

Packet Mirroring in an SRC-Managed Network



Modified: 2018-10-08

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. and/or its affiliates in the United States and other countries. All other trademarks may be 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.

Packet Mirroring in an SRC-Managed Network
Copyright © 2018 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	vii
	Documentation and Release Notes	vii
	Documentation Conventions	vii
	Documentation Conventions	viii
	Documentation Feedback	x
	Requesting Technical Support	xi
	Self-Help Online Tools and Resources	xi
	Opening a Case with JTAC	xi
Part 1	Overview	
Chapter 1	Software Features Overview	3
	SRC Component Overview	3
Chapter 2	Packet Mirroring Services	7
	Packet-Mirroring Services Overview	7
	Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API	8
Part 2	Configuration	
Chapter 3	Configuration Tasks for Packet Mirroring	11
	Configuring Packet-Mirroring Support in an SRC Network	11
	Configuring the Script Service for Packet Mirroring	12
	Configuring Parameters for the Script Service for Packet Mirroring	13
	Specifying Maximum Number of RADIUS Peers (SRC CLI)	15
Chapter 4	Example	17
	Example: Using the Sample Packet-Mirroring Application	17
	Example: Packet Mirroring for PPP Subscribers	18
	Example: Packet Mirroring for DHCP Subscribers	18
	Configuring DHCP Subscriber Sessions	18
	Disabling RADIUS Authentication for DHCP Subscribers	18

List of Tables

	About the Documentation	vii
	Table 1: Notice Icons	viii
	Table 2: Notice Icons	ix
	Table 3: Text Conventions	ix
Part 1	Overview	
Chapter 1	Software Features Overview	3
	Table 4: Descriptions of SRC Components	3
Part 2	Configuration	
Chapter 3	Configuration Tasks for Packet Mirroring	11
	Table 5: Parameter Substitutions for Packet-Mirroring Services	13

About the Documentation

- Documentation and Release Notes on page vii
- Documentation Conventions on page vii
- Documentation Feedback on page x
- Requesting Technical Support on page xi

Documentation and Release Notes

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

Documentation Conventions

[Table 1 on page viii](#) defines the notice icons used in this guide. [Table 3 on page ix](#) defines text conventions used throughout this documentation.

Table 2: Notice Icons

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

Table 3: Text Conventions

Convention	Description	Examples
Bold text like this	<ul style="list-style-type: none"> Represents keywords, scripts, and tools in text. Represents a GUI element that the user selects, clicks, checks, or clears. 	<ul style="list-style-type: none"> Specify the keyword exp-msg. Run the install.sh script. Use the pkgadd tool. To cancel the configuration, click Cancel.
Bold text like this	Represents text that the user must type.	user@host# set cache-entry-age <i>cache-entry-age</i>
Fixed-width text like this	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	<pre> nic-locators { login { resolution { resolver-name /realms/ login/A1; key-type LoginName; value-type SaeId; } } } </pre>

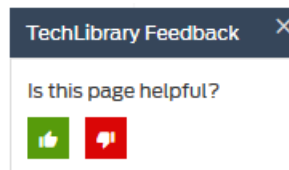
Table 3: Text Conventions (continued)

Regular sans serif typeface	<ul style="list-style-type: none"> Represents configuration statements. Indicates SRC CLI commands and options in text. Represents examples in procedures. Represents URLs. 	<ul style="list-style-type: none"> <code>system ldap server{ stand-alone;</code> Use the <code>request sae modify device failover command</code> with the <code>force</code> option <code>user@host# ...</code> https://www.juniper.net/techpubs/software/management/sdx/api-index.html
<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	<code>user@host# set local-address local-address</code>
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<i>Italic typeface</i>	<ul style="list-style-type: none"> Emphasizes words. Identifies book names. Identifies distinguished names. Identifies files, directories, and paths in text but not in command examples. 	<ul style="list-style-type: none"> There are two levels of access: <i>user</i> and <i>privileged</i>. <i>SRC-PE Getting Started Guide</i>. <i>o=Users, o=UMC</i> The <i>/etc/default.properties</i> file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	<code>Plugin.radiusAcct-1.class=\net.juniper.smgmt.sae.plugin\RadiusTrackingPluginEvent</code>
Words separated by the symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic line

Documentation Feedback

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

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



- Click the thumbs-up icon if the information on the page was helpful to you.

- Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- [Software Features Overview on page 3](#)
- [Packet Mirroring Services on page 7](#)

CHAPTER 1

Software Features Overview

- [SRC Component Overview on page 3](#)

SRC Component Overview

The SRC software is a dynamic system. It contains many components that you use to build a subscriber management environment. You can use these tools to customize and extend the SRC software for your use and to integrate the SRC software with other systems. The SRC software also provides the operating system and management tools for C Series Controllers.

[Table 4 on page 3](#) gives a brief description of the components that make up the SRC software.

Table 4: Descriptions of SRC Components

Component	Description
Server Components	
Service activation engine (SAE)	<ul style="list-style-type: none">• Authorizes, activates, and deactivates subscriber and service sessions by interacting with systems such as Juniper Networks routers, cable modem termination system (CMTS) devices, RADIUS servers, and directories.• Collects accounting information about subscribers and services from routers, and stores the information in RADIUS accounting servers, flat files, and other accounting databases.• Provides plug-ins and application programming interfaces (APIs) for starting and stopping subscriber and service sessions and for integrating with systems that authorize subscriber actions and track resource usage.
Subscriber Information Collector (SIC)	The SIC listens for RADIUS accounting events from IP edge devices (accounting clients) and forwards them to a remote AAA server, allowing the SRC software to gain increased subscriber awareness. Additionally, the SIC can optionally edit accounting events before routing them.
Network information collector (NIC)	Collects information about the state of the network and can provide a mapping from a given type of network data to another type of network data.
Redirect Server	Redirects HTTP requests received from IP Filter to a captive portal page.

Table 4: Descriptions of SRC Components (continued)

Component	Description
3GPP Gateway	The SRC Third-Generation Partnership Project (3GPP) gateway is a Diameter-based component in the SRC software, which provides integration with 3GPP Policy and Charging Control environments, to provide fixed-mobile convergence (FMC). The SRC 3GPP gateway provides Gx-based integration with the Policy and Charging Rules Function (PCRF). The SRC 3GPP gateway uses the northbound Gx interface to mediate between the PCRF and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. The northbound Gx interface on the SRC 3GPP gateway communicates with the PCRF using the Diameter protocol.
3GPP Gy	The SRC 3GPP Gy is a Diameter-based component in the SRC software, which provides Gy-based integration with the Online Charging System (OCS), to provide FMC. The SRC 3GPP Gy uses the northbound Gy interface to handle charging-related information between the OCS and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. The northbound Gy interface communicates with the OCS using the Diameter protocol.
Web Application Service	The SRC software includes a Web application server that hosts the Web Services Gateway and the Volume Tracking Application (SRC VTA). In production environments, this application server is designed to host only these applications. However, you can load your own applications into this server for testing or demonstration purposes.
Web Services Gateway	<p>Allows a gateway client—an application that is not part of the SRC network—to interact with SRC components through a Simple Object Access Protocol (SOAP) interface.</p> <p>The Web Services Gateway provides the Dynamic Service Activator which allows a gateway client to dynamically activate and deactivate SRC services for subscribers and to run scripts that manage the SAE.</p>
Repository	
Directory	<p>The SRC software includes the Juniper Networks database, which is a built-in Lightweight Directory Access Protocol (LDAP) directory for storing all SRC data including services, policies, and small subscriber databases.</p> <p>For large subscriber databases, you must supply your own directory.</p>
SRC Configuration and Management Tools	
SRC command line interface (CLI)	Provides a way to configure the SRC software on a C Series Controller from a Junos OS–like CLI. The SRC CLI includes the policies, services, and subscribers CLI, which has separate access privileges.
C-Web interface	Provides a way to configure, monitor, and manage the SRC software on a C Series Controller through a Web browser. The C-Web interface includes a policies, services, and subscribers component, which has separate access privileges.
Simple Network Management Protocol (SNMP) agent	Monitors system performance and availability. It runs on all the SRC hosts and makes management information available through SNMP tables and sends notifications by means of SNMP traps.
Service Management Applications (Run on external system)	
IMS Services Gateway	Integrates into an IP multimedia system (IMS) environment. The SRC software provides a Diameter protocol-based interface that allows the SRC software to integrate with services found on the application layer of IMS.

