through validation of the client/server certificates with the help of a public-private key pair.

To configure mTLS as shown in this example, an administrator must generate the following certificates:

- CA certificate—Run the CA certificate on your firewall and client browser.

- Server certificate—Generate a server certificate on your firewall by using the domain1.com mTLS server. Sign the server certificate with a CA certificate configured on your firewall.

- Client certificate—Generate a client certificate on your client browser and sign the client certificate with a CA certificate configured on your firewall.

> **TIP**:
> **Table 10: Estimated Timers**
>
> | | |
> |---|---|
> | Reading Time | Less than an hour. |
> | Configuration Time | Less than an hour. |

Example Prerequisites

Table 11: Requirements

Hardware requirements	Juniper Networks® SRX Series Firewall or Juniper Networks® vSRX Virtual Firewall (vSRX3.0)

Software requirements	Junos OS Release 23.4R1 or later

Before You Begin

Table 12: Let's Get Started

Benefits	With mTLS authentication, you can: • Ensure passwordless login for secure connection between a user and a server. • Provide an additional layer of security for users who log in to an organization's network or applications. • Verify the connection between the firewall and any user device that does not follow a login process. • Ensure API requests come from legitimate users and block any malicious API requests.
Know more	Identity Aware Firewall
Learn more	Firewall User Authentication

Functional Overview

This section provides a summary of the configuration components in this example.

Table 13: Configuration and Verification Details

| Technologies used | To establish the mTLS authentication, you must configure: Security zone—Configure two security zones to segregate the traffic. `untrust` `trust` Security policy—Configure the security policies `p1` and `p2` to permit unauthenticated and authenticated users, respectively. Use these policies to select and move data traffic from the `untrust` zone to the `trust` zone. Access profile—Configure the access profile `profile1` and add user1 details. Assign the user to the client groups `group1` and `group2`. mTLS profile—Configure the mTLS profile `ma2` to authenticate the client and the server. |
| Primary verification tasks | Verify mTLS authentication. |

Topology Overview

In this example, a client connects to a server through a firewall. In mTLS authentication, the client and the server verify each other's certificate by exchanging information over an encrypted TLS connection.

The firewall redirects unauthenticated clients to domain1.com upon connection to the server. This process avoids certificate errors because the CA certificate and server certificate for domain1.com are pre-installed on the firewall. The CA certificate is pre-installed on the client's browser.

Use mTLS authentication to bypass manual entry of user credentials for captive portal authentication. Ensure a valid user is configured against the Lightweight Directory Access Protocol (LDAP) profile to retrieve user information and authorization from Active Directory. When firewall authentication is applied in the policy, JIMS configuration is required.

Table 14: Devices, Roles, and Functions Used in This Configuration

Hostname	Role	Function
Client	Service requester	Initiates session with server through the SRX Series Firewall.

Table 14: Devices, Roles, and Functions Used in This Configuration *(Continued)*

Hostname	Role	Function
SRX Series Firewall	Firewall	Encrypts and decrypts packets for the client.
Server	Server	Responds to a client's request.
Active Directory	Identity source	*Active Directory as Identity Source* defines the integration of SRX Series Firewall, vSRX Virtual Firewall, Juniper Networks® cSRX Container Firewall, or Juniper Networks® NFX Series Network Services Platform with Microsoft Windows Active Directory. For more information, see Active Directory as Identity Source.
JIMS	Windows service application	Juniper® Identity Management Service (JIMS) is a Windows service application designed to collect and manage user, device, and group information from Active Directory domains. For more information, see JIMS with SRX Series Firewall.

Topology Illustration

Figure 16: Mutual-TLS (mTLS) Authentication
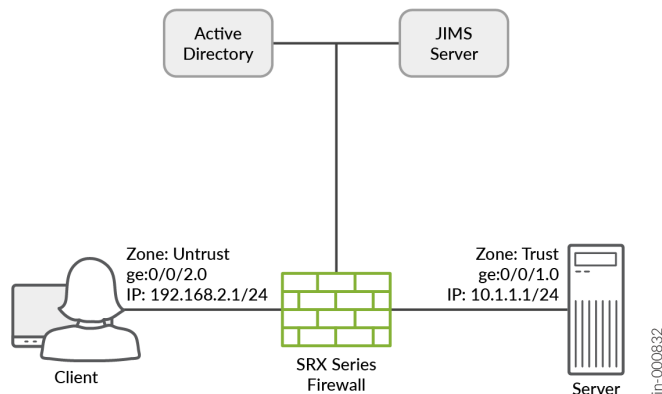
Step-By-Step Configuration on Device-Under-Test (DUT)

> **ℹ NOTE**: For complete sample configurations on the DUT, see:
> - "Appendix 1: set Commands on All Devices" on page 153

1. Configure the required interfaces.

```
set interfaces ge-0/0/2 unit 0 family inet address 192.168.2.1/24 web-authentication https
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24
```

2. Configure the security zones and assign interfaces to the zones.

```
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services tcp-encap
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone trust interfaces ge-0/0/1.0
```

3. Configure an access profile.

```
set access profile profile1 client user1 client-group group1
set access profile profile1 client user1 client-group group2
set access profile profile1 client user1 firewall-user password "$9$dPb4ZjHmzF/k.uO"
```

4. Configure a security policy to permit unauthenticated users with firewall user authentication.

```
set security policies from-zone untrust to-zone trust policy p1 match source-address any
set security policies from-zone untrust to-zone trust policy p1 match destination-address any
set security policies from-zone untrust to-zone trust policy p1 match application any
set security policies from-zone untrust to-zone trust policy p1 match source-identity
unauthenticated-user
set security policies from-zone untrust to-zone trust policy p1 match source-identity unknown-
user
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall access-profile profile1
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall web-redirect-to-https
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall web-authentication-server domain1.com:8443
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication push-to-identity-management
set security policies from-zone untrust to-zone trust policy p1 then log session-close
```

5. Configure a security policy to permit authenticated users without firewall user authentication.

```
set security policies from-zone untrust to-zone trust policy p2 match source-address any
set security policies from-zone untrust to-zone trust policy p2 match destination-address any
set security policies from-zone untrust to-zone trust policy p2 match application any
set security policies from-zone untrust to-zone trust policy p2 match source-identity
authenticated-user
set security policies from-zone untrust to-zone trust policy p2 then permit
set security policies from-zone untrust to-zone trust policy p2 then log session-close
```

6. Configure a ca-profile.

```
set security pki ca-profile ca_domain1
set security pki ca-profile ca_domain1 ca-identity ca_domain1_id
```

7. Configure an mTLS profile.

```
set security firewall-authentication mtls-profile ma2 subject CN=test1client.*
set security firewall-authentication mtls-profile-fallback-password
```

8. Configure web-management to start mTLS on the firewall on which you run the domain1.com server certificate.

```
set system services web-management https interface ge-0/0/2.0
set system services web-management https pki-local-certificate srx_domain1.com
set system services web-management https virtual-domain domain1.com pki-local-certificate
srx_domain1.com
set system services web-management https virtual-domain domain1.com mtls port 8443
set system services web-management https virtual-domain domain1.com mtls ca-profile ca_domain1
set system services web-management https virtual-domain domain1.com mtls firewall-
authentication-profile ma2
```

9. (Optional) Configure a certificate revocation list (CRL) for certificate validation. mTLS supports CRL validation of the incoming certificate. See Certificate Revocation.

```
set security pki ca-profile ca_domain1 revocation-check use-crl
set security pki ca-profile ca_domain1 revocation-check crl url http://<crl-server-ip>/
ca_crl.crl
```

Appendix 1: set Commands on All Devices

```
set interfaces ge-0/0/2 unit 0 family inet address 192.168.2.1/24 web-authentication https
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24

set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services tcp-encap
```

```
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone trust interfaces ge-0/0/1.0


set access profile profile1 client user1 client-group group1
set access profile profile1 client user1 client-group group2
set access profile profile1 client user1 firewall-user password "$9$dPb4ZjHmzF/k.uO"


set security policies from-zone untrust to-zone trust policy p1 match source-address any
set security policies from-zone untrust to-zone trust policy p1 match destination-address any
set security policies from-zone untrust to-zone trust policy p1 match application any
set security policies from-zone untrust to-zone trust policy p1 match source-identity
unauthenticated-user
set security policies from-zone untrust to-zone trust policy p1 match source-identity unknown-
user
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall access-profile profile1
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall web-redirect-to-https
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication user-firewall web-authentication-server domain1.com:8443
set security policies from-zone untrust to-zone trust policy p1 then permit firewall-
authentication push-to-identity-management
set security policies from-zone untrust to-zone trust policy p1 then log session-close



set security policies from-zone untrust to-zone trust policy p2 match source-address any
set security policies from-zone untrust to-zone trust policy p2 match destination-address any
set security policies from-zone untrust to-zone trust policy p2 match application any
set security policies from-zone untrust to-zone trust policy p2 match source-identity
authenticated-user
set security policies from-zone untrust to-zone trust policy p2 then permit
set security policies from-zone untrust to-zone trust policy p2 then log session-close

set security pki ca-profile ca_domain1
set security pki ca-profile ca_domain1 ca-identity ca_domain1_id

set security firewall-authentication mtls-profile ma2 subject CN=test1client.*
set security firewall-authentication mtls-profile-fallback-password

set system services web-management https interface ge-0/0/2.0
set system services web-management https pki-local-certificate srx_domain1.com
set system services web-management https virtual-domain domain1.com pki-local-certificate
srx_domain1.com
```

```
set system services web-management https virtual-domain domain1.com mtls port 8443
set system services web-management https virtual-domain domain1.com mtls ca-profile ca_domain1
set system services web-management https virtual-domain domain1.com mtls firewall-authentication-
profile ma2

set security pki ca-profile ca_domain1 revocation-check use-crl
set security pki ca-profile ca_domain1 revocation-check crl url http://<crl-server-ip>/ca_crl.crl
```

Generate Key Certificates for Client and Server

```
Goal :
1. Generate CA certificate.
2. Generate server cert for srx_domain1.com domain and sign it with CA cert and load it in SRX.
3. Generate client cert sign it with CA cert and load it in client browser.


---------------------------
1. Generate CA certificate :
---------------------------


First you need to set up CA, and then you sign server and client certificates.
Below steps will help creating the certificates in the Linux machine where openssl is installed.


To create CA certificate, create a basic configuration file:


$ touch openssl-ca.cnf,   Then, add the following to it:


## Begining of file


HOME            = .
RANDFILE        = $ENV::HOME/.rnd


####################################################################
[ ca ]
default_ca    = CA_default      # The default ca section


[ CA_default ]


default_days    = 365           # How long to certify for
default_crl_days = 30           # How long before next CRL
default_md       = sha256       # Use public key default MD
```

```
preserve          = no              # Keep passed DN ordering


x509_extensions = ca_extensions # The extensions to add to the cert


email_in_dn      = no              # Don't concat the email in the DN
copy_extensions = copy             # Required to copy SANs from CSR to cert


####################################################################
[ req ]
default_bits       = 4096
default_keyfile    = cakey.pem
distinguished_name = ca_distinguished_name
x509_extensions    = ca_extensions
string_mask        = utf8only


####################################################################
[ ca_distinguished_name ]
countryName         = Country Name (2 letter code)
countryName_default = US


stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = Maryland


localityName                = Locality Name (eg, city)
localityName_default        = MyLocality


organizationName            = Organization Name (eg, company)
organizationName_default    = Test CA, Limited


organizationalUnitName         = Organizational Unit (eg, division)
organizationalUnitName_default = Server Research Department


commonName         = Common Name (e.g. server FQDN or YOUR name)
commonName_default = MYCA


emailAddress         = Email Address
emailAddress_default = test@example.com


####################################################################
[ ca_extensions ]


subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always, issuer
```

```
basicConstraints       = critical, CA:true
keyUsage               = keyCertSign, cRLSign


## End of file.
```

Then, execute the following. The -nodes omits the password or passphrase so you can examine the certificate.

```
$ openssl req -x509 -config openssl-ca.cnf -days 365 -newkey rsa:4096 -sha256 -nodes -out
cacert.pem -outform PEM
```

After the command executes, cacert.pem will be your certificate for CA operations, and cakey.pem will be the private key.

