# Configuring Authentication, Authorization, and Accounting Functions

This chapter provides information about configuring authentication, authorization, and accounting (AAA) functions on the BNG router. BNG interacts with the RADIUS server to perform AAA functions. A group of RADIUS servers form a server group that is assigned specific AAA tasks. A method list defined on a server or server group lists methods by which authorization is performed. Some of the RADIUS features include creating specific AAA attribute formats, load balancing of RADIUS servers, throttling of RADIUS records, Change of Authorization (CoA), and Service Accounting for QoS.

***Table 1: Feature History for Configuring Authentication, Authorization, and Accounting Functions***

| Release | Modification |
|---|---|
| Release 4.2.0 | Initial release |
| Release 5.3.1 | RADIUS over IPv6 was introduced. |
| Release 5.3.2 | Service accounting support was added for line card subscribers. |
| Release 6.2.1 | A new MAC address format was introduced for RADIUS User-name Attribute. |

This chapter covers these topics:


- AAA Overview, on page 2
- Using RADIUS Server Group, on page 3
- Specifying Method List, on page 5
- Defining AAA Attributes, on page 7
- Making RADIUS Server Settings, on page 17
- Balancing Transaction Load on the RADIUS Server, on page 23
- Throttling of RADIUS Records, on page 25
- RADIUS Change of Authorization (CoA) Overview, on page 28
- User Authentication and Authorization in the Local Network, on page 37
- Service Accounting, on page 42
- Understanding Per-VRF AAA Function, on page 46
- RADIUS over IPv6, on page 46
- Additional References, on page 47

# AAA Overview

AAA acts as a framework for effective network management and security. It helps in managing network resources, enforcing policies, auditing network usage, and providing bill-related information. BNG connects to an external RADIUS server that provides the AAA functions.

The RADIUS server performs the three independent security functions (authentication, authorization, and accounting) to secure networks against unauthorized access. The RADIUS server runs the Remote Authentication Dial-In User Service (RADIUS) protocol. (For details about RADIUS protocol, refer to RFC 2865). The RADIUS server manages the AAA process by interacting with BNG, and databases and directories containing user information.

The RADIUS protocol runs on a distributed client-server system. The RADIUS client runs on BNG (Cisco ASR 9000 Series Router) that sends authentication requests to a central RADIUS server. The RADIUS server contains all user authentication and network service access information.

The AAA processes, the role of RADIUS server during these processes, and some BNG restrictions, are explained in these sections:

### Authentication

The authentication process identifies a subscriber on the network, before granting access to the network and network services. The process of authentication works on a unique set of criteria that each subscriber has for gaining access to the network. Typically, the RADIUS server performs authentication by matching the credentials (user name and password) the subscriber enters with those present in the database for that subscriber. If the credentials match, the subscriber is granted access to the network. Otherwise, the authentication process fails, and network access is denied.

### Authorization

After the authentication process, the subscriber is authorized for performing certain activity. Authorization is the process that determines what type of activities, resources, or services a subscriber is permitted to use. For example, after logging into the network, the subscriber may try to access a database, or a restricted website. The authorization process determines whether the subscriber has the authority to access these network resources.

AAA authorization works by assembling a set of attributes based on the authentication credentials provided by the subscriber. The RADIUS server compares these attributes, for a given username, with information contained in a database. The result is returned to BNG to determine the actual capabilities and restrictions that are to be applied for that subscriber.

### Accounting

The accounting keeps track of resources used by the subscriber during network access. Accounting is used for billing, trend analysis, tracking resource utilization, and capacity planning activities. During the accounting process, a log is maintained for network usage statistics. The information monitored include, but are not limited to - subscriber identities, applied configurations on the subscriber, the start and stop times of network connections, and the number of packets and bytes transferred to, and from, the network.

BNG reports subscriber activity to the RADIUS server in the form of accounting records. Each accounting record comprises of an accounting attribute value. This value is analyzed and used by the RADIUS server for network management, client billing, auditing, etc.

The accounting records of the subscriber sessions may timeout if the BNG does not receive acknowledgments from the RADIUS server. This timeout can be due to RADIUS server being unreachable or due to network connectivity issues leading to slow performance of the RADIUS server. If the sessions on the BNG are not acknowledged for their Account-Start request, loss of sessions on route processor fail over (RPFO) and other critical failures are reported. It is therefore recommended that a RADIUS server **deadtime** be configured on the BNG, to avoid loss of sessions. Once this value is configured, and if a particular session is not receiving an accounting response even after retries, then that particular RADIUS server is considered to be non-working and further requests are not sent to that server.

The **radius-server deadtime** *limit* command can be used to configure the **deadtime** for RADIUS server. For details, see Configuring RADIUS Server Settings, on page 18.

### Restrictions

- On session disconnect, transmission of the Accounting-Stop request to RADIUS may be delayed for a few seconds while the system waits for the "final" session statistics to be collected from the hardware. The Event-Timestamp attribute in that Accounting-Stop request should, however, reflect the time the client disconnects, and not the transmission time.

# Using RADIUS Server Group

A RADIUS server group is a named group of one or more RADIUS servers. Each server group is used for a particular service. For example, in an AAA network configuration having two RADIUS server groups, the first server group can be assigned the authentication and authorization task, while the second group can be assigned the accounting task.

Server groups can include multiple host entries for the same server. Each entry, however, must have a unique identifier. This unique identifier is created by combining an IP address and a UDP port number. Different ports of the server, therefore, can be separately defined as individual RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on the same server. Further, if two different host entries on the same RADIUS server are configured for the same service (like the authentication process), then the second host entry acts as a fail-over backup for the first one. That is, if the first host entry fails to provide authentication services, BNG tries with the second host entry. (The RADIUS host entries are tried in the order in which they are created.)

For assigning specific actions to the server group, see Configuring RADIUS Server Group, on page 3.

# Configuring RADIUS Server Group

Perform this task to define a named server group as the server host.

### SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *name*
3. **accounting accept** *radius_attribute_list_name*
4. **authorization reply accept** *radius_attribute_list_name*
5. **deadtime** *limit*
6. **load-balance method least-outstanding batch-size** *size* **ignore-preferred-server**
7. **server** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*

8. **source-interface** *name value*
9. **vrf** *name*
10. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **aaa group server radius** *name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa group server radius r1` | Configures the RADIUS server group named r1. |
| **Step 3** | **accounting accept** *radius_attribute_list_name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# accounting accept att_list` | Configures the radius attribute filter for the accounting process to accept only the attributes specified in the list. |
| **Step 4** | **authorization reply accept** *radius_attribute_list_name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# authorization reply accept att_list1` | Configures the radius attribute filter for the authorization process to accept only the attributes specified in the list. |
| **Step 5** | **deadtime** *limit*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# deadtime 40` | Configures the RADIUS server-group deadtime. The deadtime limit is configured in minutes. The range is from 1 to 1440, and the default is 0. |
| **Step 6** | **load-balance method least-outstanding batch-size** *size* **ignore-preferred-server**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# load-balance method least-outstanding batch-size 50 ignore-preferred-server` | Configures load balancing batch size after which the next host is picked. |
| **Step 7** | **server** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# server 1.2.3.4 acct-port 455 auth-port 567` | Specifies the radius server, and its IP address or host name. Configures the UDP port for RADIUS accounting and authentication requests. The accounting and authentication port number ranges from 0 to 65535. If no value is specified, then the default is 1645 for auth-port, and 1646 for acct-port.<br>From Cisco IOS XR Software Release 5.3.1 and later, IPv6 address can also be configured for the RADIUS server. |

| | Command or Action | Purpose |
|---|---|---|
| | | But, the host name option is supported only for IPv4 domain, and not for IPv6. |
| **Step 8** | **source-interface** *name value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# source-interface Bundle-Ether 455` | Configures the RADIUS server-group source-interface name and value for Bundle-Ether. |
| **Step 9** | **vrf** *name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# vrf vrf_1` | Configures the vrf to which the server radius group belongs. |
| **Step 10** | **commit** | |

### Configuring Radius Server-Group: An example

```
configure
aaa group server radius r1
accounting accept r1 r2
authorization reply accept a1 a2
deadtime 8
load-balance method least-outstanding batch-size 45 ignore-preferred-server
server host_name acct-port 355 auth-port 544
source-interface Bundle-Ether100.10
vrf vrf_1
!
end
```

# Specifying Method List

Method lists for AAA define the methods using which authorization is performed, and the sequence in which these methods are executed. Before any defined authentication method is performed, the method list must be applied to the configuration mechanism responsible for validating user-access credentials. The only exception to this requirement is the default method list (named "default"). The default method list is automatically applied if no other method list is defined. A defined method list overrides the default method list.

On BNG, you have to specify the method list and the server group that will be used for AAA services. For specifying method lists, see Configuring Method Lists for AAA, on page 5.

# Configuring Method Lists for AAA

Perform this task to assign the method list to be used by the server group for subscriber authentication, authorization, and accounting.

**SUMMARY STEPS**

1. **configure**
2. **aaa authentication subscriber default** *method-list-name* **group** *server-group-name*
3. **aaa authorization subscriber default** *method-list-name* **group** *server-group-name* | **radius**
4. **aaa accounting subscriber default** *method-list-name* **group** *server-group-name*
5. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **aaa authentication subscriber default** *method-list-name* **group** *server-group-name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa authentication subscriber default method1 group group1 radius group group2 group group3 ...` | Configures the method-list which will be applied by default for subscriber authentication. You can either enter 'default' or a user-defined name for the AAA method-list. Also, enter the name of the server group, on which the method list is applied. |
| **Step 3** | **aaa authorization subscriber default** *method-list-name* **group** *server-group-name* | **radius**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa authorization subscriber default method1 group group1 radius group group2 group group3 ...` | Configures the method-list which will be applied by default for subscriber authorization. You can either enter 'default' or a user-defined name for the AAA method-list. Also, enter the name of the server group, on which the method list is applied. |
| **Step 4** | **aaa accounting subscriber default** *method-list-name* **group** *server-group-name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa accounting subscriber default method1 group group1 radius group group2 group group3 ...` | Configures the method-list which will be applied by default for subscriber accounting. You can either enter 'default' or a user-defined name for the AAA method-list. Also, enter the name of the server group, on which the method list is applied. |
| **Step 5** | **commit** | |

### Configuring Method-list for AAA: An example

```
configure
aaa authentication subscriber default group radius group rad2 group rad3..
aaa authorization subscriber default group radius group rad1 group rad2 group rad3..
aaa accounting subscriber default group radius group rad1 group rad2 group rad3..
!
!
end
```

# Defining AAA Attributes

The AAA attribute is an element of RADIUS packet. A RADIUS packet transfers data between a RADIUS server and a RADIUS client. The AAA attribute parameter, and its value - form a Attribute Value Pair (AVP). The AVP carries data for both requests and responses for the AAA transaction.

The AAA attributes either can be predefined as in Internet Engineering Task Force (IETF) attributes or vendor defined as in vendor-specific attributes (VSAs). For more information about the list of BNG supported attributes, see RADIUS Attributes.

The RADIUS server provides configuration updates to BNG in the form of attributes in RADIUS messages. The configuration updates can be applied on a subscriber during session setup through two typical methods—per-user attributes, which applies configuration on a subscriber as part of the subscriber's authentication Access Accept, or through explicit domain, port, or service authorization Access Accepts. This is all controlled by the Policy Rule Engine's configuration on the subscriber.

When BNG sends an authentication or an authorization request to an external RADIUS server as an Access Request, the server sends back configuration updates to BNG as part of the Access Accept. In addition to RADIUS configuring a subscriber during setup, the server can send a change of authorization (CoA) message autonomously to the BNG during the subscriber's active session life cycle, even when the BNG did not send a request. These RADIUS CoA updates act as dynamic updates, referencing configured elements in the BNG and instructing the BNG to update a particular control policy or service policy.