Table 4: Descriptions of SRC Components (continued)

Component	Description
SRC Programming Interfaces	
NETCONF API	Allows you to configure or request information from the NETCONF server on a C Series Controller that runs the SRC software. Applications developed with the NETCONF API run on a system other than a C Series Controller.
CORBA plug-in service provider interface (SPI)	Tracks sessions and enables linking the rest of the service provider's operations support system (OSS) with the SRC software so that the OSS can be notified of events in the life cycle of SAE sessions. Hosted plug-ins only.
CORBA remote API	Provides remote access to the SAE core API. Applications that use these extensions to the SRC software run on a system other than a C Series Controller.
NIC access API	Performs NIC resolutions. Applications that use these extensions to the SRC software run on a system other than a C Series Controller.
SAE core API	Controls the behavior of the SRC software. Applications that use these extensions to the SRC software run on a system other than a C Series Controller.
Script services	Provides an interface to call scripts that supply custom services such as provisioning policies on a number of systems across a network.
VTA API	The Volume Tracking Application (VTA) API is a Simple Object Access Protocol (SOAP) interface that allows developers to create gateway clients and that administrators use to manage VTA subscribers and sessions. The SRC Web Services Gateway allows a gateway client—an application that is not part of the SRC network—to interact with SRC components, such as the VTA, through a SOAP interface.
Authorization and Accounting Applications	
AAA RADIUS servers	Authenticates subscribers and authorizes their access to the requested system or service. Accepts accounting data—time active and volume of data sent—about subscriber and service sessions. RADIUS servers run on a system other than a C Series Controller.
SRC Admission Control Plug-In (SRC ACP)	Authorizes and tracks subscribers' use of network resources associated with services that the SRC application manages.
Flat file accounting	Stores tracking data to accounting flat files that can be made available to external systems that send the data to a rating and billing system.
Volume Tracking Application	<p>The SRC Volume Tracking Application (SRC VTA) is an SRC component that allows service providers to track and control the network usage of subscribers and services. You can control volume and time usage on a per-subscriber or per-service basis. This level of control means that service providers can offer tiered services that use volume as a metric, while also controlling abusive subscribers and applications.</p> <p>When a subscriber or service exceeds bandwidth limits (or quotas), the SRC VTA can take actions including imposing rate limits on traffic, sending an e-mail notification, or charging extra for additional bandwidth consumed.</p>
Demonstration Applications (available on the Juniper Networks Website)	

Table 4: Descriptions of SRC Components (continued)

Component	Description
Enterprise Audit Plug-In	Defines a callback interface, which receives events when IT managers complete specified operations.
Enterprise Manager Portal	<p>Allows service providers to provision services for enterprise subscribers on routers running JunosE or Junos OS and allows IT managers to manage services.</p> <p>Enterprise Manager Portal can be used with NAT Address Management Portal to allow service providers to manage public IP addresses for use with NAT services on routers running Junos OS and to allow IT managers to make requests about public IP addresses through the Enterprise Manager Portal.</p>
Monitoring Agent application	Integrates IP address managers, such as a DHCP server or a RADIUS server, into an SRC-managed network so that the SAE is notified about subscriber events. The Monitoring Agent application runs on a Solaris platform.
Residential service selection portals	Provides a framework for building Web applications that allow residential and enterprise subscribers to manage their own network services. It comes with several full-featured sample Web applications that are easy to customize and suitable for deployment. The Residential service selection portals run on a Solaris platform.
Sample enterprise service portal	Lets service providers supply an interface to their business customers for managing and provisioning services.

Related Documentation • *SRC Product Description*

CHAPTER 2

Packet Mirroring Services

- [Packet-Mirroring Services Overview on page 7](#)
- [Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 8](#)

Packet-Mirroring Services Overview

Packet mirroring allows you to mirror subscriber traffic by configuring a script service with the SRC software that applies policies on a router running JunosE Software for RADIUS-based packet mirroring.