You can verify the certificate with the following command :

```
$ openssl x509 -in cacert.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 114858309707703032316 (0x9f65de69ceef2ffc)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=MD, L=MyLocality, CN=Test CA/emailAddress=test@example.com
        Validity
            Not Before: Jan 24 14:24:11 2014 GMT
            Not After : Feb 23 14:24:11 2014 GMT
        Subject: C=US, ST=MD, L=MyLocality, CN=Test CA/emailAddress=test@example.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:b1:7f:29:be:78:02:b8:56:54:2d:2c:ec:ff:6d:

                    ...

                    39:f9:1e:52:cb:8e:bf:8b:9e:a6:93:e1:22:09:8b:
                    59:05:9f
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                4A:9A:F3:10:9E:D7:CF:54:79:DE:46:75:7A:B0:D0:C1:0F:CF:C1:8A
            X509v3 Authority Key Identifier:
                keyid:4A:9A:F3:10:9E:D7:CF:54:79:DE:46:75:7A:B0:D0:C1:0F:CF:C1:8A

            X509v3 Basic Constraints: critical
```

```
                CA:TRUE
            X509v3 Key Usage:
                Certificate Sign, CRL Sign
    Signature Algorithm: sha256WithRSAEncryption
        4a:6f:1f:ac:fd:fb:1e:a4:6d:08:eb:f5:af:f6:1e:48:a5:c7:
        ...
        cd:c6:ac:30:f9:15:83:41:c1:d1:20:fa:85:e7:4f:35:8f:b5:
        38:ff:fd:55:68:2c:3e:37
```

You can also test its purpose with following command :

```
$ openssl x509 -purpose -in cacert.pem -inform PEM
Certificate purposes:
SSL client : No
SSL client CA : Yes
SSL server : No
SSL server CA : Yes
Netscape SSL server : No
Netscape SSL server CA : Yes
S/MIME signing : No
S/MIME signing CA : Yes
S/MIME encryption : No
S/MIME encryption CA : Yes
CRL signing : Yes
CRL signing CA : Yes
Any Purpose : Yes
Any Purpose CA : Yes
OCSP helper : Yes
OCSP helper CA : Yes
Time Stamp signing : No
Time Stamp signing CA : Yes
-----BEGIN CERTIFICATE-----
MIIFpTCCA42gAwIBAgIJAJ9l3mnO7y/8MA0GCSqGSIb3DQEBCwUAMGExCzAJBgNV
...
aQUtFrV4hpmJUaQZ7ySr/RjCb4KYkQpTkOtKJOU1Ic3GrDD5FYNBwdEg+oXnTzWP
tTj//VVoLD43
-----END CERTIFICATE-----
```

```
Load CA cert on SRX :
router# set security pki ca-profile ca_domain1 ca-identity ca_domain1_id
router# run request security pki ca-certificate load ca-profile ca_domain1 filename /var/tmp/
```

```
cacert.pem


Load CA cert on client browser :  ca_cert.pem  in 'authorities' section of client browser
certificates [Can also be done after step3].


--------------------------------------------------------------------------
2. Generate server cert for srx_domain1.com domain and sign it with CA cert :
--------------------------------------------------------------------------
First, touch the openssl-server.cnf (you can make one of these for user certificates also) :


$ touch openssl-server.cnf
Then open it, and add the following.
## Begining of file.


HOME            = .
RANDFILE        = $ENV::HOME/.rnd


######################################################################
[ req ]
default_bits       = 2048
default_keyfile    = serverkey.pem
distinguished_name = server_distinguished_name
req_extensions     = server_req_extensions
string_mask        = utf8only


######################################################################
[ server_distinguished_name ]
countryName         = Country Name (2 letter code)
countryName_default = US


stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = MD


localityName         = Locality Name (eg, city)
localityName_default = MyLocality


organizationName         = Organization Name (eg, company)
organizationName_default     = Test Server, Limited


commonName         = Common Name (e.g. server FQDN or YOUR name)
commonName_default   = srx_domain1
```

```
emailAddress        = Email Address
emailAddress_default = srx_domain1@srx_domain1.com


#####################################################################
[ server_req_extensions ]


subjectKeyIdentifier = hash
basicConstraints     = CA:FALSE
keyUsage             = digitalSignature, keyEncipherment
subjectAltName       = @alternate_names
nsComment            = "OpenSSL Generated Certificate"


#####################################################################
[ alternate_names ]


DNS.1  = srx_domain1.com
DNS.2  = www.srx_domain1.com
DNS.3  = mail.srx_domain1.com
DNS.4  = ftp.srx_domain1.com


# IPv4 localhost
IP.1    = 127.0.0.1


# IPv6 localhost
IP.2    = ::1


## End of file.
```

Now, create the server certificate request using below command.

```
$ openssl req -config openssl-server.cnf -newkey rsa:2048 -sha256 -nodes -out servercert.csr -
outform PEM
```
After this command executes, you will have a request in servercert.csr and a private key in serverkey.pem.

Verify the created certificate with below command :

```
$ openssl req -text -noout -verify -in servercert.csr
Certificate:
    verify OK
    Certificate Request:
        Version: 0 (0x0)
        Subject: C=US, ST=MD, L=MyLocality, CN=srx_domain1/
```

```
emailAddress=srx_domain1@srx_domain1.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ce:3d:58:7f:a0:59:92:aa:7c:a0:82:dc:c9:6d:

                    ...

                    f9:5e:0c:ba:84:eb:27:0d:d9:e7:22:5d:fe:e5:51:
                    86:e1
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Subject Key Identifier:
                1F:09:EF:79:9A:73:36:C1:80:52:60:2D:03:53:C7:B6:BD:63:3B:61
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Subject Alternative Name:
                DNS:example.com, DNS:www.example.com, DNS:mail.example.com, DNS:ftp.example.com
            Netscape Comment:
                OpenSSL Generated Certificate
    Signature Algorithm: sha256WithRSAEncryption
        6d:e8:d3:85:b3:88:d4:1a:80:9e:67:0d:37:46:db:4d:9a:81:

        ...

        76:6a:22:0a:41:45:1f:e2:d6:e4:8f:a1:ca:de:e5:69:98:88:
        a9:63:d0:a7
```

Next, you have to sign it with your CA.

First, open openssl-ca.cnf and add the following two sections.

```
####################################################################
[ signing_policy ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

```
####################################################################
[ signing_req ]
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid,issuer
basicConstraints       = CA:FALSE
keyUsage               = digitalSignature, keyEncipherment
```

Second, add the following to the [CA_default] section of openssl-ca.cnf. I left them out earlier, because they can complicate things (they were unused at the time). Now you'll see how they are used, so hopefully they will make sense.

```
base_dir      = .
certificate   = $base_dir/cacert.pem   # The CA certificate
private_key   = $base_dir/cakey.pem    # The CA private key
new_certs_dir = $base_dir              # Location for new certs after signing
database      = $base_dir/index.txt    # Database index file
serial        = $base_dir/serial.txt   # The current serial number

unique_subject = no  # Set to 'no' to allow creation of
                     # several certificates with same subject.
```

Third, touch index.txt and serial.txt:

```
$ touch index.txt
$ echo '01' > serial.txt
```

Then, perform the following:

```
$ openssl ca -config openssl-ca.cnf -policy signing_policy -extensions signing_req -out
servercert.pem -infiles servercert.csr
```

You should see similar to the following:

```
Using configuration from openssl-ca.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :ASN.1 12:'MD'
localityName          :ASN.1 12:'MyLocality'
commonName            :ASN.1 12:'Test CA'
emailAddress          :IA5STRING:'test@example.com'
Certificate is to be certified until Oct 20 16:12:39 2016 GMT (1000 days)
Sign the certificate? [y/n]:Y


1 out of 1 certificate requests certified, commit? [y/n]Y
```

```
Write out database with 1 new entries
Data Base Updated
After the command executes, you will have a freshly minted server certificate in servercert.pem.
The private key was created earlier and is available in serverkey.pem.


Finally, you can inspect your freshly minted certificate with the following:


$ openssl x509 -in servercert.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 9 (0x9)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=MD, L=MyLocality, CN=Test CA/emailAddress=test@srx_domain1.com
        Validity
            Not Before: Jan 24 19:07:36 2014 GMT
            Not After : Oct 20 19:07:36 2016 GMT
        Subject: C=US, ST=MD, L=MyLocality, CN=Test Server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ce:3d:58:7f:a0:59:92:aa:7c:a0:82:dc:c9:6d:

                    ...

                    f9:5e:0c:ba:84:eb:27:0d:d9:e7:22:5d:fe:e5:51:
                    86:e1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                1F:09:EF:79:9A:73:36:C1:80:52:60:2D:03:53:C7:B6:BD:63:3B:61
            X509v3 Authority Key Identifier:
                keyid:42:15:F2:CA:9C:B1:BB:F5:4C:2C:66:27:DA:6D:2E:5F:BA:0F:C5:9E


            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Subject Alternative Name:
                DNS:example.com, DNS:www.example.com, DNS:mail.example.com, DNS:ftp.example.com
            Netscape Comment:
                OpenSSL Generated Certificate
    Signature Algorithm: sha256WithRSAEncryption
        b1:40:f6:34:f4:38:c8:57:d4:b6:08:f7:e2:71:12:6b:0e:4a:
```

```
        ...
        45:71:06:a9:86:b6:0f:6d:8d:e1:c5:97:8d:fd:59:43:e9:3c:
        56:a5:eb:c8:7e:9f:6b:7a
```

Please refer to openssl documentation for copy_extension, unique_subject, policy_match etc and
add those the file if needed.
This document describes only the basic key generation steps.

Load sever certificate on SRX :
router# run request security pki local-certificate load filename server_CASignedcert.pem key
server_key.pem certificate-id srx_domain1

```
--------------------------------------------------------
3. Generate client cert and load it in client browser :
--------------------------------------------------------
```
Follow the steps in step-2 and generate client certificate.
In case you need to connect CN of the certificate to the username from Active-directory - where
user's role can be fetched, ensure that client CN is test1client@srx_domain1.com,
where, Active-directory already has a user with name test1client in domain srx_domain1.com.

Active-directory entry :
PS C:\Users\Administrator> Get-ADUser -Filter {SamAccountName -like "*test*"}
DistinguishedName : CN=test1client N,DC=srx_domain1,DC=com
Enabled   : True
GivenName   : test1Client
Name   : test1client N
ObjectClass   : user
ObjectGUID   : a28777ef-0023-45f1-a192-147eff664cbd
SamAccountName   : test1client
UserPrincipalName : test1client@srx_domain1.com

After the client certificate is generated in pem format, convert it in pkcs format :
openssl pkcs12 -export -out client_cert.p12 -in clientcert.pem -inkey clientkey.pem -passin
pass:root -passout pass:root

```
Now, on  client machine, on web browser, load client_cert.p12 in 'your certificates' section.
Also, load ca_cert.pem  in 'authorities' section of browser certificates.
```

Verification

IN THIS SECTION

- Verify mTLS authentication | **165**

This section provides a list of show commands that you can use to verify the feature in this example.

Verify mTLS authentication

Purpose

Verify the mTLS Authentication.

Action

From operational mode, enter the `show services user-identification debug-counters | match MTLS` command to view the status of the mTLS authentication.