BNG supports the concept of a "service", which is a group of configured features acting together to represent that service. Services can be represented as either features configured on dynamic-templates through CLI, or as features configured as RADIUS attributes inside Radius Servers. Services are activated either directly from CLI or RADIUS through configured "activate" actions on the Policy Rule Engine, or through CoA "activate-service" requests. Services can also be deactivated directly (removing all the involved features within the named service) through configured "deactivate" action on the Policy Rule Engine or through CoA "deactivate-service" requests.

The attribute values received from RADIUS interact with the subscriber session in this way:

- BNG merges the values received in the RADIUS update with the existing values that were provisioned statically by means of CLI commands, or from prior RADIUS updates.
- In all cases, values received in a RADIUS update take precedence over any corresponding CLI provisioned values or prior RADIUS updates. Even if you reconfigured the CLI provisioned values, the system does not override session attributes or features that were received in a RADIUS update.
- Changes made to CLI provision values on the dynamic template take effect immediately on all sessions using that template, assuming the template features have not already been overridden by RADIUS. Same applies to service updates made through CoA "service-update" requests.

### AAA Attribute List

An attribute list is named list that contains a set of attributes. You can configure the RADIUS server to use a particular attribute list to perform the AAA function.

To create an attribute list, see Configuring RADIUS Attribute List, on page 13.

### AAA Attribute Format

It is possible to define a customized format for some attributes. The configuration syntax for creating a new format is:

```
aaa attribute format <format-name> format-string [length] <string> *[<Identity-Attribute>]
```

where:

- format-name — Specifies the name given to the attribute format. This name is referred when the format is applied on an attribute.
- length — (Optional) Specifies the maximum length of the formatted attribute string. If the final length of the attribute string is greater than the value specified in LENGTH, it is truncated to LENGTH bytes. The maximum value allowed for LENGTH is 255. If the argument is not configured, the default is also 255.
- string — Contains regular ASCII characters that includes conversion specifiers. Only the % symbol is allowed as a conversion specifier in the STRING. The STRING value is enclosed in double quotes.
- Identity-Attribute — Identifies a session, and includes user-name, ip-address, and mac-address. A list of currently-defined identity attributes is displayed on the CLI.

Once the format is defined, the FORMAT-NAME can be applied to various AAA attributes such as username, nas-port-ID, calling-station-ID, and called-station-ID. The configurable AAA attributes that use the format capability are explained in the section Creating Attributes of Specific Format, on page 8.

To create a customized nas-port attribute and apply a predefined format to nas-port-ID attribute , see Configuring RADIUS Attribute Format, on page 14.

Specific functions can be defined for an attribute format for specific purposes. For example, if the input username is "text@abc.com", and only the portion after "@" is required as the username, a function can be defined to retain only the portion after "@" as the username. Then, "text" is dropped from the input, and the new username is "abc.com". To apply username truncation function to a named-attribute format, see Configuring AAA Attribute Format Function, on page 16.

# Creating Attributes of Specific Format

BNG supports the use of configurable AAA attributes. The configurable AAA attributes have specific user-defined formats. The following sections list some of the configurable AAA attributes used by BNG.

### Username

BNG has the ability to construct AAA username and other format-supported attributes for subscribers using MAC address, circuit-ID, remote-ID, and DHCP Option-60 (and a larger set of values available in CLI). The DHCP option-60 is one of the newer options that is communicated by the DHCP client to the DHCP server in its requests; it carries Vendor Class Identifier (VCI) of the DHCP client's hardware.

The MAC address attribute is specified in the CLI format in either of these forms:

- mac-address: for example, 0000.4096.3e4a
- mac-address-ietf: for example, 00-00-40-96-3E-4A
- mac-address-raw: for example, 000040963e4a

An example of constructing a username in the form "mac-address@vendor-class-ID" is:

```
aaa attribute format USERNAME-FORMAT format-string “%s@%s” mac-address dhcp-vendor-class
```

### NAS-Port-ID

The NAS-Port-ID is constructed by combining BNG port information and access-node information. The BNG port information consists of a string in this form:

```
"eth phy_slot/phy_subslot/phy_port:XPI.XCI"
```

For 802.1Q tunneling (QinQ), XPI is the outer VLAN tag and XCI is the inner VLAN tag.

If the interface is QinQ, the default format of nas-port-ID includes both the VLAN tags; if the interface is single tag, it includes a single VLAN tag.

In the case of a single VLAN, only the outer VLAN is configured, using this syntax:

```
<slot>/<subslot>/<port>/<outer_vlan>
```

In the case of QinQ, the VLAN is configured using this syntax:

```
<slot>/<subslot>/<port>/<inner_vlan>.<outer_vlan>
```

In the case of a bundle-interface, the phy_slot and the phy_subslot are set to zero (0); whereas the phy_port number is the bundle number. For example, 0/0/10/30 is the NAS-Port-ID for a Bundle-Ether10.41 with an outer VLAN value 30.

The nas-port-ID command is extended to use the 'nas-port-type' option so that the customized format (configured with the command shown above) can be used on a specific interface type (nas-port-type). The extended nas-port-ID command is:

```
aaa radius attribute nas-port-id format <FORMAT_NAME> [type <NAS_PORT_TYPE>]
```

If 'type' option is not specified, then the nas-port-ID for all interface types is constructed according to the format name specified in the command. An example of constructing a maximum 128 byte NAS-Port-ID, by combining the BNG port information and Circuit-ID is:

```
aaa attribute format NAS-PORT-ID-FORMAT1 format-string length 128 “eth %s/%s/%s:%s.%s %s”
physical-slot physical-subslot physical-port outer-vlan-Id inner-vlan-id circuit-id-tag
```

An example of constructing the NAS-Port-ID from just the BNG port information, and with "0/0/0/0/0/0" appended at the end for circuit-ID, is:

```
aaa attribute format NAS-PORT-ID-FORMAT2 format-string “eth %s/%s/%s:%s.%s 0/0/0/0/0/0”
physical-slot physical-subslot physical-port outer-vlan-Id inner-vlan-id
```

