

Configuring a RADIUS Packet Template

There are two ways to define RADIUS packets for flexible RADIUS accounting and authentication plug-ins:

- Define attributes in a template, and then apply the template to flexible RADIUS accounting and authentication plug-ins.
- Define attributes in the packet definition configuration of a flexible plug-in instance. These definitions override definitions in packet templates.

Use the following configuration statements to configure a RADIUS packet template:

```
shared sae configuration radius-packet-template name ...
shared sae configuration radius-packet-template name radius-attributes name ...
shared sae configuration radius-packet-template name radius-attributes name
  attributes name {
    value ;
  }
shared sae configuration plug-ins name name flex-radius-accounting
  radius-packet-definition name ...
shared sae configuration plug-ins name name flex-radius-accounting
  radius-packet-definition name attributes name {
    value ;
  }
shared sae configuration plug-ins name name flex-radius-authentication
  radius-packet-definition name ...
shared sae configuration plug-ins name name flex-radius-authentication
  radius-packet-definition name attributes name {
    value ;
  }
}
```

To configure a template:

1. From configuration mode, access the RADIUS packet template configuration. In this sample procedure, the stdAcct template is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration
radius-packet-template stdAcct
```

2. Create an attribute instance using the names in [Unresolved xref], and enter the configuration for the RADIUS attribute instance.

```
[edit shared sae group west-region configuration radius-packet-template stdAcct]
user@host# edit radius-attributes name
```

3. Add RADIUS attribute definitions to the attribute instance. Repeat this step for each attribute.

```
[edit shared sae group west-region configuration radius-packet-template stdAcct
  radius-attributes svcstop]
user@host# set attributes name value
```

For example:

```
[edit shared sae group west-region configuration radius-packet-template stdAcct
 radius-attributes svcstop]
user@host# set attributes Acct-Session-ID sessionId
```

4. (Optional) Verify the configuration of your attribute instance.

```
[edit shared sae group west-region configuration radius-packet-template
stdAcct radius-attributes svcstop]
user@host# show
attributes {
  Acct-Input-Octets lowWord(inOctets);
  Acct-Output-Octets lowWord(outOctets);
  Acct-Input-Packets lowWord(inPackets);
  Acct-Output-Packets lowWord(outPackets);
  Acct-Input-Gigawords highWord(inOctets);
  Acct-Output-Gigawords highWord(outOctets);
}
```

5. (Optional) Verify the configuration of the RADIUS packet template.

```
[edit shared sae group west-region configuration radius-packet-template
stdAcct radius-attributes svcstop]
user@host# up
[edit shared sae group west-region configuration radius-packet-template
stdAcct]
user@host# show
radius-attributes svcstop {
  attributes {
    Acct-Input-Octets lowWord(inOctets);
    Acct-Output-Octets lowWord(outOctets);
    Acct-Input-Packets lowWord(inPackets);
    Acct-Output-Packets lowWord(outPackets);
    Acct-Input-Gigawords highWord(inOctets);
    Acct-Output-Gigawords highWord(outOctets);
  }
}
radius-attributes stop {
  attributes {
    Acct-Session-Time sessionTime;
    Acct-Terminate-Cause terminateCause;
  }
}
radius-attributes svcacct {
  attributes {
    Class radiusClass;
  }
}
radius-attributes acct {
  attributes {
    Acct-Session-Id sessionId;
    NAS-Identifier localNasId;
    NAS-IP-Address localNasIp;
    Event-Time eventTime;
  }
}
radius-attributes startstop {
```

```
attributes {  
    Acct-Multi-Session-Id ifSessionId;  
    NAS-Port-Id "\"%s %s\""%(routerName, portId or interfaceName);  
    NAS-Port "nasPort or None";  
}
```

- Related Topics**
- Using Flexible RADIUS Packet Definitions
 - Overview of Flexible RADIUS Plug-Ins
 - Defining the Values of RADIUS Attributes

Published: 2009-09-13