```
user@host> show services user-identification debug-counters | match MTLS
  MTLS Authentication Successful          :        2
  MTLS Authentication failed              :        0
  MTLS profile match failed, fallback password :    0
  MTLS auth processed by userfw           :        1
  MTLS auth processed by fwauthd          :        0
  MTLS auth processed by none             :        0
  MTLS failure due to NULL domain         :        0
  MTLS PTIM failed                        :        0
```

Meaning

The sample output confirms:

- You have successfully configured mTLS authentication.

- The user firewall has successfully processed mTLS authentication.

United Access Control (UAC)

SUMMARY

Learn about the use of SRX Series Firewall as an Infranet Enforcer in a Unified Access Control (UAC) network.

A Unified Access Control (UAC) 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.

Unified Access Control in Junos OS

An SRX Series Firewall 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 Firewall is called a Junos OS Enforcer.

Benefits

- Granularly and dynamically controls end user access based on user identity, device security state, and location information.

- Leverages your existing network infrastructure from user authentication to access points and switches, Juniper firewalls and IDP Series appliances through an open, standards-based architecture.

How does UAC work in Junos OS Environment?

Figure 17: Working of UAC in Junos OS Environment

Protected resources
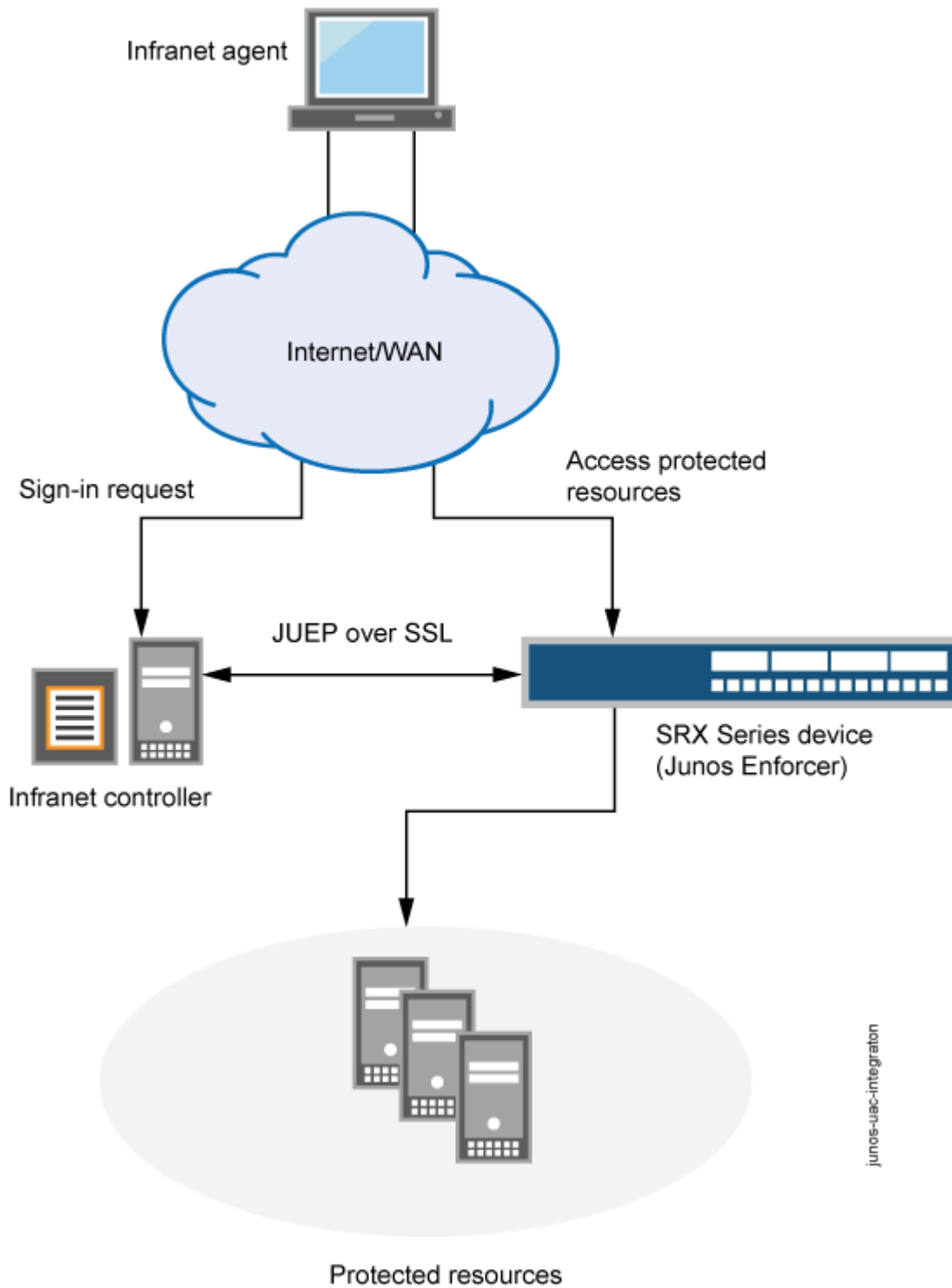
1. Set up the interfaces through which UAC traffic should enter the SRX Series Firewall.

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.

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 Firewall 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.

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
```

Junos OS Enforcer with IC Series UAC Appliance

What is Junos OS Enforcer with IC Series UAC appliance?

In a Unified Access Control (UAC) network, an SRX Series Firewall is called as Junos OS Enforcer when it is deployed in the UAC environment. The SRX Series Firewall verifies the certificate which IC Series UAC appliance submits. The SRX Series Firewall and IC Series UAC appliance perform mutual authentication. After authentication, the IC Series UAC appliance sends user and resource access policy information to the SRX Series Firewall to act as the Junos OS Enforcer.

How Junos OS Enforcer with IC Series UAC appliance works?

When you configure an SRX Series Firewall to connect to an IC Series UAC Appliance, the SRX Series Firewall and the IC Series UAC appliance establish secure communications as follows:

1. If more than one IC Series device are configured as Infranet Controllers on the SRX Series Firewall, 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 Firewall. If configured to do so, the SRX Series Firewall 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 Firewall 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 Firewall, the IC Series appliance sends its user authentication and resource access policy information. The SRX Series Firewall 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)*.

What is Junos OS Enforcer with 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.

The IC Series appliances configured on a Junos OS Enforcer should all be members of the same IC Series appliance cluster.

Junos OS Enforcer with IPsec

To configure an SRX Series Firewall 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 Firewall, 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`)

Policy Enforcement and Endpoint Security with Junos OS Enforcer

In a Unified Access Control (UAC) environment, after an SRX Series Firewall becomes Junos OS Enforcer, the SRX Series Firewall 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.

Enforce Policy with Junos OS Enforcer

Once the SRX Series Firewall 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.

   a. 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.

   b. 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.

   c. 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. 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.

b. 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.

c. 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).

Endpoint Security using Infranet Agent with 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.

2. 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.

3. The Infranet agent transmits the compliance information to the Junos OS Enforcer.

4. 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.

Captive Portal with 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. For more information, see "Example: Creating a Captive Portal Policy on Junos OS Enforcer" on page 191. 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 18: Captive Portal with Junos OS Enforcer
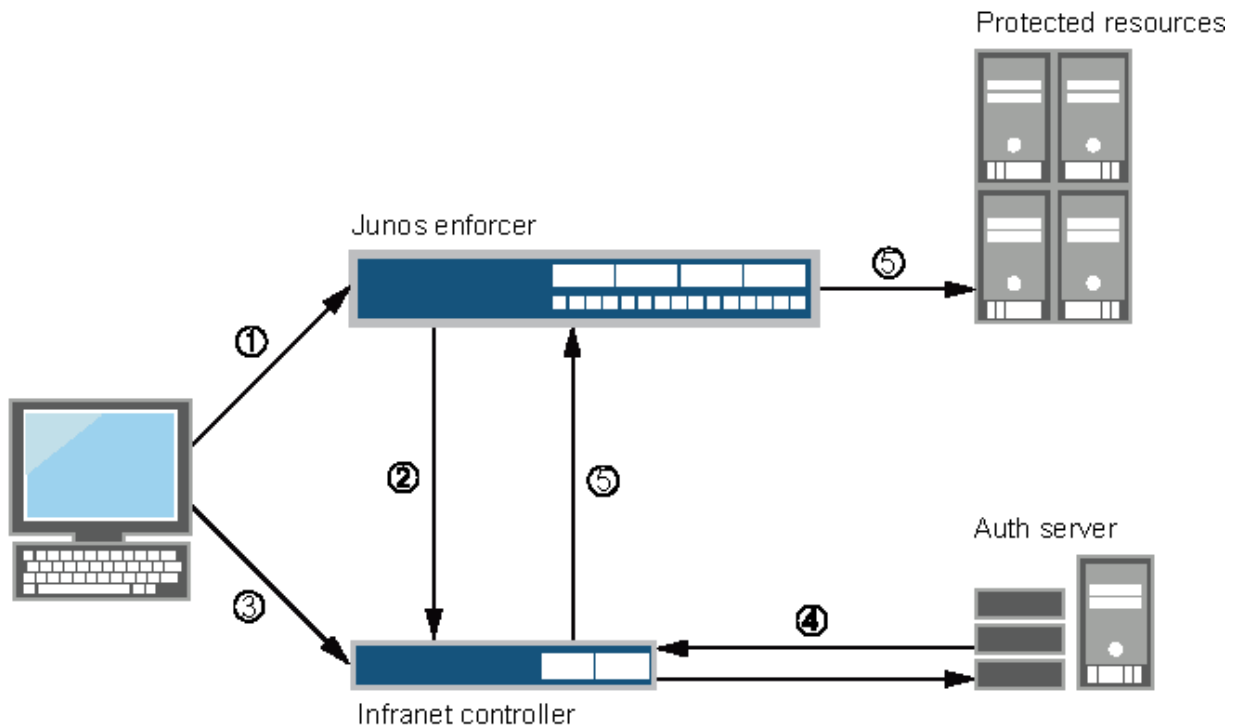
1. Users point to a protected resource using the browser.

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.

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.

Configure Unified Access Control (UAC)

SUMMARY

Learn how to configure an SRX Series Firewall to act as a Junos OS Enforcer in a UAC deployment.

Configure Junos OS Enforcer with IC Series UAC appliance

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See Enabling UAC in a Junos OS Environment (CLI Procedure).

