

Configuring Authentication Plug-Ins

You can perform the following tasks to configure authentication plug-ins:

1. Limiting Subscribers on Router Interfaces on page 1
2. Configuring Basic RADIUS Authentication Plug-Ins on page 2
3. Configuring Flexible RADIUS Authentication Plug-Ins on page 3
4. Configuring Custom RADIUS Authentication Plug-Ins on page 6
5. Configuring LDAP Authentication Plug-Ins on page 8

Limiting Subscribers on Router Interfaces

You can limit the number of authenticated subscribers who connect to an IP interface on the router. This plug-in does not limit the number of unauthenticated subscribers who connect to an IP interface, and does not limit the number of subscribers who connect to a physical or link-layer interface. In the case of subscriber interfaces, the plug-in limits the number of authenticated subscribers on the subscriber interface but not on the underlying primary IP interface.

Use the following configuration statement to set up a plug-in that limits the number of subscribers who connect to interfaces:

```
shared sae configuration plug-ins name name interface-subscriber-limit {  
    concurrent-subscribers concurrent-subscribers ;  
}
```

To set up a plug-in that limits the number of subscribers on interfaces:

1. From configuration mode, access the custom RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called subsLimit is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
subsLimit interface-subscriber-limit
```

2. Configure the number of authenticated subscribers who can connect to an IP interface on the router simultaneously.

```
[edit shared sae group west-region configuration plug-ins name subsLimit  
interface-subscriber-limit]  
user@host# set concurrent-subscribers concurrent-subscribers
```

3. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name subsLimit  
interface-subscriber-limit]  
user@host# show  
concurrent-subscribers 1;
```

Configuring Basic RADIUS Authentication Plug-Ins

You can use basic RADIUS authentication plug-ins to send authentication information to an external RADIUS accounting server or a group of redundant servers. To communicate with nonredundant servers, you need to create additional instances of the plug-in.

Use the following configuration statements to set up basic RADIUS authentication plug-ins:

```
shared sae configuration plug-ins name name radius-authentication {  
    load-balancing-mode (failover | roundRobin);  
    fallback-timer fallback-timer ;  
    nas-ip (Ssplp | Erxlp);  
    retry-interval retry-interval ;  
    maximum-queue-length maximum-queue-length ;  
    bind-address bind-address ;  
    udp-port udp-port ;  
    default-peer default-peer ;  
}
```

To set up basic RADIUS authentication plug-ins:

1. From configuration mode, access the basic RADIUS authentication plug-in configuration. In this sample procedure, the plug-in called RadiusAuth is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
RadiusAuth radius-authentication
```

2. Configure the mode for load-balancing RADIUS servers.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth  
radius-authentication]  
user@host# set load-balancing-mode (failover | roundRobin)
```

3. Specify if and when the SAE attempts to fail back to the default peer.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth  
radius-authentication]  
user@host# set fallback-timer fallback-timer
```

4. (Optional) Configure the value of the NAS-Ip attribute.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth  
radius-authentication]  
user@host# set nas-ip (Ssplp | Erxlp)
```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth  
radius-authentication]  
user@host# set retry-interval retry-interval
```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
radius-authentication]
user@host# set maximum-queue-length maximum-queue-length
```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
radius-authentication]
user@host# set bind-address bind-address
```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
radius-authentication]
user@host# set udp-port udp-port
```

9. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
radius-authentication]
user@host# set default-peer default-peer
```

10. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
radius-authentication]
user@host# show
load-balancing-mode failover;
failback-timer -1;
retry-interval 3000;
maximum-queue-length 10000;
default-peer peer1;
```

Configuring Flexible RADIUS Authentication Plug-Ins

Flexible RADIUS authentication plug-ins provide the same features as basic RADIUS authentication plug-ins. In addition, they allow you to customize RADIUS authentication packets that the system sends to RADIUS servers and specify which fields are included in the RADIUS authentication packets and what information is contained in the fields.