When the service activation engine (SAE) activates a packet-mirroring service session, the session sends dynamic RADIUS requests, such as change-of-authorization (COA) messages, to a RADIUS device such as a router running JunosE Software.

In RADIUS-based packet mirroring on a router running JunosE Software, a RADIUS administrator uses RADIUS attributes to configure packet mirroring of a particular subscriber's traffic. The router creates dynamic secure policies for the mirroring operation. The original traffic is sent to its intended destination, and the mirrored traffic is sent to an analyzer device (the mediation device). The mirroring operations are transparent to the subscriber whose traffic is being mirrored. This dynamic method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber whose traffic is to be mirrored and to trigger the mirroring session. RADIUS-based mirroring uses dynamically created secure policies based on certain RADIUS VSAs. You attach the secure policies to the interface used by the mirrored subscriber. The packet-mirroring VSAs that the RADIUS server sends to the E Series router are MD5 salt-encrypted.

You must deploy RADIUS-based packet mirroring on routers running JunosE Software to monitor the subscriber traffic.

Related Documentation

- [Configuring Packet-Mirroring Support in an SRC Network on page 11](#)
- [Configuring the Script Service for Packet Mirroring on page 12](#)
- [Configuring Parameters for the Script Service for Packet Mirroring on page 13](#)
- [Example: Using the Sample Packet-Mirroring Application on page 17](#)

Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API

The SRC software provides two ways to define RADIUS attributes for dynamic RADIUS authorization requests:

- Service definition
- SAE core API



NOTE: Parameters set in the API override parameters set by the service definition.

To send dynamic RADIUS authorization requests with the SAE core API, the script service uses the `sendDynamicRadius` and `getRouterDynRadiusAddr` methods in the `ServiceSessionInfo` interface to provide the content of the RADIUS packet for the dynamic authorization request to the router running JunosE Software that is attached to the service session.

For information about the `ServiceSessionInfo` interface, see the script service documentation in the SAE core API documentation on the Juniper Networks website at

<https://www.juniper.net/documentation/software/management/src/api-index.html>

For a sample implementation, see the following file in the `SDK+AppSupport+Demos+Samples.tar.gz` file:

`SDK/scriptServices/packetMirroring/java/net/juniper/smgmt/scriptServices/packetMirroring/LiService.java`.

Related Documentation

- [Configuring Parameters for the Script Service for Packet Mirroring on page 13](#)

PART 2

Configuration

- [Configuration Tasks for Packet Mirroring on page 11](#)
- [Example on page 17](#)

CHAPTER 3

Configuration Tasks for Packet Mirroring

- [Configuring Packet-Mirroring Support in an SRC Network on page 11](#)
- [Configuring the Script Service for Packet Mirroring on page 12](#)
- [Configuring Parameters for the Script Service for Packet Mirroring on page 13](#)
- [Specifying Maximum Number of RADIUS Peers \(SRC CLI\) on page 15](#)

Configuring Packet-Mirroring Support in an SRC Network

To support packet mirroring in an SRC network, configure a script service that can be activated to set up RADIUS-based packet-mirroring policies on a router running JunosE Software. The script service defines the parameters needed to mirror subscriber traffic, such as the address of the subscriber or the analyzer device. This script service is activated for the subscriber whose traffic should be mirrored.

You must have preconfigured RADIUS-based packet mirroring on routers running JunosE Software. The JunosE software provides RADIUS-based packet mirroring, which allows the router to create dynamic secure policies for the mirroring operation. The RADIUS administrator can configure and manage interface mirroring services that are activated by means of COA.

To set up the SRC software for packet mirroring:

- Create a script service for packet mirroring.

The SRC software includes a sample script service that you can configure to send dynamic RADIUS requests to the router running JunosE Software. You can use the sample service definition and customize it for your environment by modifying the service substitutions.

See [“Configuring Parameters for the Script Service for Packet Mirroring” on page 13](#).

- Configure subscriptions to the packet-mirroring service.

You can set up the subscriptions to activate immediately on login.

See *Configuring Subscriptions (SRC CLI)*.