An example of constructing the NAS-Port-ID from just the Circuit-ID is:

```
aaa attribute format NAS-PORT-ID-FORMAT3 format-string “%s” circuit-id-tag
```

The NAS-Port-ID formats configured in the above examples, can be specified in the nas-port-ID command, thus:

```
For  IPoEoQINQ interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT1  type 41

For  Virtual IPoEoQINQ interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT2 type 44

For  IPOEoE interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT3 type 39
```

### NAS-Port-Type on Interface or VLAN Sub-interface

In order to have different production models for subscribers on the same BNG router, but different physical interfaces of same type, the NAS-Port-Type is made configurable for each physical interface, or VLAN sub-interface. With a different NAS-Port-Type value configured on the interface, the NAS-Port and NAS-Port-ID gets formatted according to the formats defined globally for the new NAS-Port-Type configured on the interface, instead of the actual value of NAS-Port-Type that the interface has. This in turn sends different formats of NAS-Port, NAS-Port-ID and NAS-Port-Type to the RADIUS server for the subscribers under different production models.

In the case of sub-interfaces, the hierarchy to be followed in deciding the format of NAS-Port-Type to be sent to the RADIUS server is:

1. Verify whether the NAS-Port-Type is configured on the sub-interface in which the subscriber session arrives.
2. If NAS-Port-Type is not configured on the sub-interface, verify whether it is configured on the main physical interface.

   The format of NAS-Port or NAS-Port-ID is based on the NAS-Port-Type retrieved in Step 1 or Step 2.
3. If NAS-Port-Type is configured on neither the sub-interface nor the main physical interface, the format of NAS-Port or NAS-Port-ID is based on the format of the default NAS-Port-Type of the sub-interface.
4. If a NAS-Port or NAS-Port-ID format is not configured for the NAS-Port-Type retrieved in steps 1, 2 or 3, the format of NAS-Port or NAS-Port-ID is based on the default formats of NAS-Port or NAS-Port-ID.

Use this command to configure NAS-Port-Type per interface or VLAN sub-interface:

```
aaa radius attribute nas-port-type <nas-port-type>
```

where:

<nas-port-type> is either a number ranging from 0 to 44, or a string specifying the nas-port-type.

Refer Configuring RADIUS Attribute Nas-port-type, on page 15.

### Calling-Station-ID and Called-Station-ID

BNG supports the use of configurable calling-station-ID and called-station-ID. The calling-station-ID is a RADIUS attribute that uses Automatic Number Identification (ANI), or similar technology. It allows the network access server (NAS) to send to the Access-Request packet, the phone number from which the call came from. The called-station-ID is a RADIUS attribute that uses Dialed Number Identification (DNIS), or similar technology. It allows the NAS to send to the Access-Request packet, the phone number that the user called from.

The command used to configure the calling-station-ID and called-station-ID attributes is:

```
aaa radius attribute calling-station-id format <FORMAT_NAME>

aaa radius attribute called-station-id format <FORMAT_NAME>
```

Examples of constructing calling-station-ID from mac-address, remote-ID, and circuit-ID are:

```
aaa radius attribute calling-station-id format CLID-FORMAT

aaa attribute format CLID-FORMAT format-string “%s:%s:%s” client-mac-address-ietf
remote-id-tag circuit-id-tag
```

Examples of constructing called-station-ID from mac-address, remote-ID, and circuit-ID are:

```
aaa radius attribute called-station-id format CLDID-FORMAT

aaa attribute format CLDID-FORMAT format-string "%s:%s" client-mac-address-raw circuit-id-tag
```

### NAS-Port Format

NAS-Port is a 4-byte value that has the physical port information of the Broadband Remote Access Server (BRAS), which connects the Access Aggregation network to BNG. It is used both by Access-Request packets and Accounting-Request packets. To uniquely identify a physical port on BRAS, multiple pieces of information such as shelf, slot, adapter, and so on is used along with the port number. A configurable format called format-e is defined to allow individual bits or group of bits in 32 bits of NAS-Port to represent or encode various pieces that constitute port information.

Individual bits in NAS-Port can be encoded with these characters:

- Zero: 0
- One: 1
- PPPoX slot: S
- PPPoX adapter: A
- PPPoX port: P
- PPPoX VLAN Id: V
- PPPoX VPI: I
- PPPoX VCI: C
- Session-Id: U
- PPPoX Inner VLAN ID: Q

```
aaa radius attribute nas-port format e [string] [type {nas-port-type}]
```

The above command is used to configure a format-e encode string for a particular interface of NAS-Port type (RADIUS attribute 61). The permissible nas-port type values are:

| Nas-port-types | Values | Whether value can be derived from associated interface | Whether value can be configured on the interface configuration mode |
|---|---|---|---|
| ASYNC | 0 | No | Yes |
| SYNC | 1 | No | Yes |
| ISDN | 2 | No | Yes |
| ISDN_V120 | 3 | No | Yes |
| ISDN_V110 | 4 | No | Yes |

| Nas-port-types | Values | Whether value can be derived from associated interface | Whether value can be configured on the interface configuration mode |
|---|---|---|---|
| VIRTUAL | 5 | No | Yes |
| ISDN_PIAFS | 6 | No | Yes |
| X75 | 9 | No | Yes |
| ETHERNET | 15 | No | Yes |
| PPPATM | 30 | No | Yes |
| PPPOEOA | 31 | No | Yes |
| PPPOEOE | 32 | Yes | Yes |
| PPPOEOVLAN | 33 | Yes | Yes |
| PPPOEOQINQ | 34 | Yes | Yes |
| VIRTUAL_PPPOEOE | 35 | Yes | Yes |
| VIRTUAL_PPPOEOVLAN | 36 | Yes | Yes |
| VIRTUAL_PPPOEOQINQ | 37 | Yes | Yes |
| IPSEC | 38 | No | Yes |
| IPOEOE | 39 | Yes | Yes |
| IPOEOVLAN | 40 | Yes | Yes |
| IPOEOQINQ | 41 | Yes | Yes |
| VIRTUAL_IPOEOE | 42 | Yes | Yes |
| VIRTUAL_IPOEOVLAN | 43 | Yes | Yes |
| VIRTUAL_IPOEOQINQ | 44 | Yes | Yes |