Use the following configuration statements to set up flexible RADIUS authentication plug-ins:

```
shared sae configuration plug-ins name name flex-radius-authentication {
```

```

load-balancing-mode (failover | roundRobin);
failback-timer failback-timer ;
timeout timeout ;
retry-interval retry-interval ;
maximum-queue-length maximum-queue-length ;
bind-address bind-address ;
udp-port udp-port ;
error-handling (0 | 1);
default-peer default-peer;
template template ;
}

```

To set up flexible RADIUS authentication plug-ins:

1. From configuration mode, access the flexible RADIUS authentication plug-in configuration. In this sample procedure, the plug-in called flexRadiusAuth is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
flexRadiusAuth flex-radius-authentication

```

2. Configure the mode for load-balancing RADIUS servers.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set load-balancing-mode (failover | roundRobin)

```

3. Specify if and when the SAE attempts to fail back to the default peer.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set failback-timer failback-timer

```

4. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set timeout timeout

```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set retry-interval retry-interval

```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set maximum-queue-length maximum-queue-length

```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set bind-address bind-address
```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set udp-port udp-port
```

9. Configure the way the SAE handles errors.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set error-handling (0 | 1)
```

10. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set default-peer default-peer
```

11. Configure the name of the RADIUS packet template that defines attributes for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set template template
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name
flexRadiusAuth flex-radius-authentication]
user@host# show
load-balancing-mode failover;
failback-timer -1;
timeout 15000;
retry-interval 3000;
maximum-queue-length 10000;
error-handling 0;
default-peer 1;
template stdAuth;
peer-group 1 {
    server-address ;
    server-port 1812;
    secret *****;
}
```

Configuring Custom RADIUS Authentication Plug-Ins

The custom RADIUS authentication plug-ins provide the same functions as the flexible RADIUS authentication plug-ins, but are designed to deliver better system performance. To use a custom plug-in, you must provide a Java class that implements the SPI defined in the RADIUS client library. Use this SPI to specify which fields and field values to include in RADIUS accounting packets. The RADIUS client library is part of the SAE core API.

See the documentation for the RADIUS client library 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 in the `SDK+AppSupport+Demos+Samples.tar.gz` file on the Juniper Networks Web site at:

<https://www.juniper.net/support/csc/swdist-erx/src.html> The application is located the following directory:

`SDK/plugin/java/src/net/juniper/smg/sample/radiuslib/RadiusPacketHandlerImpl.java`.

Use the following configuration statements to set up custom RADIUS authentication plug-ins:

```
shared sae configuration plug-ins name name custom-radius-authentication {  
  java-class-radius-packet-handler java-class-radius-packet-handler ;  
  class-path-radius-packet-handler class-path-radius-packet-handler ;  
  require-mandatory-attributes;  
  load-balancing-mode (failover | roundRobin);  
  fallback-timer fallback-timer ;  
  timeout timeout ;  
  retry-interval retry-interval ;  
  maximum-queue-length maximum-queue-length ;  
  bind-address bind-address ;  
  udp-port udp-port ;  
  default-peer default-peer;  
}
```

To set up custom RADIUS authentication plug-ins:

1. From configuration mode, access the custom RADIUS authentication plug-in configuration. In this sample procedure, the plug-in called `customRadiusAuth` is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
customRadiusAuth custom-radius-authentication
```

2. Configure the name of the Java class that implements the `RadiusPacketHandler` interface in the RADIUS client library.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth  
 custom-radius-authentication]  
user@host# set java-class-radius-packet-handler java-class-radius-packet-handler
```

3. Configure the URLs that identify a location from which Java classes are loaded when the plug-in is initialized.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set class-path-radius-packet-handler class-path-radius-packet-handler
```

4. (Optional) Specify that a RADIUS authentication or accounting request must contain all mandatory RADIUS attributes before sending the request packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set require-mandatory-attributes
```

5. Configure the mode for load-balancing RADIUS servers.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set load-balancing-mode (failover | roundRobin)
```

6. Specify if and when the SAE attempts to fail back to the default peer.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set fallback-timer fallback-timer
```

7. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set timeout timeout
```

8. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set retry-interval retry-interval
```

9. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set maximum-queue-length maximum-queue-length
```

10. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
```

```
user@host# set bind-address bind-address
```

11. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth  
custom-radius-authentication]  
user@host# set udp-port udp-port
```

12. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth  
custom-radius-authentication]  
user@host# set default-peer default-peer
```

13. (Optional) From operational mode, verify your configuration.