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 Firewall. 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 Firewall 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 Firewall should connect.

To configure an SRX Series Firewall to act as a Junos OS Enforcer:

1. Specify the IC Series appliance(s) to which the SRX Series Firewall 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 Firewall should use to initiate secure communications with the IC Series appliance:

> **ℹ** **NOTE**: Any change in the Unified Access Control's (UAC) contact interval and timeout values in the SRX Series Firewall will be effective only after the next reconnection of the SRX Series Firewall 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 Firewall needs to verify the certificate.

   - To specify the server certificate subject that the SRX Series Firewall 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
   ```

> **ℹ** **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.

Configure Junos OS Enforcer with IPsec

To configure an SRX Series Firewall 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 Firewalls, 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;
    }
}
}
```

Configure Junos OS Enforcer Failover Options

Before you begin:

1. Enable UAC through the relevant Junos OS security policies.

2. Configure the SRX Series Firewall 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).

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.

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 Firewall will be effective only after the next reconnection of the SRX Series Firewall 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

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See Enabling UAC in a Junos OS Environment (CLI Procedure)

2. Configure the SRX Series Firewalls as a Junos OS Enforcer. See Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure).

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).

When configured in test-only mode, the SRX Series Firewall 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.

To activate or deactivate test-only mode, enter the following configuration statement:

```
user@host#  set services unified-access-control test-only-mode (true | false)
```

Verify Junos OS Enforcer Policy Enforcement

IN THIS SECTION

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

IN THIS SECTION

- Action | **190**

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.

Configure Endpoint Security Using Infranet Agent with 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.

Example: Creating a Captive Portal Policy on Junos OS Enforcer

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).

- 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`.

Topology

Configuration

IN THIS SECTION

- Procedure | **192**

Procedure

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

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 from an Active Directory Server

SUMMARY

Learn how to acquire username and role information from an Active Directory authentication server.

IN THIS SECTION

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 Firewall. 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 Firewall, 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 Firewall.

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 Firewall 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 Firewall 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 Firewall 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 Firewall.

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 Firewall 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 Firewall 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 Firewall 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 Firewall, the MAG Series device, the browser, and the authentication server.

Topology

Figure 19 on page 198 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 19: Single Sign-On Support Topology
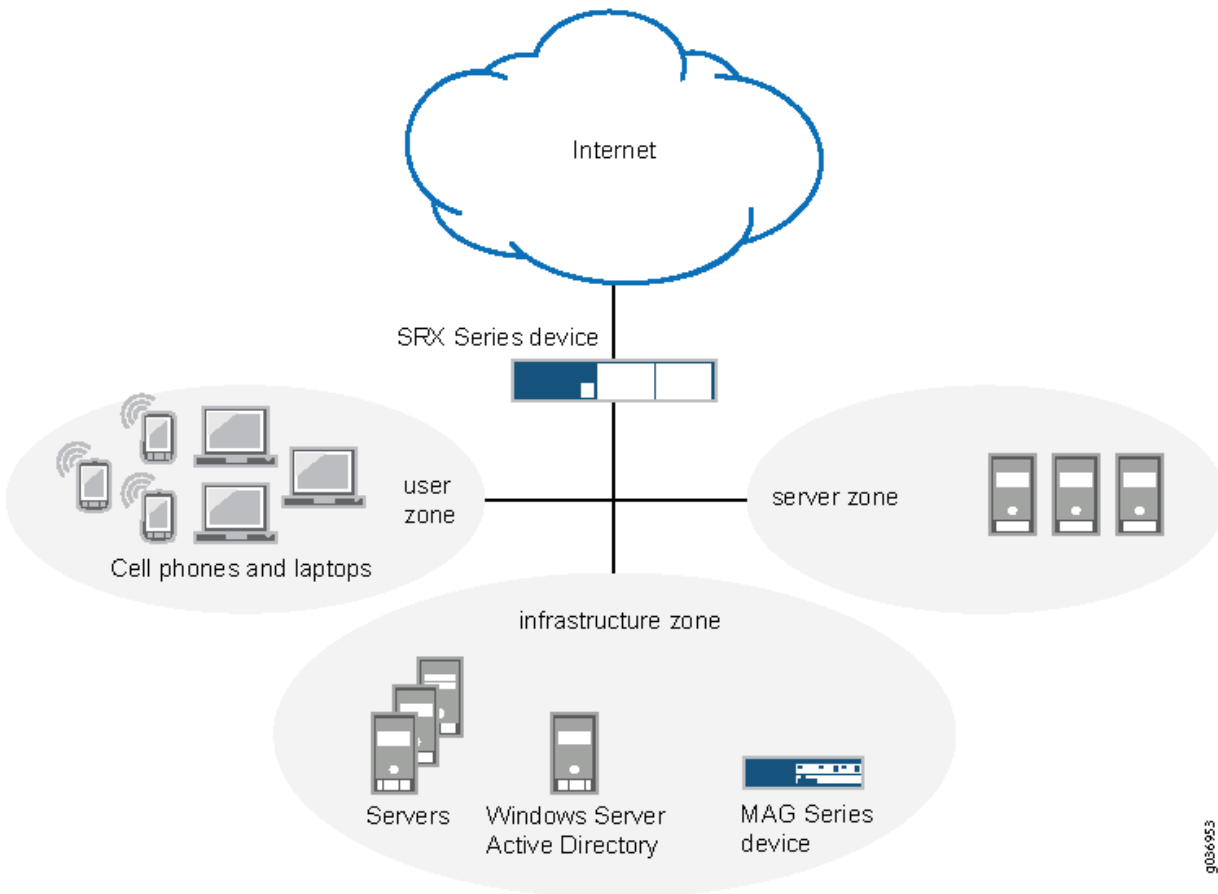
A user's request to access another resource is controlled by roles and groups associated with the user. For example, a user belonging to a group of developers named Dev might have access to a particular test server. The same user might also be the manager and belong to the Mgr group that can access certain 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 Firewall 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 Firewall.

Configuration

Configure the devices for this solution by performing the following tasks.

- Connect the SRX Series Firewall 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 Firewall 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 Firewall.

- Configure endpoint browsers for the SPNEGO protocol

> **(i)** **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 Firewall.

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 Firewall 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 Firewall communicate over a secure channel. When the SRX Series Firewall 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 Firewall and the Access Control Service.

To connect the SRX Series Firewall and the Access Control Service on the MAG Series device:

1. Configure the SRX Series Firewall.

   Step-by-Step Procedure

   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 Firewall.

```
[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
Firewall 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 Firewall, enter commit from configuration mode.

2. Configure the Access Control Service from the administrator console on the MAG Series device.

Step-by-Step Procedure

a. Navigate to the Infranet Enforcer page, and click **New Enforcer**.

b. Select Junos, enter the password set previously on the SRX Series Firewall (InSub321), and enter
the serial number of the SRX Series Firewall.

c. Click **Save Changes**.

Results

When both devices are configured, the SRX Series Firewall 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 Firewall. 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 Firewall, 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 Firewall. 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 Firewall.

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 Firewall without affecting other network operations. A later task modifies this configuration to provide user role retrieval from the remote Active Directory server.

> (i)   **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.

   Step-by-Step Procedure

   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.

    Step-by-Step Procedure

    a. Select **Authentication>Auth. Servers**.

    b. Select **System Local**. This establishes the MAG Series device as the default authentication server.

3. Create users.

    Step-by-Step Procedure

    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.

    Step-by-Step Procedure

    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.

Step-by-Step Procedure

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.

Step-by-Step Procedure

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.

Step-by-Step Procedure

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 Firewall:

Step-by-Step Procedure

a. Confirm that the authentication table on the SRX Series Firewall 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 Firewall 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 Firewall 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.

> (i) **NOTE**: Other services, such as IDP, Content Security, 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 Firewall to the Access Control Service:

1. From configuration mode on the SRX Series Firewall, 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 Firewall 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
```

> **(i) 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

Step-by-Step Procedure

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.

Step-by-Step Procedure

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.

   i **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 Firewall, verify that the authentication data and roles from the Access Control Service were pushed to the SRX Series Firewall 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:

   Step-by-Step Procedure

   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 Firewall 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.

    Step-by-Step Procedure

    a. Name the Active Directory server.

    b. Enter its NetBIOS domain name in the domain box.

    > (i) **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.

Step-by-Step Procedure

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.

Step-by-Step Procedure

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.

1. 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.

2. 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**.

   > **(i)** **NOTE**: You need to specify the URL of the resource (in this solution, the FQDN or domain controller value UCDC.com).

3. 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.

SEE ALSO

Authentication and Integrated User Firewalls User Guide

Classify Traffic Based on User Roles through SRX Firewall Users

SUMMARY

Learn how to obtain username and role information through SRX firewall users.

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.

Aruba ClearPass

SUMMARY

Learn about how the SRX Series Firewall and NFX Series devices communicate with Aruba ClearPass. You can learn about the Web API and user query function.

IN THIS SECTION

The SRX Series Firewall and NFX Series devices associate with Aruba 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.

Communication Between ClearPass and Firewall

The SRX Series Firewall and the ClearPass Policy Manager (CPPM) communicate with each other to authenticate users and provide access to the Internet and internal, protected resources.

Benefits

- Quick and easy access to data that help to manage and maintain your networks, clients, and devices.

- Continuous monitoring and advance analytics that offer real-time visibility into your network.

How communication happens between the firewall and the ClearPass?

Figure 20: ClearPass and SRX Series Firewall Communication
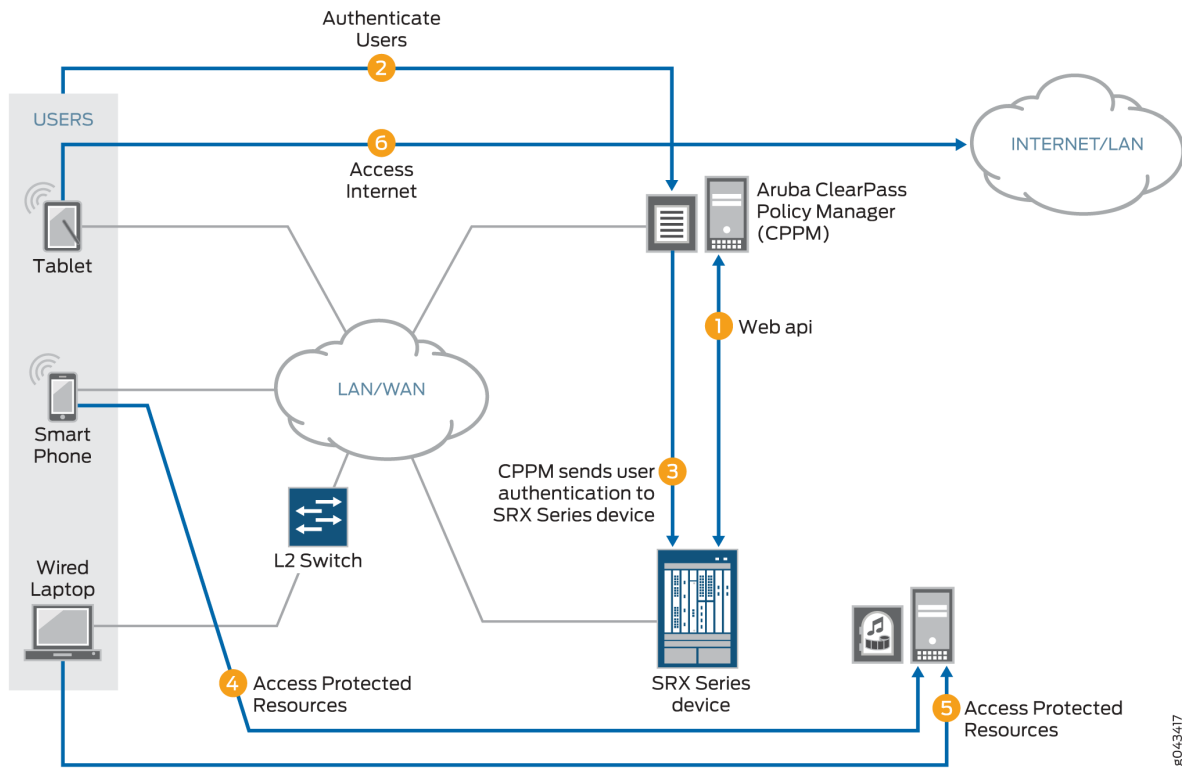
1. The ClearPass Policy Manager (CPPM) initiates a secure connection with the SRX Series Firewall using the Web API.