Examples:

```
For non-bundle: GigabitEthernet0/1/2/3.11.pppoe5

where:
PPPoEoQinQ (assuming 2 vlan tags): interface-type
1: slot
2: adapter
3: port
vlan-ids: whatever the outer and inner vlan-ids received in the PADR were
5: session-id

aaa radius attribute nas-port format e SSAAPPPPQQQQQQQQQQQVVVVVVVVVVVUUUU type 34
Generated NAS-Port:     01100011QQQQQQQQQQVVVVVVVVV0101
```

```
For bundle: Bundle-Ether17.23.pppoe8
where:
Virtual-PPPoEoQinQ (assuming 2 vlan tags): interface-type
0: slot
0: adapter
17 (bundle-id): port
Vlan-Ids: whatever the outer and inner vlan-ids received in the PADR were.
8: session-id

aaa radius attribute nas-port format e PPPPPPQQQQQQQQQQVVVVVVVVVVUUUUUU type 37
Generated NAS-Port:     010001QQQQQQQQQQVVVVVVVVV000101
```

NAS-port format for IP/DHCP sessions are represented in these examples:

```
For IPoEoVLAN interface type:
aaa radius attribute nas-port format e SSAAAPPPPPVVVVVVVVVVVVVVVVVVVVV type 40

For IPoEoQinQ:
aaa radius attribute nas-port format e SSAAAPPPPPQQQQQQQQQQQVVVVVVVVVV type 41

For virtual IPoEoVLAN:
aaa radius attribute nas-port format e PPPPPPPPVVVVVVVVVVVVVVVVUUUUUUUU type 43
```

NAS-port format for PPPoE sessions are represented in these examples:

```
For PPPoEoVLAN  interface type:
aaa radius attribute nas-port format e SSAAAPPPPPVVVVVVVVVVVVVVVVVVUUUU type 33

For Virtual PPPoEoVLAN:.
aaa radius attribute nas-port format e PPPPPPPPVVVVVVVVVVVVVVVVUUUUUUUU type 36
```

**Note** If a NAS-Port format is not configured for a NAS-Port-Type, the system looks for a default CLI configuration for the NAS-Port format. In the absence of both these configurations, for sessions with that particular NAS-Port-Type, the NAS-Port attribute is not sent to the RADIUS server.

# Configuring RADIUS Attribute List

Perform this task to create a RADIUS attribute list that is used for filtering authorization and accounting attributes.

### SUMMARY STEPS

1. **configure**
2. **radius-server attribute list** *listname*
3. **attribute** *list_of_radius_attributes*
4. **attribute vendor-id** *vendor-type number*
5. **vendor-type** *vendor-type-value*
6. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **radius-server attribute list** *listname*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server attribute list l1` | Defines the name of the attribute list. |
| **Step 3** | **attribute** *list_of_radius_attributes*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-attribute-filter)# attribute a1, a2` | Populates the list with radius attributes.<br>**Note** For more information about supported attributes, see RADIUS Attributes . |
| **Step 4** | **attribute vendor-id** *vendor-type number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# attribute vendor-id 6456` | Configures the attribute filtering to be applied to vendor specific attributes (VSAs) by allowing vendor specific information for VSAs to be specified in radius attribute list CLI. Vendor specific information comprises of vendor-id, vendor-type, and optional attribute name in case of Cisco generic VSA. The vendor-id ranges from 0 to 4294967295. |
| **Step 5** | **vendor-type** *vendor-type-value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-attribute-filter-vsa)# vendor-type 54` | Configures the vendor specific information such as the vendor-type to be specified in radius attribute list. The range of the vendor-type value is from 1 to 254. |
| **Step 6** | **commit** | |

### Configuring RADIUS Attribute List: An example

```
configure
radius-server attribute list list_! attribute B C
attribute vendor-id vendor-type 10
vendor-type 30
!
end
```

# Configuring RADIUS Attribute Format

Perform this task to the define RADIUS attribute format for the nas-port attribute, and apply a predefined format on nas-port-ID attribute.

**SUMMARY STEPS**

1. **configure**
2. **aaa radius attribute**

3. **nas-port format e** *string* **type** *nas-port-type value*
4. **nas-port-id format** *format name*
5. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **aaa radius attribute**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa radius attribute` | Configures the AAA radius attribute. |
| **Step 3** | **nas-port format e** *string* **type** *nas-port-type value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# nas-port format e format1 type 30` | Configures the format for nas-port attribute. The string represents a 32 character string representing the format to be used. The nas-port-value ranges from 0 to 44. |
| **Step 4** | **nas-port-id format** *format name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# nas-port-id format format2` | Applies a predefined format to the nas-port-ID attribute. |
| **Step 5** | **commit** | |

### Configuring RADIUS Attribute Format: An example

```
configure
aaa radius attribute
nas-port format e abcd type 40
nas-port-id format ADEF
!
end
```

# Configuring RADIUS Attribute Nas-port-type

Perform this task to configure RADIUS Attribute nas-port-type on a physical interface or VLAN sub-interface:

**SUMMARY STEPS**

1. **configure**
2. **interface** *type interface-name*
3. **aaa radius attribute nas-port-type** {*value* | *name*}
4. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface gigabitEthernet 0/0/0/0` | Enters the interface configuration mode. |
| **Step 3** | **aaa radius attribute nas-port-type** {*value* \| *name*}<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# aaa radius attribute nas-port-type 30`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# aaa radius attribute nas-port-type Ethernet` | Configures the RADIUS Attribute nas-port-type value.<br>The range of *value* is from 0 to 44.<br>See table in NAS-Port Format, on page 11, for permissible nas-port-type values within this range. |
| **Step 4** | **commit** | |

### Configuring RADIUS Attribute Nas-port-type: An example

```
configure
interface gigabitEthernet 0/0/0/0
 aaa radius attribute nas-port-type Ethernet
!
end
```

# Configuring AAA Attribute Format Function

Perform this task to configure a function for the AAA attribute format. The function is for stripping the user-name till the delimiter.