- (Optional) Configure the maximum number of RADIUS peers.

See [“Specifying Maximum Number of RADIUS Peers \(SRC CLI\)” on page 15](#).

For information about configuring RADIUS-based packet mirroring on the router running JunosE Software, see the *JunosE Policy Management Configuration Guide*.

For information about dynamic RADIUS requests, see RFC 3576—Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) (July 2003).

**Related
Documentation**

- [Configuring the Script Service for Packet Mirroring on page 12](#)
- [Example: Using the Sample Packet-Mirroring Application on page 17](#)
- [Packet-Mirroring Services Overview on page 7](#)

Configuring the Script Service for Packet Mirroring

To configure the script service for packet mirroring:

1. Create a script service in the services global service **name** hierarchy or the services scope **name** service **name** hierarchy. For example:

```
[edit]
user@host# edit services global service packetMirroring
```

2. Set the type to script.

```
[edit services global service packetMirroring]
user@host# set type script
```

3. (Optional) Configure other properties as needed for your service.
4. Configure the script properties.

- a. Access the script hierarchy for the configured script service.

```
[edit services global service packetMirroring]
user@host# edit script
```

- b. Specify URL as the script type.

```
[edit services global service packetMirroring script]
user@host# set script-type url
```

- c. Specify the name of the Java class that implements the script service.

```
[edit services global service packetMirroring script]
user@host# set class-name net.juniper.smgt.sae.packetMirroring.LiService
```

- d. Configure the URL of the script service or the path and filename of the service.

```
[edit services global service packetMirroring script]
user@host# set file file:///opt/UMC/sae/var/run/pm.jar
```


5. Verify the configuration.

```
[edit services global service packetMirroring script]
user@host# show
type script;
status active;
available;
script {
  script-type url;
  class-name net.juniper.smgmt.sae.packetMirroring.LiService;
  file file:///opt/UMC/sae/var/run/pm.jar;
}
```

6. Configure the parameters for the script service.

See [“Configuring Parameters for the Script Service for Packet Mirroring” on page 13](#).

Related Documentation

- [Configuring Packet-Mirroring Support in an SRC Network on page 11](#)
- [Adding a Normal Service \(SRC CLI\)](#)
- [Customizing Service Implementations](#)
- [Example: Using the Sample Packet-Mirroring Application on page 17](#)
- [SRC Script Services Overview](#)
- [Packet-Mirroring Services Overview on page 7](#)

Configuring Parameters for the Script Service for Packet Mirroring

Provide parameter substitutions with the values that are in the service definitions for the script service.

[Table 5 on page 13](#) lists the parameters specified by the sample packet-mirroring script service. In most cases, you can use the sample script service without modification.

Table 5: Parameter Substitutions for Packet-Mirroring Services

Parameter Name	Description
dynAnalyzerIPAddress	RADIUS VSA that is the IP address of the analyzer device. This attribute is required.
dynAnalyzerPortNumber	RADIUS VSA that is the UDP port number of the monitoring application in the analyzer device. If specified, dynMirrorIdentifier must also be specified.
dynMirrorIdentifier	RADIUS VSA in the form of a hexadecimal string. If specified, dynAnalyzerPortNumber must also be specified.
dynClientIp	IP address of the dynamic RADIUS client.

Table 5: Parameter Substitutions for Packet-Mirroring Services (continued)

