Chapter 15

Managing Services on Third-Party Devices in the SRC Network

This chapter describes how the SRC network manages services on third-party devices that do not support COPS by exchanging change-of-authorization (CoA) messages. Topics include:

Overview of CoA Script Service

The SAE can use change-of-authorization (CoA) messages to manage services for a specific subscriber session. The CoA script service allows the SAE to exchange CoA messages with third-party devices that do not support Common Open Policy Service (COPS) protocol to activate or deactivate services for specific subscriber sessions. When the SAE activates a CoA script service session, the session sends CoA messages to a RADIUS-enabled device. This method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber session whose services are to be activated or deactivated.

Configuring CoA Script Services

To support CoA message exchange in an SRC network, configure a script service that can be activated on a third-party device. The script service defines the parameters needed to activate or deactivate services for a subscriber session, such as the address of the third-party device. This script service is activated for the subscriber session whose services are activated or deactivated. For detailed information about configuring script services, see *SRC-PE Services and Policies Guide, Chapter 1, Managing Services with the SRC CLI*.

When you use the CoA script service with third-party devices that do not notify the SAE about subscriber events, you must set up the Monitoring Agent application to handle RADIUS accounting request packets.

For information about configuring services on the third-party device, see the device's software documentation.

The tasks to set up the SRC software for CoA message exchange are:

- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 144
- Creating the CoA Script Service with the SRC CLI on page 144
- *Configuring the CoA Script Service with the SRC CLI* on page 145
- Configuring Subscriptions to the CoA Script Service on page 147

The SRC software includes a sample script service that you can configure to exchange CoA messages with the third-party device. You can use the sample service definition and customize it for your environment by modifying the service substitutions. For information about the sample CoA script service, see *Example: Using the Sample CoA Script Service* on page 147.

Configuring Monitoring Agent to Receive RADIUS Accounting Messages

If you install the Monitoring Agent application on the same host as the RADIUS server, you must disable the MonAgent.radius.server property.

You can configure Monitoring Agent to act as a pseudo–RADIUS server that listens for RADIUS accounting packets sent to the RADIUS accounting port. To receive RADIUS packets from RADIUS clients:

- Make sure there is no other RADIUS server listening on the RADIUS accounting port, and enable the MonAgent.radius.server property.
- Configure the shared secret between the RADIUS server and the RADIUS client by specifying the MonAgent.radius.secret.<IP address> property.

For information about installing and using Monitoring Agent, see the *SRC Sample Applications Guide*.

Creating the CoA Script Service with the SRC CLI

To create the script service:

1. From configuration mode, enter the service configuration. In this sample procedure, the service is configured in the global service scope, and CoAservice is the name of the service.

```
user@host# edit services global service CoAservice
```

2. Configure the type of service.

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

3. (Optional) Specify whether the service is visible only to administrators who have permission to see secret information.

   ```
   [edit services global service CoAservice]
   user@host# set secret
   ```

4. Configure URL as the type of script that the sample CoA script service uses.

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

5. Configure net.juniper.smgt.sae.coa.CoaService as the name of the class that implements the script service.

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

6. Configure the URL of the script service or the path and filename of the service. Copy the */lib/coa.jar* file used by the script service to a location that is accessible by a URL (such as an FTP or HTTP server). In this sample procedure, the *coa.jar* file was copied to the */opt/UMC/sae/var/run* directory.

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

7. (Optional) Verify your configuration.

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

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 1, Managing Services with the SRC CLI*.

Configuring the CoA Script Service with the SRC CLI

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

To configure parameters:

1. From configuration mode, enter the service parameter configuration. In this sample procedure, the service called CoAservice is configured in the global service scope.

   ```
   user@host# edit services global service CoAservice parameter
   ```

2. (Optional) Configure actual values for other parameters.

```
[edit services global service CoAservice parameter]
user@host# set substitution [substitution...]
```

Table 8 lists the parameters specified by the sample CoA script service, which is the */SDK/scriptServices/coa/ldif/BOD1M.ldif* file in the SRC software distribution. You can use the sample script service as a starting point.

Table 8: Parameter Substitutions for CoA Services

Parameter Name	Description
dynClientIp	IP address of the third-party device.
dynClientPort	UDP port number of the third-party device.
dynSecret	Shared secret between RADIUS server and RADIUS client.
dynRetry	Number of retries for sending CoA messages when no RADIUS response is received. The retry interval is 3 seconds.
dynConfig	Content of service definition in the format <action>.<radiusAttributeName> = <pluginEventAttribute>\n ■ action—Action that is executed on packet content (attribute): ■ start ■ stop ■ start-stop ■ svcstart ■ svcstop ■ radiusAttributeName—Valid RADIUS attribute specified as follows: ■ Standard RADIUS attribute name or number ■ Third-party VSA in the format vendor-specific.<vendor#>.<vsa#>.string ■ pluginEventAttribute—Valid expression in the format: ■ Python expression ■ <commandCode> <serviceName>; the entire expression must be enclosed in single quotation marks and you must use three backslashes (\\\) to escape the backslash that starts a <commandCode> For example: \x0b would be replaced by \\\\x0b ■ \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\nsvcstart.vendor-specific.9.252.string=’\\\\x0bBOD1M’\nsvcstop.vendor-specific.9.252.string=’\\\\x0cBOD1M’\n"

You can also configure dynamic RADIUS requests with the sendDynamicRadius method of the ServiceSessionInfo interface (see *Defining RADIUS Attributes for CoA Requests with the API* on page 148).

For detailed information about configuring services, see *SRC-PE Services and Policies Guide, Chapter 1, Managing Services with the SRC CLI*.

Configuring Subscriptions to the CoA Script Service

You need to configure subscriptions to the CoA script service. You can set up the subscriptions to activate immediately on login.

For more information, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 12, Configuring Subscribers and Subscriptions with the SRC CLI*.

Example: Using the Sample CoA Script Service

To use the sample CoA script service provided:

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

   The */SDK/scriptServices/coa/ldif/BOD1M.ldif* file (in the SRC software distribution) is the sample service definition for exchanging CoA messages with a Cisco 10000 Series router.

2. Copy the */lib/coa.jar* file used by the script service to a location that is accessible to the SAE by a URL, such as an FTP or HTTP server. If you do not have multiple SAEs, it can be convenient to copy the file to the */var/run* directory in the SAE installation directory (*/opt/UMC/sae* by default).

3. Modify the service substitutions for your device.

   You can make these substitutions by defining the parameter substitutions in the BOD1M service with the SRC CLI or by passing the values through the SAE core API.

   For information about parameter substitutions, see *Configuring the CoA Script Service with the SRC CLI* on page 145. For information about passing the values through the SAE core API, see *Defining RADIUS Attributes for CoA Requests with the API* on page 148.

4. Configure a subscription to the BOD1M service that is activated on login.

   For more information about subscriptions, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 12, Configuring Subscribers and Subscriptions with the SRC CLI*.

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.

Defining RADIUS Attributes for CoA Requests with the API

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

- Service definition (see *Configuring the CoA Script Service with the SRC CLI* on page 145)
- 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 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/src/api-index.html

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

SDK/scriptServices/coa/java/net/juniper/smgt/scriptServices/coa/CoaService.java