**SUMMARY STEPS**

1. **configure**
2. **aaa attribute format** *format-name*
3. **username-strip prefix-delimiter** *prefix_delimiter*
4. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **aaa attribute format** *format-name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa attribute format red` | Specifies the format name for which the function is defined. |
| **Step 3** | **username-strip prefix-delimiter** *prefix_delimiter*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-id-format)# username-strip prefix-delimiter @` | Configures the function to strip the username preceding the prefix delimiter, which is @. |
| **Step 4** | **commit** | |

### Configuring AAA Attribute Format Function: An example

```
configure
aaa attribute format red
username-strip prefix-delimiter @
!
!
end
```

# Making RADIUS Server Settings

In order to make BNG interact with the RADIUS server, certain server specific settings must be made on the BNG router. This table lists some of the key settings:

| Settings | Description |
|---|---|
| Server host | Defines the RADIUS server details to which BNG will connect. |
| Attribute list | Defines which attribute list is to be used. |
| Server key | Defines the encryption status. |
| Dead criteria | Defines the criteria that is used to mark a RADIUS server as dead. |
| Retransmit value | Defines the number of retries the BNG makes to send data to RADIUS server. |
| Timeout value | Defines how long BNG waits for the RADIUS server to reply. |
| Automated testing | Defines the duration after which automated testing will start and the username to be tested. |
| IP DSCP | Allows RADIUS packets to be marked with a specific Differentiated Services Code Point (DSCP) value. |

For more making RADIUS server settings, see Configuring RADIUS Server Settings, on page 18.

For more making specific automated testing settings, see Configuring Automated Testing, on page 21.

For more making specific IP DSCP settings, see Setting IP DSCP for RADIUS Server, on page 22.

### Restriction

The service profile push or asynchronously pushing a profile to the system is not supported. To download a profile from Radius, the profile must be requested initially as part of the subscriber request. Only service-update is supported and can be used to change a service that was previously downloaded.

# Configuring RADIUS Server Settings

Perform this task to make RADIUS server specific settings on the BNG router.

### SUMMARY STEPS

1. **configure**
2. **radius-server host** *ip-address* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*
3. **radius-server attribute list** *list_name attribute_list*
4. **radius-server key** *7 encrypted_text*
5. **radius-server disallow null-username**
6. **radius-server dead-criteria time** *value*
7. **radius-server dead-criteria tries** *value*
8. **radius-server deadtime** *limit*
9. **radius-server ipv4 dscp** *codepoint_value*
10. **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*
11. **radius-server retransmit** *retransmit_value*
12. **radius-server source-port extended**
13. **radius-server timeout** *value*
14. **radius-server vsa attribute ignore unknown**
15. **radius source-interface Loopback** *value* **vrf** *vrf_name*
16. **commit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **radius-server host** *ip-address* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server host 1.2.3.4 acct-port 455 auth-port 567` | Specifies the radius server and its IP address. Configures the UDP port for RADIUS accounting and authentication requests. The accounting and authentication port numbers range from 0 to 65535. If no value is specified, then the default is 1645 for the auth-port and 1646 for the acct-port.<br>From Cisco IOS XR Software Release 5.3.1 and later, IPv6 address can also be configured for the RADIUS server host. |
| **Step 3** | **radius-server attribute list** *list_name attribute_list*<br>**Example:** | Specifies the radius server attributes list, and customizes the selected radius attributes. |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config)# radius-server attribute list rad_list a b` | |
| **Step 4** | **radius-server key** *7 encrypted_text*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-radius-host)# radius-server key 7 rngiry` | Specifies the per-server encryption key that overrides the default, and takes the value 0 or 7, which indicates that the unencrypted key will follow. |
| **Step 5** | **radius-server disallow null-username**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server disallow null-username` | Specifies that the null-username is disallowed for the radius server. |
| **Step 6** | **radius-server dead-criteria time** *value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria time 40` | Specifies the dead server detection criteria for a configured RADIUS server. The time (in seconds) specifies the minimum time that must elapse since a response is received from this RADIUS server. |
| **Step 7** | **radius-server dead-criteria tries** *value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria tries 50` | Specify the value for the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. The value ranges from 1 to 100. |
| **Step 8** | **radius-server deadtime** *limit*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server deadtime 67` | Specifies the time in minutes for which a RADIUS server is marked dead. The deadtime limit is specified in minutes and ranges from 1 to 1440. If no value is specified, the default is 0. |
| **Step 9** | **radius-server ipv4 dscp** *codepoint_value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp 45` | Allows radius packets to be marked with a specific differentiated services code point (DSCP) value. This code point value ranges from 0 to 63. |
| **Step 10** | **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding ignore-preferred-server batch-size 500` | Configures the radius load-balancing options by picking the server with the least outstanding transactions. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25. |
| **Step 11** | **radius-server retransmit** *retransmit_value*<br>**Example:** | Specifies the number of retries to the active server. The retransmit value indicates the number of retries in numeric |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config)# radius-server retransmit 45` | and ranges from 1 to 100. If no value is specified, then the default is 3. |
| **Step 12** | **radius-server source-port extended**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server source-port extended` | Configures BNG to use a total of 200 ports as the source ports for sending out RADIUS requests. |
| **Step 13** | **radius-server timeout** *value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server timeout` | Specifies the time to wait for a radius server to reply. The value is in seconds and ranges from 1 to 1000. The default is 5. |
| **Step 14** | **radius-server vsa attribute ignore unknown**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server vsa attribute ignore unknown` | Ignores the unknown vendor-specific attributes for the radius server. |
| **Step 15** | **radius source-interface Loopback** *value* **vrf** *vrf_name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius source-interface Loopback 655 vrf vrf_1` | Specifies loopback interface for source address in RADIUS packets. The value ranges from 0 to 65535. |
| **Step 16** | **commit** | |