Parameter Name	Description
dynClientPort	UDP port number of the dynamic RADIUS client.
dynServerIp	IP address of the C Series Controller.
dynServerPort	UDP port number of the C Series Controller.
dynSecret	Shared secret.
dynRetry	Number of retries for sending dynamic RADIUS packet when no RADIUS response is received. The retry interval is 3 seconds.
dynConfig	<p>Content of dynamic RADIUS request packets in the format <action>. <radiusAttributeName>=<pluginEventAttribute>\n</p> <ul style="list-style-type: none"> • action—Action that is executed on packet content (attribute) <ul style="list-style-type: none"> • start • stop • start-stop • radiusAttributeName—Valid RADIUS attribute specified as follows: <ul style="list-style-type: none"> • Standard RADIUS attribute name or number. • JunosE VSA in one of the following formats: vendor-specific.4874.<vsa#>[.salt] 26.4874.<vsa#>[.salt] where .salt indicates that the attribute is MD5 salt-encrypted in the RADIUS packet. • pluginEventAttribute—Valid Python expression • \n—New-line character included between the lines of a configuration containing multiple lines; the entire configuration must be enclosed in quotation marks <p>For example:</p> <pre>start-stop.Acct-Session-Id = ifSessionId " start-stop.Acct-Session-Id=ifSessionId\n start.vendor-specific.4874.58.salt=1\n start.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt= custom['dynMirrorIdentifier'] \n start.vendor-specific. JUNIPER.Unisphere-Med-Ip-Address.salt= intIp(custom['dynAnalyzerIpAddress'])\n start.vendor-specific. JUNIPER.Unisphere-Med-Port-Number.salt= int(custom ['dynAnalyzerPortNumber'])\n stop.vendor- specific.4874.58.salt=0"</pre>

To configure substitutions for the script parameters:

1. At the hierarchy for the script service, specify substitutions for the parameters. For example:

```
[edit services global service packetMirroring]
user@host# set parameter substitution [ dynAnalyzerIpAddress=10.227.6.221
dynAnalyzerPortNumber=9100 dynMirrorIdentifier=0x0000000100000001
```

```

dynSecret="\secret\" dynRetry=2 dynClientIp=10.227.7.111 dynClientPort=9099
"dynConfig=\"start-stop.Acct-Session-Id =
ifSessionId\\nstart.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=1\\nstar
t.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=custom['dynMirrorIden
tifier']\\nstart.vendor-specific.JUNIPER.Unisphere-Med-IP-Address.salt=intIp(c
ustom['dynAnalyzerIPAddress'])\\nstart.vendor-specific.JUNIPER.Unisphere-Me
d-Port-Number.salt =
int(custom['dynAnalyzerPortNumber'])\\nstop.vendor-specific.JUNIPER.Unisph
ere-LI-Action.salt=0\""] ]

```

2. Verify the configuration.

```

[edit services global service packetMirroring]
user@host# show
type script;
status active;
parameter {
  substitution [ dynAnalyzerIPAddress=10.227.6.221 dynAnalyzerPortNumber=9100
dynMirrorIdentifier=0x0000000100000001 dynSecret=secret dynRetry=2
dynClientIp=10.227.7.111 dynClientPort=9099 "dynConfig=\"start-stop.
Acct-Session-Id = ifSessionId\\nstart.vendor-specific.JUNIPER.
Unisphere-LI-Action.salt= 1\\nstart.vendor-specific.JUNIPER.
Unisphere-Med-Dev-Handle.salt= custom['dynMirrorIdentifier']
\\nstart.vendor-specific.JUNIPER.
Unisphere-Med-IP-Address.salt= intIp(custom['dynAnalyzerIPAddress'])
\\nstart.vendor-specific.JUNIPER.
Unisphere-Med-Port-Number.salt = int(custom['dynAnalyzerPortNumber'])
\\nstop.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=0\""] ];
}
script {
  script-type url;
  class-name net.juniper.smgmt.scriptServices.packetMirroring.LiService;
  file file:///opt/UMC/sae/lib/pm.jar;
}

```

Related Documentation

- [Configuring Packet-Mirroring Support in an SRC Network on page 11](#)
- [Adding a Normal Service \(SRC CLI\)](#)
- [Setting Parameter Values for Services \(SRC CLI\)](#)
- [Customizing Service Implementations](#)
- [Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 8](#)

Specifying Maximum Number of RADIUS Peers (SRC CLI)

The dynamic RADIUS server can maintain a certain number of peers.