```
[edit shared sae configuration plug-ins name customRadiusAuth  
custom-radius-authorization]  
user@host# show  
java-class-radius-packet-handler  
net.juniper.smgt.radius.RadiusPacketHandlerImpl;  
require-mandatory-attributes;  
load-balancing-mode failover;  
failback-timer -1;  
timeout 15000;  
retry-interval 3000;  
maximum-queue-length 10000;  
default-peer peer4;
```

Configuring LDAP Authentication Plug-Ins

Use the following configuration statements to configure LDAP authentication plug-ins:

```
shared sae configuration plug-ins name name ldap-authentication {  
  method (search | bind);  
  server server ;  
  bind-dn bind-dn ;  
  bind-password bind-password ;  
  search-filter search-filter ;  
  (ldaps);  
  search-base-dn search-base-dn ;  
  name-attribute name-attribute ;  
  password-attribute password-attribute ;  
  service-bundle-attribute service-bundle-attribute ;  
  session-volume-quota session-volume-quota ;  
  timeout timeout ;  
}
```

To create LDAP authentication plug-ins:

1. From configuration mode, access the custom LDAP authentication plug-in configuration. In this sample procedure, the plug-in called `LdapAuth` is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
LdapAuth ldap-authentication
```

2. Configure the LDAP authentication method that the SAE uses.

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]  
user@host# set method (search | bind)
```

3. (Optional) Configure a comma-separated list of IP addresses or hostnames of the LDAP authentication server.

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]  
user@host# set server server
```

4. (Optional) Configure the DN used to authenticate access to the directory.

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]  
user@host# set bind-dn bind-dn
```

5. (Optional) Configure the password that the SAE uses to authenticate its access to the directory to search for the subscriber profile. If you do not specify a bind DN or bind password, the SAE uses anonymous access.

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]  
user@host# set bind-password bind-password
```

6. (Optional) Configure the additional LDAP search filter that the SAE uses to search the directory for the subscriber profile.

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]  
user@host# set search-filter search-filter
```

7. (Optional) Enable the secure protocol used for LDAP connections with the directory. LDAPS, the only secure protocol supported, causes communication with the directory to be encrypted with Secure Sockets Layer (SSL).

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]  
user@host# set ldaps
```

8. (Optional) Configure the base DN for searching entries in the directory.

```
[edit shared sae group west-region configuration plug-ins name LdapAuth  
ldap-authentication]
```

```
user@host# set search-base-dn search-base-dn
```

9. (Optional) Configure the name of the directory attribute that holds the username.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth  
ldap-authentication]
```

```
user@host# set name-attribute name-attribute
```

10. (Optional) Configure the name of the directory attribute that stores the password.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth  
ldap-authentication]
```

```
user@host# set password-attribute password-attribute
```

11. (Optional) Configure the name of the directory attribute that contains the name of the service bundle that is used for subscriber authentication. This value is made available to the subscriber classification process and can be used to select the subscriber profile to load.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth  
ldap-authentication]
```

```
user@host# set service-bundle-attribute service-bundle-attribute
```

12. (Optional) Configure the name of the LDAP attribute that contains the value of the session volume quota. The LDAP plug-in sets the session volume quota to this value.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth  
ldap-authentication]
```

```
user@host# set session-volume-quota session-volume-quota
```

13. (Optional) Configure the maximum time the SAE waits for a response from a directory server.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth  
ldap-authentication]
```

```
user@host# set timeout timeout
```

14. (Optional) From operational mode, verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth  
ldap-authentication]  
user@host# show  
method search;  
search-filter (objectClass=umcSubscriber);  
name-attribute uniqueId;  
timeout 5000;
```