### Configuring RADIUS Server Settings: Examples

```
\\Configuring RADIUS Server Options
configure
radius-server attribute list list1 a b
radius-server dead-criteria time 100
radius-server deadtime 30
radius-server disallow null-username
radius-server host 1.2.3.4 acct-port 655 auth-port 566
radius-server ipv4 dscp 34
radius-server key 7 ERITY$
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 25
radius-server retransmit 50
radius-server source-port extended
radius-server timeout 500
radius-server vsa attribute ignore unknown
!
!
end

\\Configuring RADIUS Attribute List
radius-server attribute list list_! attribute B C
attribute vendor-id vendor-type 10
vendor-type 30
```

```
!
end

\\Configuring RADIUS Server Host
configure
radius-server host 1.3.5.7 acct-port 56 auth-port 66
idle-time 45
ignore-acct-port
ignore-auth-port 3.4.5.6
key 7 ERWQ
retransmit 50
test username username
timeout 500
!
end

\\Configuring RADIUS Server Key
configure
radius-server key 7  ERWQ
!
end

\\Configuring Load Balancing for RADIUS Server
configure
radius-server load-balance method least-outstanding batch-size 25
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 45
!
end

\\Ignoring Unknown VSA Attributes in RADIUS Server
configure
radius-server vsa attribute ignore unknown
!
end

\\Configuring Dead Criteria for RADIUS Server
configure
radius-server dead-criteria time 60
radius-server dead-criteria tries 60
!
end

\\Configuring Disallow Username
configure
radius-server disallow null-username
!
end

\\Setting IP DSCP for RADIUS Server
configure
radius-server ipv4 dscp 43
radius-server ipv4 dscp default
!
end
```

# Configuring Automated Testing

Perform this task to test if the external RADIUS server is UP or not.

**SUMMARY STEPS**

1. **configure**
2. **radius-server idle-time** *idle_time*
3. **radius-server test username** *username*
4. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **radius-server idle-time** *idle_time*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-radius-host)# radius-server idle-time 45` | Specifies the idle-time after which the automated test should start. The idle time is specified in minutes, and ranges from 1 to 60. |
| **Step 3** | **radius-server test username** *username*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-radius-host)# radius-server test username user1` | Specifies the username to be tested for the automated testing functionality. |
| **Step 4** | **commit** | |

**Configuring Automated Testing: An example**

```
configure
radius-server idle-time 60
radius-server test username user_1
!
end
```

# Setting IP DSCP for RADIUS Server

Perform this task to set IP differentiated services code point (DSCP) for RADIUS server.

**SUMMARY STEPS**

1. **configure**
2. **radius-server ipv4 dscp** *codepoint_value*
3. **radius-server ipv4 dscp default**
4. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **radius-server ipv4 dscp** *codepoint_value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp 45` | Allows radius packets to be marked with a specific differentiated services code point (DSCP) value that replaces the outdated IP precedence, a 3-bit field in the Type of Service byte of the IP header originally used to classify and prioritize types of traffic. This code point value ranges from 0 to 63. |
| **Step 3** | **radius-server ipv4 dscp default**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp default` | Matches the packets with default dscp (000000). |
| **Step 4** | **commit** | |

**Setting IP DSCP for RADIUS Server: An example**

```
configure
radius-server ipv4 dscp 43
radius-server ipv4 dscp default
!
end
```

# Balancing Transaction Load on the RADIUS Server

The RADIUS load-balancing feature is a mechanism to share the load of RADIUS access and accounting transactions, across a set of RADIUS servers. Each AAA request processing is considered to be a transaction. BNG distributes batches of transactions to servers within a server group.

When the first transaction for a new is received, BNG determines the server with the lowest number of outstanding transactions in its queue. This server is assigned that batch of transactions. BNG keeps repeating this determination process to ensure that the server with the least-outstanding transactions always gets a new batch. This method is known as the least-outstanding method of load balancing.

You can configure the load balancing feature either globally, or for RADIUS servers that are part of a server group. In the server group, if a preferred server is defined, you need to include the keyword "ignore-preferred-server" in the load-balancing configuration, to disable the preference.

For configuring the load balancing feature globally, see Configuring Load Balancing for Global RADIUS Server Group, on page 24.

For configuring the load balancing feature on RADIUS servers that are part of a named server group, see Configuring Load Balancing for a Named RADIUS Server Group, on page 24.

# Configuring Load Balancing for Global RADIUS Server Group

Perform this task to activate the load balancing function for the global RADIUS server group. As an example, in this configuration the preferred server is set to be ignored.

### SUMMARY STEPS

1. **configure**
2. **radius-server load-balance method least-outstanding batch-size** *size*
3. **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*
4. **commit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **radius-server load-balance method least-outstanding batch-size** *size*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding batch-size 500` | Configures the radius load-balancing options by picking the server with the least-outstanding transactions. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25. |
| **Step 3** | **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding ignore-preferred-server batch-size 500` | Configures the radius load-balancing options by disabling the preferred server for this Server Group. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25. |
| **Step 4** | **commit** | |

#### Configuring Load Balancing for RADIUS Server: An example

```
configure
radius-server load-balance method least-outstanding batch-size 25
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 45
!
end
```

# Configuring Load Balancing for a Named RADIUS Server Group

Perform this task to activate the load balancing function for a named RADIUS server group. As an example, in this configuration the preferred server is set to be ignored.

**SUMMARY STEPS**

1. **configure**
2. **aaa group server radius** *server_group_name* **load-balance method least-outstanding batch-size** *size*
3. **aaa group server radius** *server_group_name* **load-balance method least-outstanding ignore-preferred-server batch-size** *size*
4. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **aaa group server radius** *server_group_name* **load-balance method least-outstanding batch-size** *size*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa group server radius sg1 load-balance method least-outstanding batch-size 500` | Configures the radius load-balancing options by picking the server with the least-outstanding transactions. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25. |
| **Step 3** | **aaa group server radius** *server_group_name* **load-balance method least-outstanding ignore-preferred-server batch-size** *size*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa group server radius sg1 load-balance method least-outstanding ignore-preferred-server batch-size 500` | Configures the radius load-balancing options by disabling the preferred server for this Server Group. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25. |
| **Step 4** | **commit** | |

# Throttling of RADIUS Records

The Throttling of AAA (RADIUS) records is a mechanism to avoid RADIUS congestion and instability. This function is useful in situations when there is insufficient bandwidth to accommodate a sudden burst of AAA requests generated by the BNG for the RADIUS server.

