Chapter 14

Providing Packet Mirroring in the SRC Network

This chapter describes how the SRC network handles traffic mirroring on a JUNOSe router and how to configure policies, services, and subscribers that support RADIUS-based packet mirroring. Topics include:

Overview of Packet Mirroring

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

When the 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 JUNOSe router.

In RADIUS-based packet mirroring on a JUNOSe router, 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 JUNOSe routers to monitor the subscriber traffic.

Configuring Packet Mirroring

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 JUNOSe router. 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. For detailed information about configuring script services, see *SRC-PE Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform*.

You must have preconfigured RADIUS-based packet mirroring on JUNOSe routers. 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. For information about configuring RADIUS-based packet mirroring on the JUNOSe router, 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).

To set up the SRC software for packet mirroring, perform the following tasks:

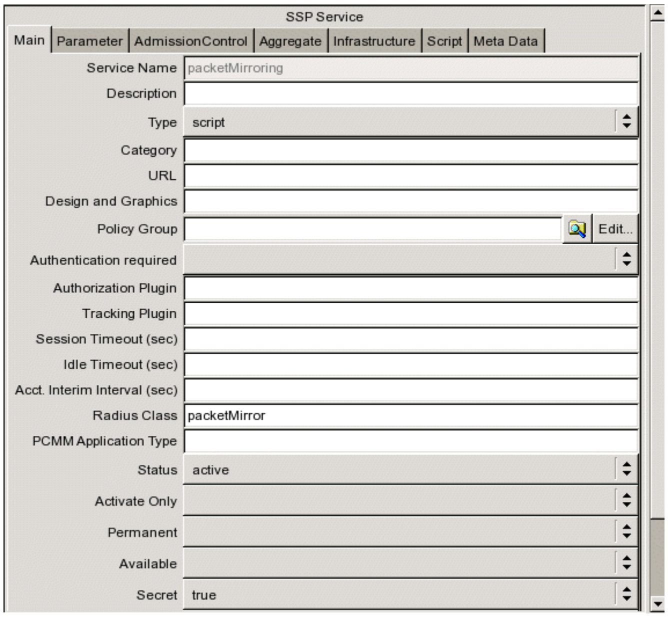
- Creating the Script Service for Packet Mirroring on page 135
- *Configuring the Script Service for Packet Mirroring* on page 137
- Configuring Subscriptions to the Packet-Mirroring Service on page 138
- (Optional) Specifying Maximum Number of Peers on page 139

The SRC software includes a sample script service that you can configure to send dynamic RADIUS requests to the JUNOSe router. You can use the sample service definition and customize it for your environment by modifying the service substitutions. For information about the sample packet mirroring application, see *Example: Using the Sample Packet-Mirroring Application* on page 139.

Creating the Script Service for Packet Mirroring

To create the script service:

1. In the SDX Admin navigation pane, right-click the Services folder, highlight **New**, and then click **SSP Service**.
2. In the New SSP Service dialog box, enter a service name or select a name from the drop-down list.
3. In the Main tab pane, select **script** in the Type field.
4. If you want to hide the service from users and unauthorized administrators, select **true** from the menu in the Secret field.



Service Name packetMirroring

Type script

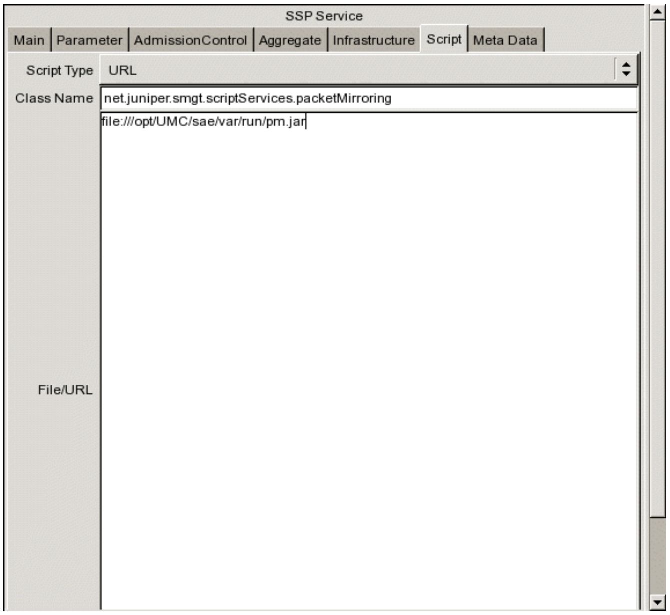
Radius Class packetMirror

Status active

Secret true

5. Click the Script tab.

   The Script pane appears.



Script Type URL

Class Name net.juniper.smgt.scriptServices.packetMirroring

File/URL file:///opt/UMC/sae/var/run/pm.jar

6. Edit the values in the Script fields for the sample packet-mirroring script service.

   - In the Script Type field, select **URL**.
   - In the Class Name field, enter net.juniper.smgt.sae.packetMirroring.LiService.
   - In the File/URL field, enter *file:///opt/UMC/sae/var/run/pm.jar*.