To specify the maximum number of peers with the SRC CLI:

1. From configuration mode, access the SAE configuration statement that configures dynamic RADIUS options.

```
[edit]  
user@host# edit shared sae configuration dynamic-radius-server
```

2. Specify the maximum number of peers maintained by the dynamic RADIUS server.

```
[edit shared sae configuration dynamic-radius-server]  
user@host# set maximum-cached-peer maximum-cached-peer
```

**Related
Documentation**

- [Configuring Packet-Mirroring Support in an SRC Network on page 11](#)
- [Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 8](#)
- [Example: Using the Sample Packet-Mirroring Application on page 17](#)
- [Packet-Mirroring Services Overview on page 7](#)

CHAPTER 4

Example

- Example: Using the Sample Packet-Mirroring Application on page 17

Example: Using the Sample Packet-Mirroring Application

To use the sample packet-mirroring application:

1. Download the SRC sample applications to your system from the Juniper Networks website:

<https://www.juniper.net/support/downloads/?p=src#sw>

2. Locate the file that contains the service definition:

`/SDK/scriptServices/packetMirroring/ldif/service.ldif`

3. Import the sample service definition to the Juniper Networks Database on the C Series Controller. To load the sample data into the database, you can use an LDAP tool, such as **ldapadd**.

You can obtain **ldapadd** from the following website:

<http://www.openldap.org/>

To load data into the Juniper Networks database, you need the IP address of the database and the database credentials. The default bind distinguished name (DN) for the database is `cn=umcadmin,o=umc` and the password is `admin123`.

4. Copy the `/lib/pm.jar` file used by the script service to the `/opt/UMC/sae/var/run` directory on the C Series Controller.

5. Modify the service substitutions for your environment.

You can make these substitutions by defining the parameter substitutions in the packetMirroring service (`serviceName=packetMirroring, o=Services, o=umc`) with the SRC CLI or by passing the values through the SAE core API.

For information about parameter substitutions, see “Configuring Parameters for the Script Service for Packet Mirroring” on page 13. For information about passing the values through the SAE core API, see “Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API” on page 8.

6. Configure a subscription to the packetMirroring service that is activated on login.

For information about subscriptions, see *Subscriptions Overview*.

7. If you are modifying the sample application, copy the *sae.jar* and *logger.jar* files from the *SKD/lib* directory, and add the *sae.jar* and *logger.jar* files to the class path when you compile your application.

Example: Packet Mirroring for PPP Subscribers

When a PPP subscriber is subscribed to the packet-mirroring service, configure the service as an activate-on-login service at user connection time. After the subscriber has logged in through the SAE remote API, the packet-mirroring service can be subscribed to the PPP subscriber and activated. When the service is activated, a COA request is sent to the router running JunosE Software that includes the PPP subscriber's accounting session ID to start packet mirroring for this subscriber.

Example: Packet Mirroring for DHCP Subscribers

When a DHCP subscriber is subscribed to the packet-mirroring service, configure the service as an activate-on-login service at user connection time. After the subscriber has logged in through the SAE remote API, the packet-mirroring service can be subscribed to the DHCP subscriber and activated. When the service is activated, a COA request is sent to the router running JunosE Software that includes the DHCP subscriber's IP address and virtual router name for the router running JunosE Software to start packet mirroring for this subscriber.

Configuring DHCP Subscriber Sessions

You can use DHCP option 82 to identify the subscriber session. For example, if you set DHCP option 82 as the user login name, an external application can use this setting to search for the subscriber session. The following subscriber classification script illustrates this example:

```
[retailername=default,o=Users,o=UMC?loginName=<-dhcp[82].suboptions[1].string
->?sub?(interfaceName=<-dhcp[82].suboptions[1].string->)]
loginType = " ADDR"
[<-retailerDN->??sub?(uniqueID=<-userName->)]
retailerDN != " "
& userName != " "
[<-unauthenticatedUserDn->]
loginType == "ADDR"
loginType == "AUTHADDR"
```

Disabling RADIUS Authentication for DHCP Subscribers

Packet mirroring for DHCP subscribers does not involve RADIUS authentication, so you might have to configure authentication to grant all IP subscriber management interfaces access without authentication. For example, configure the router running JunosE Software with the following authentication:

```
aaa authentication ip default none
```

You can still configure other subscribers to use RADIUS authentication. For example, configure the router running JunosE Software with the following authentication for PPP subscribers:

```
aaa authentication ppp default radius
```

**Related
Documentation**

- [Configuring Packet-Mirroring Support in an SRC Network on page 11](#)
- [Packet-Mirroring Services Overview on page 7](#)