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 firewall in POST request messages using the Web API.

   When traffic from a user arrives at the firewall, the firewall:

   • 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 firewall. Because all of the conditions identified in Step 3 are met and the security policy permits it, the firewall 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 firewall. Because all of the conditions identified in Step 3 are met and the security policy permits it, the firewall allows the user connection to the resource.

6. Traffic from the tablet user who is requesting access to the Internet arrives at the firewall. Because all of the conditions identified in Step 3 are met and the security policy permits it, the firewall allows the user connection to the Internet.

7. 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.

8. 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 2.

Figure 21: User Information from the CPPM to the SRX Series Firewall Routing Engine Synchronized to the ClearPass Authentication Table
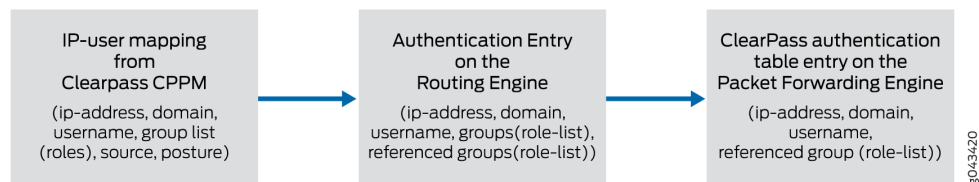
The firewall uses the authenticated user identity information in the following process. When a user attempts to access an internal, protected resource or the Internet, the device:

- 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 firewall 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 firewall 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.

  The firewall 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.

Enforce Security with Aruba ClearPass

The SRX Series Firewall, or NFX Series devices collaborate with Aruba ClearPass 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.

Why you need Aruba ClearPass with Enforcement Security?

1. Risk and Challenges—Use of company smartphones poses one of the biggest IT security risks to businesses.

   - 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.

   - 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.

   - Attackers can launch information-gathering ventures, stop business activity, and steal sensitive corporate data.

2. Need for Business—The proliferation of mobile devices and cloud services and securing them has become a fundamental strategic part of enterprise cybersecurity.

   - In a work environment that supports mobile devices, knowing the identity of the user is important.

   - An identity of the user provides IT administrators with improved advantage in identifying the source of the attack and stemming future potential attacks that follow the same strategy.

   - The ClearPass with enforcement security protects against malicious intrusions introduced through use of mobile devices and multiple concurrently connected devices.

3. Protection using ClearPass—The ClearPass with enforcement security 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.

    -   ClearPass 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 and not only by their devices.

    -   You can easily control their network access and any egregious activity on their behalf, whether intended or not.

How Aruba ClearPass with Enforcement Security works?

The Aruba ClearPass with enforcement security delivers the protection of the SCREENS, IDP and Content Security 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.

The ClearPass with enforcement security gives you granular control at the user level:

-   As administrator of the device, you can now specify in the identity source 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 Firewall 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 Firewall. 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.

SRX Series Firewall 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 firewall 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.

Web API Function

The firewall 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.

If you are deploying the 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.

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 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
```

The Web API daemon runs on the primary Routing Engine in a chassis cluster environment. After an Chassis Cluster switchover, the daemon will start automatically on the new primary Routing Engine. It has no effect on the Packet Forwarding Engine. Web API supports both the IPv4 and IPv6 address user entries obtained from CPPM.

Integrity of Data Sent from ClearPass to the Firewall

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 Firewall can process a maximum of 209 roles.

  - The SRX Series Firewall 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.

- The CPPM checks the health and posture of a firewall and it can send that information to the firewall as part of the user information that it posts.

- You cannot define posture on the firewall. Also, firewall 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 Firewall 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)

User Query Function

You can obtain user authentication and identity information for an individual user when that information is not posted directly to the firewall by the ClearPass Policy Manager (CPPM).

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 firewall, the firewall 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 firewall does not search its other authentication tables. Rather, it sends a query to the CPPM requesting authentication information for the user. In this example:

Figure 22: The ClearPass Integration User Query Function
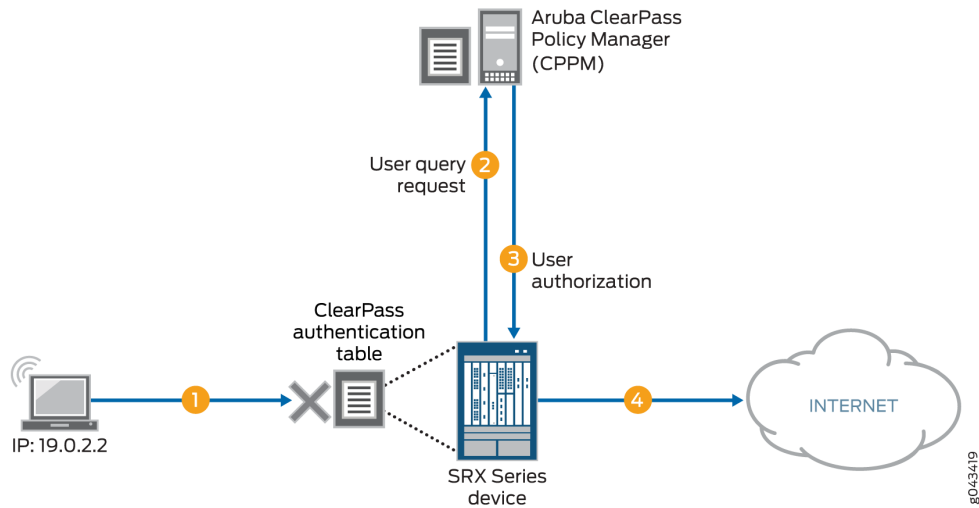
1. A user attempts to access a resource. The firewall receives the traffic requesting access. The SRX Series Firewall searches for an entry for the user in its ClearPass authentication table, but none is found.

2. The firewall 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 firewall creates an entry for the user in its ClearPass authentication table, and grants the user access to the Internet.

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. The `delay-query-time` parameter influences the query process:

    1. 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.

    2. 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 firewall can enter the user identity information in its ClearPass authentication table.

3. 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.

4. After the delay timeout expires, the firewall 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.

5. If there are many query requests in the queue, the firewall 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 firewall 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 15: 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 16: 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

Filter and Rate-limit Threat and Attack Logs

The SRX Series Firewall transmits the threat and attack logs recorded to the ClearPass Policy Manager (CPPM). CPPM can use the log data to harden the security. You can also configure the threats and attacks related to a specific device and their users. For more information, see "Example: Configure ClearPass to Filter and Rate-limit Threat and Attack Logs" on page 292.

How ClearPass Detect Threats and Attacks and Notifies the CPPM

When the firewall detect threat and attack events, the event is recorded in the firewall event log. The firewall 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 firewall and define appropriate actions on the CPPM to harden your security.

Junos OS on the SRX Series Firewall generates over 100 different types of log entries issued by more than 10 of its modules. Among the firewall that generate threat and attack logs are SCREENS, IDP, and Content Security. To avoid overburdening the firewall and the log server, the ClearPass allows you to configure the firewall 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 Content 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 firewall log transmission follows the syslog protocol, which has a message format that allows vendor-specific extensions to be provided in a structured way.

Table 17: 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

Table 17: Attack Log Fields Using Example Log *(Continued)*

Log Entry Component	Meaning	Format	Example
Time and Time Zone	When the log was recorded and in what time zone.	*y-m-dThs.ms+time zone* • y = year • m=month • d = day • 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. The process ID is not meaningful in this context, so *pid* is replaced by "-". The value "-" is a placeholder for process ID.	*pid*	-
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.	[ ]	-

Table 17: Attack Log Fields Using Example Log *(Continued)*

Log Entry Component	Meaning	Format	Example
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

Threat and Attack Logs Sent to Aruba ClearPass

The firewall 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 Content Security components clearly identify the types of attacks and threats that threaten a company's network security.

The firewall 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 Firewall transmits the logs in volumes determined by the rate-limiting conditions that you set.

Table 18: 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	
RT_SCREEN_TCP	TCP attack

Table 18: Threat and Attack Log Entries Generated by SRX Series Components *(Continued)*

Log Type	Description
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
AV_VIRUS_DETECTED_MT_LS	A virus was detected by the antivirus scanner.
ANTISPAM_SPAM_DETECTED_MT	spam
ANTISPAM_SPAM_DETECTED_MT_LS	The identified e-mail was detected to be spam.
IDP_APPDDOS_APP_ATTACK_EVENT	Application-level distributed denial of service (AppDDoS) attack
IDP_APPDDOS_APP_ATTACK_EVENT_LS	The AppDDoS attack occurred when the number of client transactions exceeded the user-configured connection, context, and time binding thresholds.
IDP_APPDDOS_APP_STATE_EVENT	AppDDoS attack

The AppDDoS state transition occurred when the number of application transactions exceeded the user-configured connection or context thresholds. |

Table 18: Threat and Attack Log Entries Generated by SRX Series Components *(Continued)*

Log Type	Description
IDP_APPDDOS_APP_STATE_EVENT_LS	
IDP_ATTACK_LOG_EVENT	Attack discovered by IDP
IDP_ATTACK_LOG_EVENT_LS	IDP generated a log entry for an attack.

ClearPass with JIMS

The firewall relies on Juniper Identity Management Service (JIMS) and ClearPass for user identity information. You can configure ClearPass and Juniper Identity Management Service (JIMS) at the same time. When you configure ClearPass and JIMS at the same time, the firewall can query JIMS for user identification entries, and ClearPass can push these entries to the devices through the Web API. For more information, see .

How ClearPass works with JIMS?

When a user gets authenticated by CPPM, the CPPM uses a Web API to push user or device information to a firewall. The firewall builds up the authentication entry or device information for the user, and the user traffic can pass-through the firewall based on security policy. When windows Active Directory client log on to domain, firewall 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, firewall 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 firewall receives a response from CPPM without IP-user mapping, then the firewall queries JIMS to obtain IP-user or group mapping.

When the IP-user or group mapping is received from both JIMS and CPPM, firewall 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.

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.

Different Scenarios of how ClearPass works with JIMS

A detailed explanation with scenarios of how ClearPass with JIMS works is as follows:

Scenario 1: What Firewall Does If CPPM Responds with IP-User or Group Mapping Information

Figure 4 shows when an SRX Series Firewall 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 Firewall 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 Firewall queries ClearPass for user identity.

3. The ClearPass sends the IP-user or group mapping information to the SRX Series Firewall.

4. The SRX Series Firewall adds the information to the authentication table.

Figure 23: What Firewall Does If CPPM Responds with IP-User or Group Mapping Information
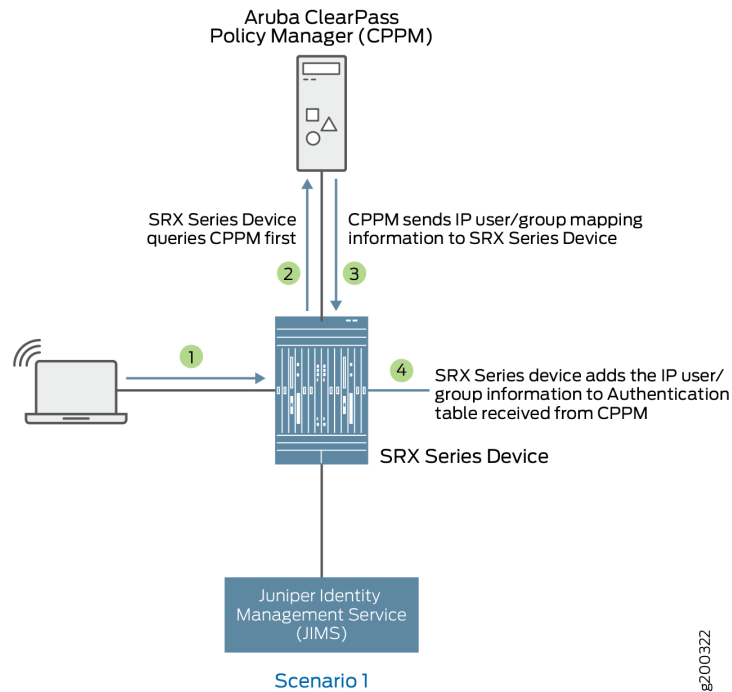
Scenario 2: What Firewall Does If CPPM Does Not Respond or CPPM Responds with No IP-User or Group Mapping Information

Figure 5 shows when an SRX Series Firewall 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 firewall 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 firewall queries ClearPass for user identity.

3. If the firewall does not receive a response from ClearPass, the firewall queries JIMS.

4. The JIMS sends IP-user or group mapping information to the firewall.

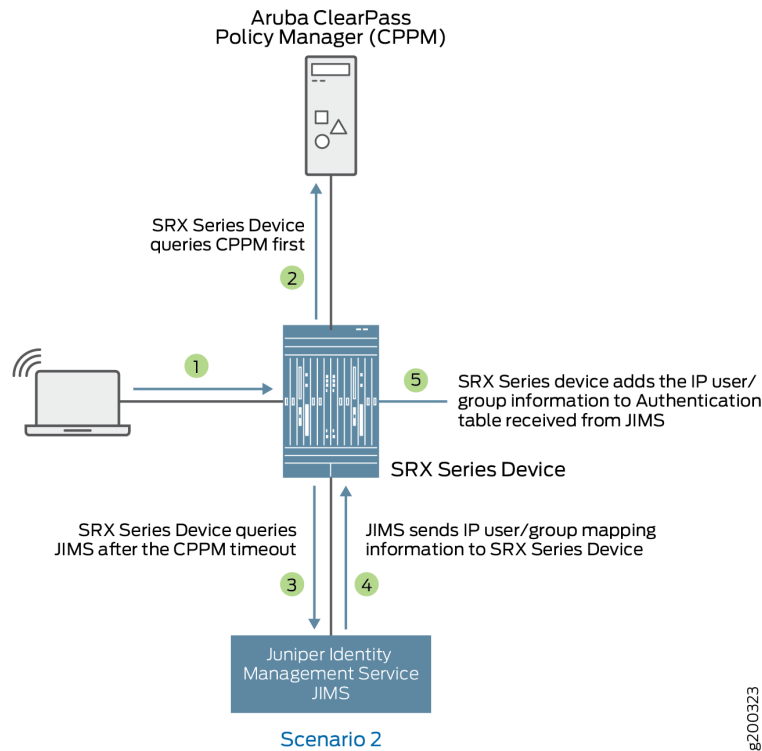5. The firewall adds the information received from JIMS to the authentication table.

Figure 24: What Firewall Does If CPPM Does Not Respond or CPPM Responds with No IP-User or Group Mapping Information

Domain and Interested Groups

How the user identity group information is managed on the device is dominated by two concepts: Domain group and Interested group.

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 19: Assigning a Domain to a Group

Does the IP-User Mapping Contain a Domain Name?	What Domain Is Applied to the Group?
No For example: `IP, , user1, group-list` 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 For example: `IP, domain1, user1, group-list` 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 Firewall:

```
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 20 on page 239 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 20: 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: The SRX Series Firewall gets the IP-user mapping for a user from the CPPM. None of the groups in the user mapping are referenced by security policies.	
IP-user mapping from the CPPM: `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. `203.0.113.9 , ,user1`
Case 2: The SRX Series Firewall 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: `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: `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.

ClearPass Authentication Table

The SRX Series Firewall receives information from the CPPM. The firewall extracts the user authentication and identity information, and analyzes it. The firewall creates a ClearPass authentication table on the Packet Forwarding Engine side to hold this user information. When the firewall receives the information from ClearPass, the firewall generates entries in the ClearPass authentication table for the authenticated users. When the firewall 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.

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 firewall to check the ClearPass authentication table first if there are other authentication tables on the Packet Forwarding Engine.

How the Firewall manages the ClearPass Authentication Table?

The firewall 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. ClearPass acts as the authentication source for the firewall. The CPPM sends to the firewall identity information about users that it has authenticated. The UserID daemon process in the firewall

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.

User Authentication Entries in the ClearPass Authentication Table

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 firewall is tied to the username and its groups in the ClearPass authentication table entry. 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. The 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.

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 num record-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 firewall 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 firewall 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
```

If more than 2048 sessions are associated with a single authentication entry in the ClearPass authentication table, the active directory 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.

ClearPass Timeout Setting

What is timeout setting for Aruba ClearPass?

Authentication entries in the Aruba ClearPass authentication table contain a timeout value after which the entry expires. 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.

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 configure a value for the invalid timeout setting, that timeout is applied to the entry. If you don't configure the invalid entry timeout, then its default timeout of 30 minutes is applied to the new entry. The invalid entry timeout is also applied to entries whose state is changed from valid or pending to INVALID.

How timeout setting for Aruba ClearPass works?

Use the following command to configure the invalid authentication entry timeout for entries in the ClearPass authentication table. Here, 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 21: Invalid Authentication Timeout 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

Example: Enforce SRX Series Security Policy Using Aruba ClearPass as the Authentication Source

Configure Aruba ClearPass

SUMMARY

Learn how to configure the SRX Series Firewall to include security policies with Aruba ClearPass.

IN THIS SECTION

- Example: Enforce Security Policy with Aruba ClearPass | **246**
- Example: Configure Web API Function | **269**
- Example: Configure User Query Function | **281**
- Example: Configure ClearPass to Filter and Rate-limit Threat and Attack Logs | **292**

Example: Enforce Security Policy with Aruba ClearPass

This example covers how to configure security to protect your resources and control access to the internet using the SRX Series Firewall 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 Firewall, see the following topics:

- "Web API Function" on page 223

- "User Query Function" on page 225

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.

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 Firewall 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 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.

  It is assumed that the CPPM is configured to provide the SRX Series Firewall 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 Firewall 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 Firewall 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 Firewall 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 Firewall. 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 Firewall 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 Firewall can search its other authentication tables, if you have configured a search order that includes them. See Table 1 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.

You configure role mappings on the CPPM, not on the SRX Series Firewall.

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 Firewall 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 Firewall 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 Firewall. 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 Firewall 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 Firewall.

- The SRX Series Firewall 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 Firewall 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.

summarizes the users, their groups, and the zones to which they belong. All users belong to the default GLOBAL domain.

Table 22: Authenticated User Information for Security Policy Example

User	Group	Zone
Abe (abew1)	• marketing-access-limited-grp	marketing-zone

Table 22: Authenticated User Information for Security Policy Example *(Continued)*

User	Group	Zone
John (jxchan)	- posture-healthy - marketing-access-for-pcs-limited-group - marketing-general - sales-limited - corporate-limited	marketing-zone
Lin (lchen1)	- posture-healthy - human-resources-grp - accounting-limited - corporate-limited	human-resources-zone
Viki (viki2)	- posture-healthy - accounting-grp - accounting-grp-and-company-device - corporate-limited	accounting-zone
guest1	- posture-healthy - guest	public-zone
guest2	- posture-healthy - guest-device-byod	public-zone

Topology

shows the topology for this example.

Figure 25: Topology for the Integrated ClearPass Authentication Enforcement Through Security Policies Example
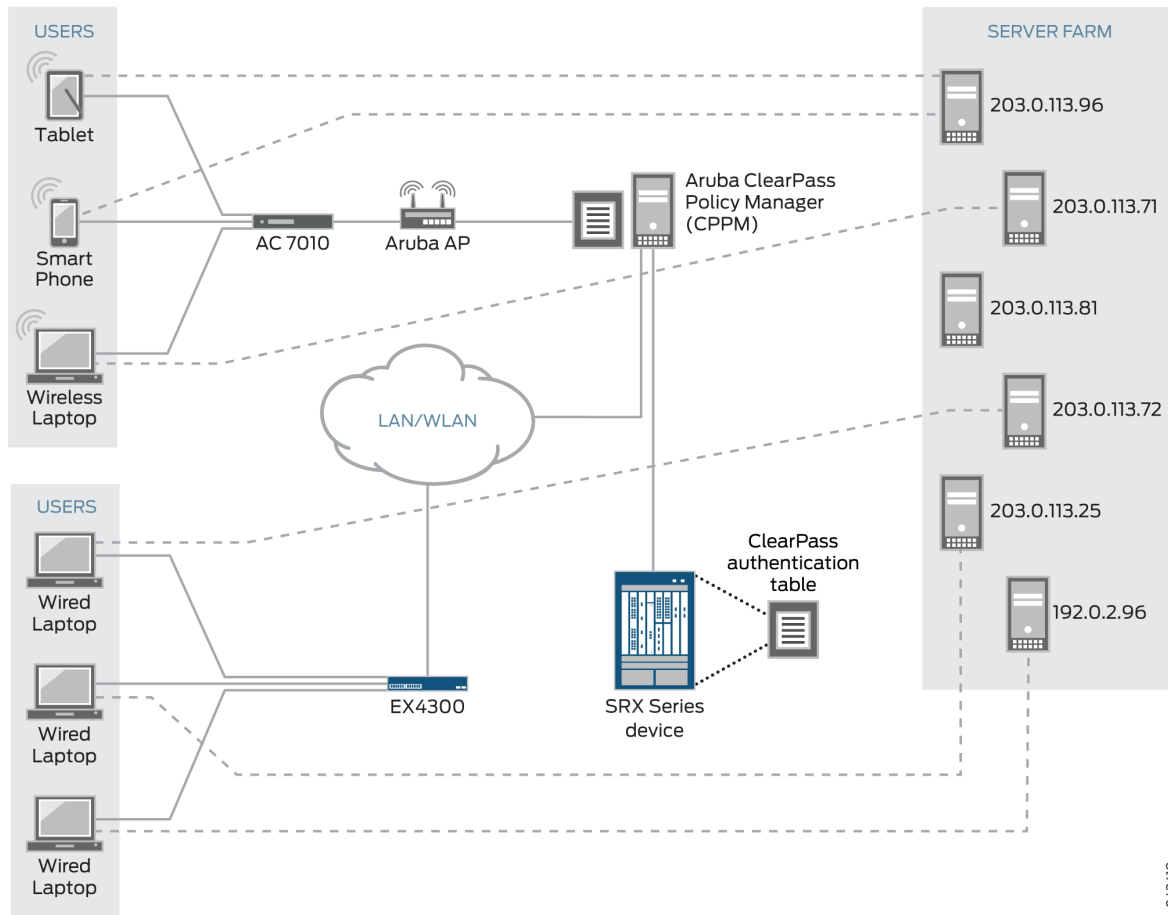
Configuration

IN THIS SECTION

This section covers how to configure the SRX Series Firewall 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 Firewall:

```
[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

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
```

Example: Configure Web API Function

IN THIS SECTION

- Requirements | **269**
- Overview | **270**
- Configuration | **274**

The SRX Series Firewall 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 Firewall must authenticate users in conjunction with applying security policies that match their requests. For the integrated ClearPass authentication and enforcement feature, the SRX Series Firewall 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 Firewall. The CPPM uses this connection to post user authentication information to the SRX Series Firewall. In their relationship, the SRX Series Firewall 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 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 Firewall 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 Firewall 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

IN THIS SECTION

You can configure identity-aware security policies on the SRX Series Firewall 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 Firewall relies on the CPPM for user authentication. The SRX Series Firewall 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 Firewall 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 Firewall to establish a connection between the SRX Series Firewall and the CPPM.

Figure 26 on page 271 illustrates the communication cycle between the SRX Series Firewall and the CPPM, including user authentication.

Figure 26: ClearPass and SRX Series Firewall Communication and User Authentication Process
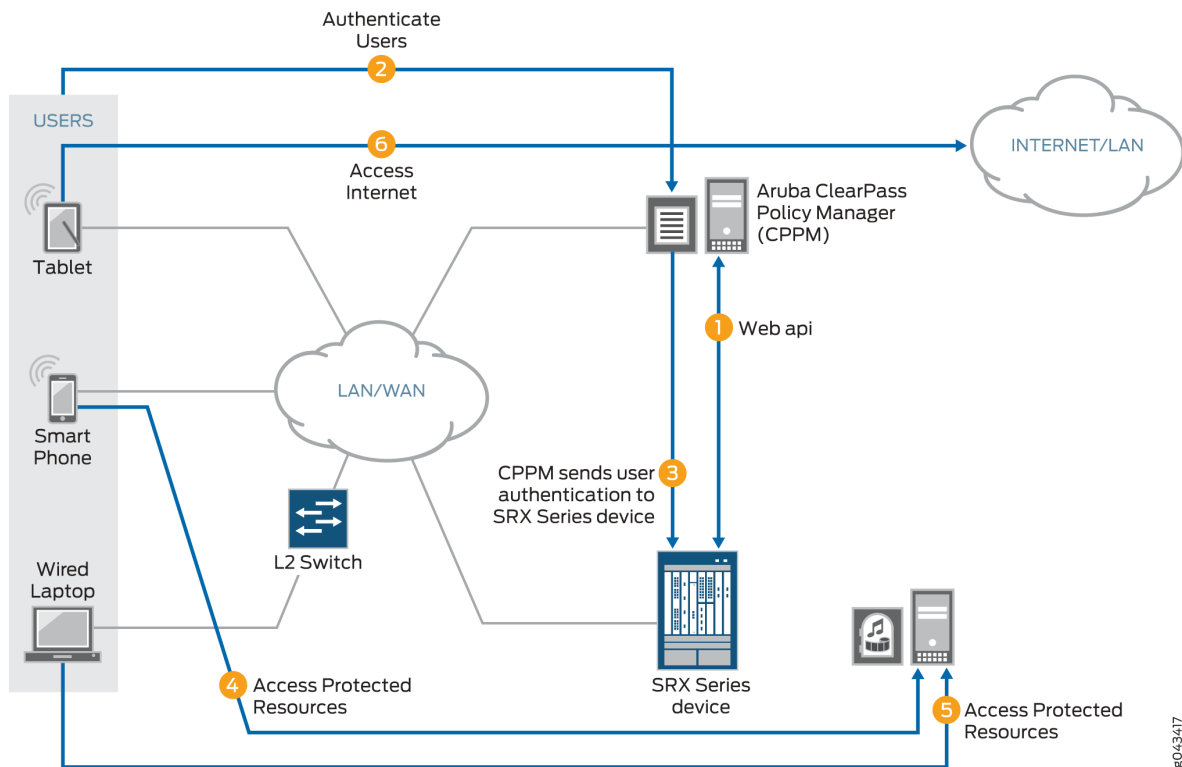
As depicted, the following activity takes place:

1. The CPPM initiates a secure connection with the SRX Series Firewall 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 Firewall in POST request messages using the Web API.

When traffic from a user arrives at the SRX Series Firewall, the SRX Series Firewall:

- 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 Firewall. Because all of the conditions identified in Step 3 are met and the security policy permits it, the SRX Series Firewall 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 Firewall. Because all of the conditions identified in Step 3 are met and the security policy permits it, the SRX Series Firewall 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 Firewall. Because all of the conditions identified in Step 3 are met and the security policy permits it, the SRX Series Firewall 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.

> **(i)** **NOTE**: Note that if you deploy Web management and the SRX Series Firewall together, they must run on different HTTP or HTTPS service ports.

See No Link Title 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 Firewall 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 Firewall 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 Firewall 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 Firewall. 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

shows the topology used for the integrated ClearPass deployment examples.

Figure 27: Integrated ClearPass Authentication and Enforcement Deployment Topology
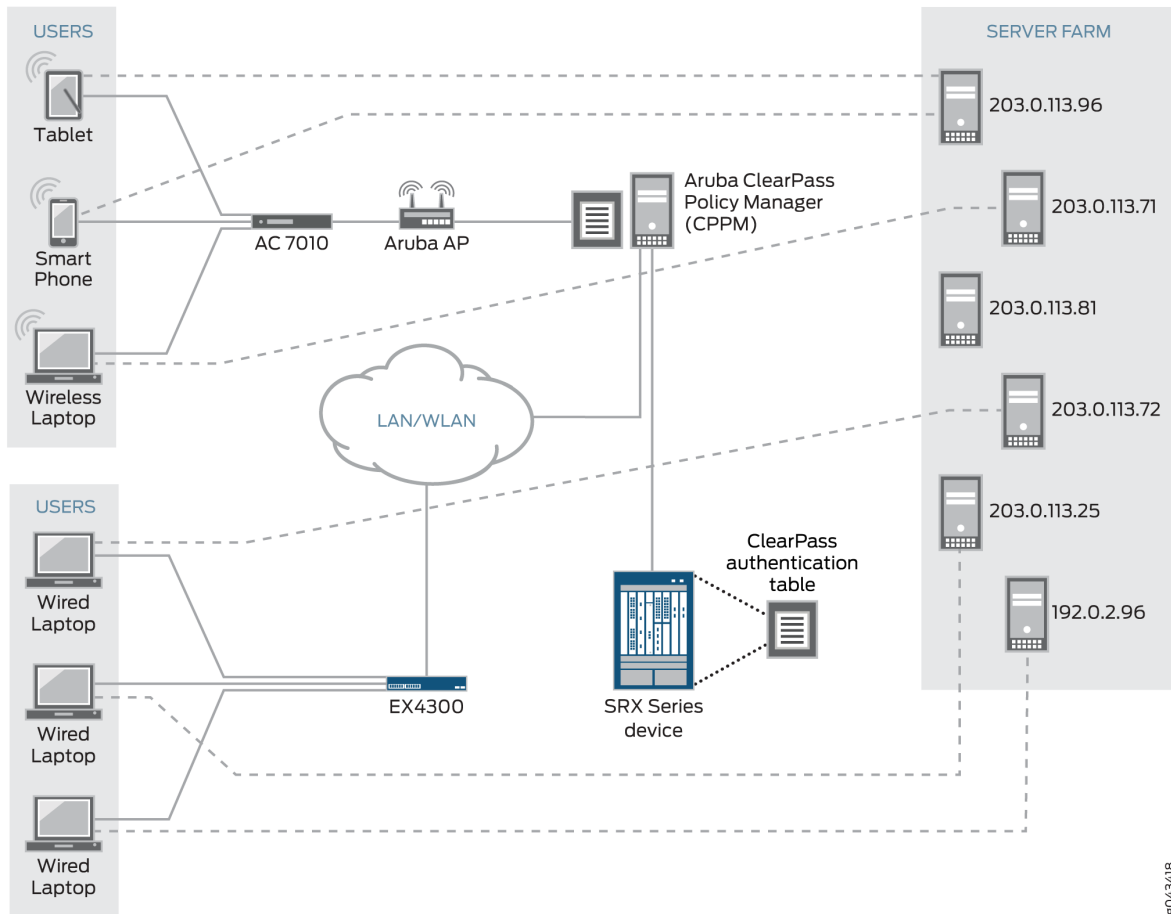
Configuration

> **IN THIS SECTION**
>
> - CLI Quick Configuration | **275**
> - Configuring the SRX Series Web API Daemon | **275**
> - Configuring the ClearPass Authentication Table Entry Timeout and Priority | **278**

This section covers how to enable and configure the SRX Series Web API.

> **ℹ** **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 Firewall. No separate connection configuration is required.

It is assumed that the CPPM is configured to provide the SRX Series Firewall 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 Firewall, the SRX Series Firewall treats the role mappings as groups. The SRX Series Firewall does not distinguish them from other groups.

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 Firewall 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 Firewall 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 Firewall 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 Firewall 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 Firewall 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 Firewall 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 Firewall to check the ClearPass authentication table first if there are other authentication tables on the Packet Forwarding Engine. shows the new authentication table search priority.

Table 23: SRX Series Firewall 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

Understanding Enforcement of ClearPass User and Group Authentication

Understanding the Integrated ClearPass Authentication and Enforcement User Query Function

Example: Configure User Query Function

This example covers how to configure the SRX Series Firewall to enable it to query Aruba ClearPass automatically for user authentication and identity information for an individual user when that information is not available.

> (*i*) **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 29 on page 285 for the topology. For details on the user query process, see Figure 28 on page 283.

The hardware and software components are:

- Aruba ClearPass (CPPM). The CPPM is configured to use its local authentication source to authenticate users.

> (*i*) **NOTE**: It is assumed that the CPPM is configured to provide the SRX Series Firewall 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 Firewall 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

IN THIS SECTION

-

You can configure the user query function to enable the SRX Series Firewall 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 Firewall 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 Firewall 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 Firewall 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.

shows the user query flow process, which encompasses the following steps:

1. A user attempts to access a resource. The SRX Series Firewall 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 28: User Query Function Process
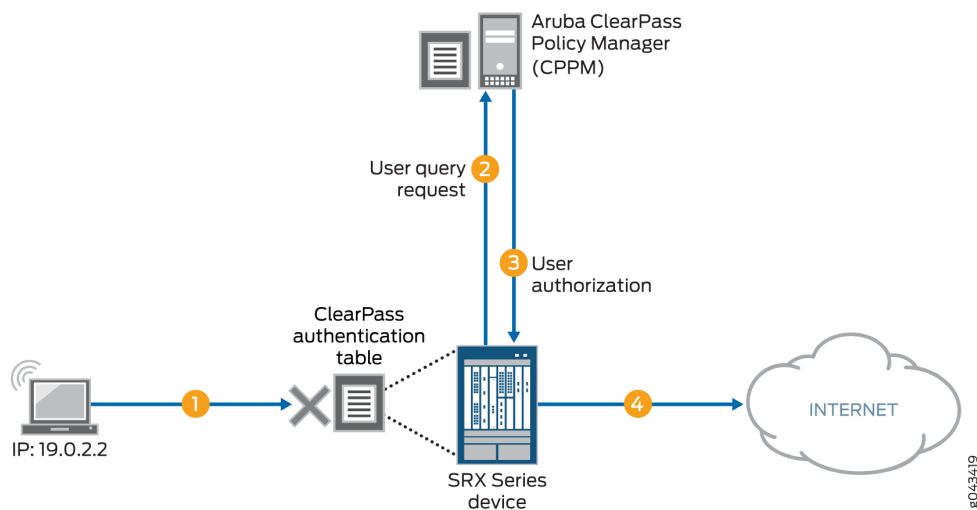
For details on the parameters that you can use to control when the device issues the query, see Understanding the Integrated ClearPass Authentication and Enforcement User Query Function.

> **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:
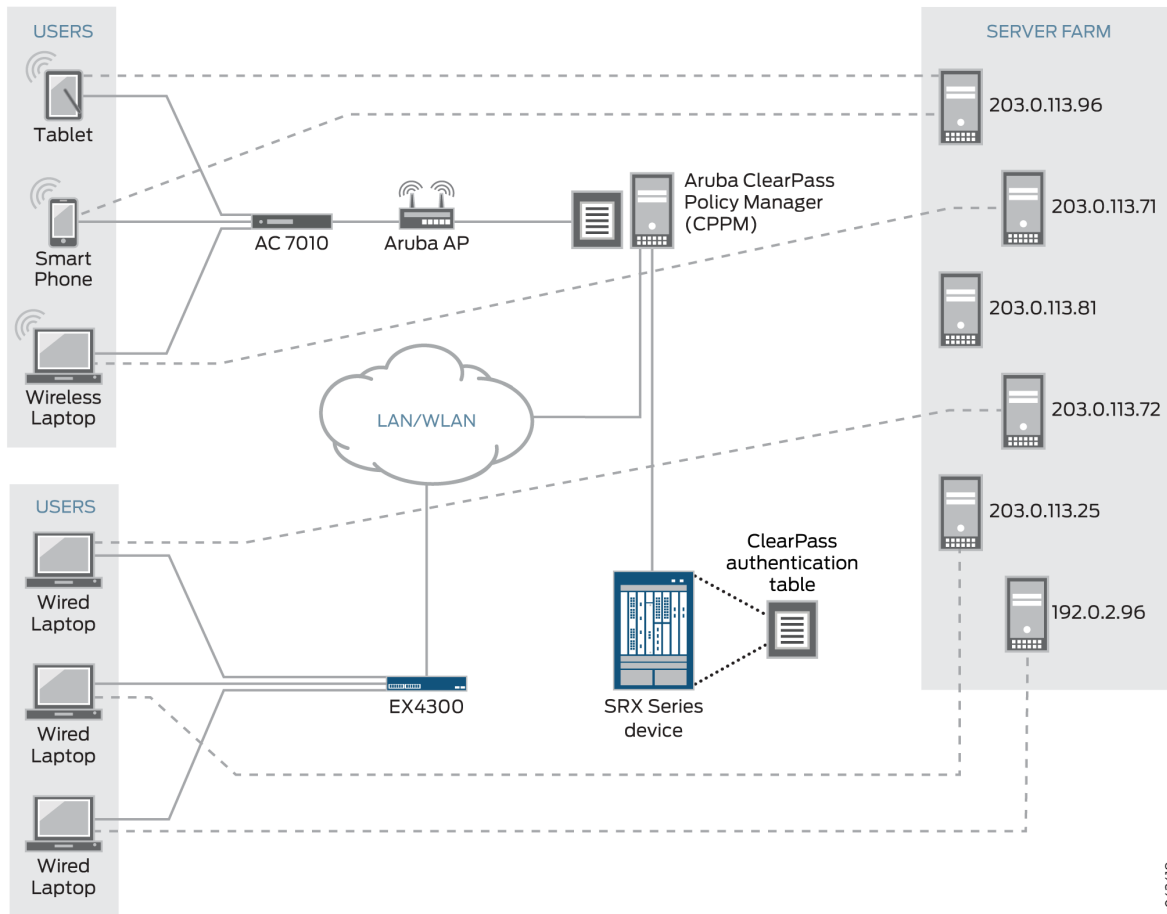
```
{
"access_token":"ae79d980adf83ecb8e0eaca6516a50a784e81a4e",
"expires_in":2880,
"token_type":"Bearer",
"scope"=nu;
}
```

Before the access token expires, the device can obtain a new token using the same message.

Topology

shows the overall topology for this deployment, which encompasses the user query environment.

Figure 29: Topology for the Overall Deployment that Includes User Query

Configuration

IN THIS SECTION

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 Firewall 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 Firewall is the HTTP client. By it sends HTTPS requests to the CPPM on port 443.

To enable the SRX Series Firewall 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 Firewall 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 Firewall. In this example, the following path is used:

```
/var/tmp/RADUISServerCertificate.crt
```

3. Configure the client ID and the secret that the SRX Series Firewall 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

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.

> *(i)* **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
```

Example: Configure ClearPass to Filter and Rate-limit Threat and Attack Logs

The SRX Series Firewall 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 Firewall 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 Firewall running Junos OS that includes the integrated ClearPass feature. The SRX Series Firewall is connected to the Juniper Networks EX4300 switch and to the Internet. The SRX Series Firewall 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 Firewall 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 Firewall 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 Firewall 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 Firewall to send only threat and attack logs to the CPPM. This filter allows you to ensure that the SRX Series Firewall 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.

> (i) **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

shows the topology for this example.

Figure 30: Integrated ClearPass Authentication and Enforcement Deployment Topology
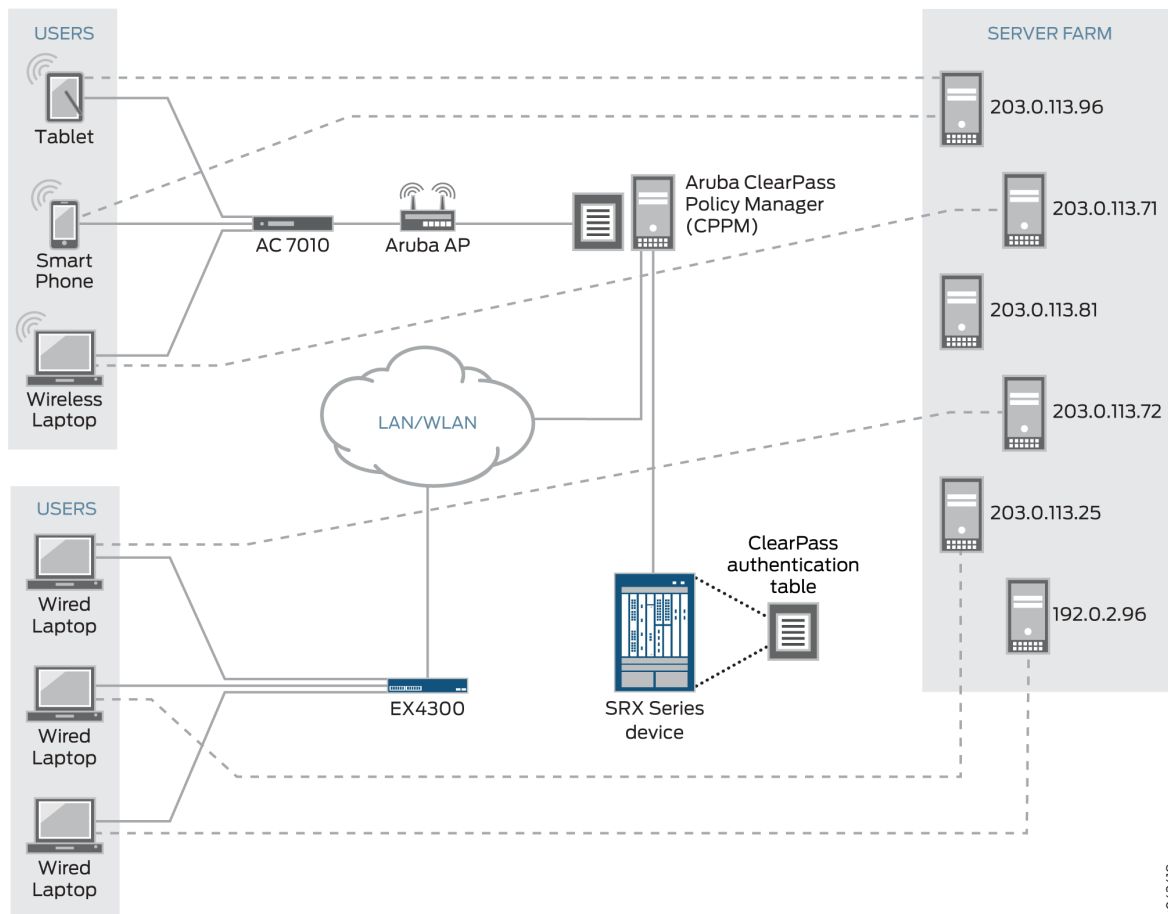
Configuration

IN THIS SECTION

- CLI Quick Configuration | **295**
- Configuring Integrated ClearPass Authentication and Enforcement to Filter for Threat and Attack Logs Sent to the CPPM | **295**
- Results | **296**

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;
    }
  }
}
```

Example: Configure ClearPass with JIMS

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 Firewall.

- 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.

  > *(i)* **NOTE**: It is assumed that the CPPM is configured to provide the SRX Series Firewall 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 Firewall obtains the user or device identity information from different authentication sources. After the SRX Series Firewall obtains the device identity information, it creates an entry in the device identity authentication table. The SRX Series Firewall relies on JIMS and ClearPass for user identity information. By enabling JIMS and ClearPass at the same time, an SRX Series Firewall 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 Firewall through Web API.

When both JIMS IP query and ClearPass user query are enabled, SRX Series Firewall always queries ClearPass first. When the IP-user or group mapping is received from both JIMS and CPPM, an SRX Series Firewall 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 Firewall 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
```

Procedure

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 Firewall 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 Firewall 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 Firewall 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 Firewall 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 | **304**

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.

3

CHAPTER

Configuration Statements and Operational Commands

Junos CLI Reference Overview | 309

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements

- Operational Commands