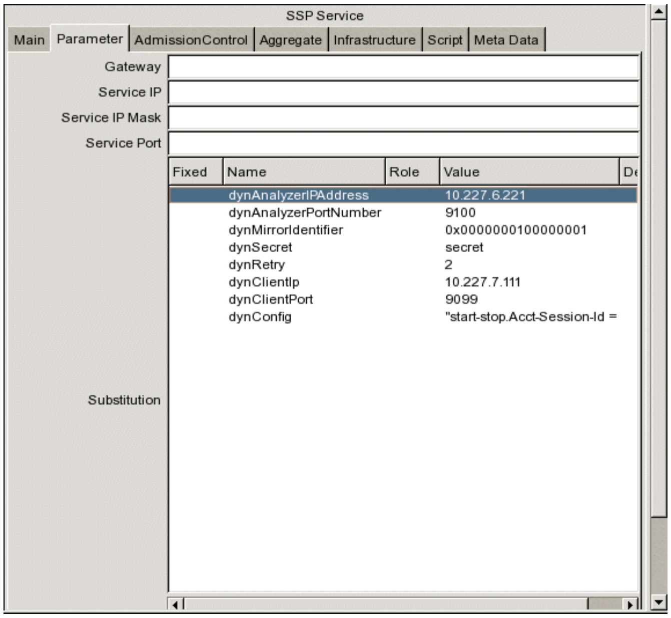
7. Click **Save**.

After you create the script service, you need to configure parameters for the script service. For more information about configuring script services and parameters, see *SRC-PE Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform*.

Configuring the Script Service for Packet Mirroring

To configure the script service, you provide parameter substitutions with the values that are in the service definitions. To do so:

1. In SDX Admin, select the Parameter tab in the script service configuration. The parameter pane appears.



Fixed	Name	Role	Value	De
	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 =	

2. Configure the parameters.

   Table 7 lists the parameters specified by the sample packet-mirroring script service. In most cases, you can use the sample script service without modification.

Table 7: 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.
dynClientPort	UDP port number of the dynamic RADIUS client.
dynSecret	Shared secret.
dynRetry	Number of retries for sending dynamic RADIUS packet when no RADIUS response is received. The retry interval is 3 seconds.

Table 7: Parameter Substitutions for Packet-Mirroring Services (continued)

Parameter Name	Description
dynConfig	Content of dynamic RADIUS request packets in the format <action>. <radiusAttributeName> = <pluginEventAttribute>\n - action—Action that is executed on packet content (attribute) - start - stop - start-stop - radiusAttributeName—Valid RADIUS attribute specified as follows: - 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 For example: start-stop.Acct-Session-Id = ifSessionId “start-stop.Acct-Session-Id = ifSessionId\nstart.vendor-specific.4874.58.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.4874.58.salt = 0"

You can also configure dynamic RADIUS requests with the sendDynamicRadius method of the ServiceSessionInfo interface (see *Defining RADIUS Attributes for Dynamic Authorization Requests with the API* on page 141).

For detailed information about configuring services, see *SRC-PE Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform*.

Configuring Subscriptions to the Packet-Mirroring Service

You need to configure subscriptions to the packet-mirroring service. You can set up the subscriptions to activate immediately on login.

For more information, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 13, Configuring Subscribers and Subscriptions with SDX Admin*.

Specifying Maximum Number of Peers

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
   ```

Example: Using the Sample Packet-Mirroring Application

To use the sample packet-mirroring application provided:

1. Import the sample service definition using an LDAP browser.

   The */SDK/scriptServices/packetMirroring/ldif/service.ldif* file (in the SRC software distribution) is the sample service definition.

2. Copy the */lib/pm.jar* file used by the script service to the */var/run* directory in the SAE installation directory (*/opt/UMC/sae* by default).

3. 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 SDX Admin or by passing the values through the SAE core API.

   For information about parameter substitutions, see *Configuring the Script Service for Packet Mirroring* on page 137. For information about passing the values through the SAE core API, see *Defining RADIUS Attributes for Dynamic Authorization Requests with the API* on page 141.

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

   For more information about subscriptions, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 13, Configuring Subscribers and Subscriptions with SDX Admin*.

If you are modifying the sample application, add the *sae.jar* and *logger.jar* files to the classpath when you compile your application. These two files can be found in the *lib* directory of the SAE installation directory.

Example: Packet Mirroring for PPP Subscribers

When a PPP subscriber is subscribed to the packet-mirroring service, the service should be configured 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 JUNOSe router 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, the service should be configured 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 JUNOSe router that includes the DHCP subscriber's IP address and virtual router name for the JUNOSe router 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 JUNOSe router with the following authentication:

```
aaa authentication ip default none
```

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

```
aaa authentication ppp default radius
```

Defining RADIUS Attributes for Dynamic Authorization Requests with the API

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

- Service definition (see *Configuring the Script Service for Packet Mirroring* on page 137)
- 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 JUNOSe router that is attached to the service session.

For information about the ServiceSessionInfo interface, see the script service documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/sdx/api-index.html

For a sample implementation, see the following file in the SRC software distribution:

SDK/scriptServices/packetMirroring/java/net/juniper/smgt/scriptServices/packetMirroring/LiService.java