While configuring throttling, a threshold rate, which corresponds to the maximum number of outstanding requests, is defined. It is possible to configure independent throttling rates for access (authentication and authorization) and accounting requests. After a threshold value is reached for a server, no further requests of that type are sent to the server. However, for the pending requests, a retransmit timer is started, and if the outstanding request count (which is checked after every timer expiry), is less than the threshold, then the request is sent out.

As a session may timeout due to throttle on the access requests, a limit is set for the number of retransmit attempts. After this limit is reached, further access requests are dropped. Throttled accounting requests, however, are processed through the server-group failover process.

The throttling feature can be configured globally, or for a server-group. However, the general rule of configuration preference is that the server-group configuration overrides global configuration, if any.

The syntax for the throttling CLI command is:

```
radius-server throttle {[accounting THRESHOLD] [access THRESHOLD [access-timeout
NUMBER_OF-TIMEOUTS]]}
```

where:

- accounting THRESHOLD—Specifies the threshold for accounting requests. The range is from 0 to 65536. The default is 0, and indicates that throttling is disabled for accounting requests.
- access THRESHOLD—Specifies the threshold for access requests. The range is from 0 to 65536. The default is 0, and indicates that throttling is disabled for accounting requests.
- access-timeout NUMBER_OF-TIMEOUTS—Specifies the number of consecutive timeouts that must occur on the router, after which access-requests are dropped. The range of is from 0 to 10. The default is 3.

**Note** By default, the throttling feature is disabled on BNG.

For activating throttling globally, see Configuring RADIUS Throttling Globally, on page 26.

For activating throttling on a server group, see Configuring RADIUS Throttling on a Server Group, on page 27.

# Configuring RADIUS Throttling Globally

Perform this task to activate RADIUS throttling globally.

### SUMMARY STEPS

1. **configure**
2. **radius-server throttle access** *threshold_value*
3. **radius-server throttle access** *threshold_value* **access-timeout** *value*
4. **radius-server throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*
5. **radius-server throttle accounting** *threshold_value* **access** *value* **access-timeout** *value*
6. **commit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **radius-server throttle access** *threshold_value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10` | Controls the number of access requests sent to a RADIUS server. The threshold value denotes the number of outstanding access requests after which throttling should be performed. The range is from 0 to 65535, and the preferred value is 100. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **radius-server throttle access** *threshold_value* **access-timeout** *value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10 access-timeout 5` | Specifies the number of timeouts, after which a throttled access request is dropped. The value denotes the number of timeouts for a transaction. The range is from 1 to 10, and the default is 3. |
| **Step 4** | **radius-server throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10 access-timeout 5 accounting 10` | Controls the number of access timeout requests sent to a RADIUS server. The threshold value denotes the number of outstanding accounting transactions after which throttling should be performed. The range is from 0 to 65535, and the preferred value is 100. |
| **Step 5** | **radius-server throttle accounting** *threshold_value* **access** *value* **access-timeout** *value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# radius-server throttle accounting 56 access 10 access-timeout 5` | Controls the number of accounting requests sent to a RADIUS server. The threshold value denotes the number of outstanding accounting transactions after which throttling should be performed. The value ranges between 0 to 65535 and the preferred value is 100. |
| **Step 6** | **commit** | |

### Configuring RADIUS Throttling Globally: An example

```
configure
radius-server throttle access 10 access-timeout 5 accounting 10
!
end
```

# Configuring RADIUS Throttling on a Server Group

Perform this task to activate RADIUS throttling on a server group.

### SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *server_group_name*
3. **server** *hostname* **acct-port** *acct_port_value* **auth-port** *auth_port_value*
4. **throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*
5. **commit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **aaa group server radius** *server_group_name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aaa group server radius SG1` | Configures the AAA (RADIUS) server-group definition. |
| **Step 3** | **server** *hostname* **acct-port** *acct_port_value* **auth-port** *auth_port_value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# server 99.1.1.10 auth-port 1812 acct-port 1813` | Configures a RADIUS server accounting or authentication port with either the IP address or hostname (as specified). The accounting port number and the authentication port number ranges from 0 to 65535. |
| **Step 4** | **throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-sg-radius)# radius-server throttle access 10 access-timeout 5 accounting 10` | Configures the RADIUS throttling options to control the number of access and accounting requests sent to a RADIUS server. The threshold value denotes the number of outstanding access requests or accounting transactions after which throttling should be performed. The range is from 0 to 65535, and for both access and accounting requests the preferred value is 100. |
| **Step 5** | **commit** | |

#### Configuring RADIUS Throttling on a Server Group: An example

```
configure
aaa group server radius SG1
server 99.1.1.10 auth-port 1812 acct-port 1813
radius-server throttle access 10 access-timeout 5 accounting 10
!
end
```

# RADIUS Change of Authorization (CoA) Overview

The Change of Authorization (CoA) function allows the RADIUS server to change the authorization settings for a subscriber who is already authorized. CoA is an extension to the RADIUS standard that allows sending asynchronous messages from RADIUS servers to a RADIUS client, like BNG.

**Note** A CoA server can be a different from the RADIUS server.

To identify the subscriber whose configuration needs to be changed, a RADIUS CoA server supports and uses a variety of keys (RADIUS attributes) such as Accounting-Session-ID, Username, IP-Address, and ipv4:vrf-id.

The RADIUS CoA supports:

- account-logon — When a user logs into a network, an external web portal that supports CoA sends an account-logon request to BNG with the user's credentials (username and password). Account-logon on BNG then attempts to authenticate the user through RADIUS with those credentials.
- account-logoff— BNG processes the account-logoff request as a disconnect event for the subscriber and terminates the session.

**Note** The RADIUS CoA server does not differentiate between originators of the disconnect event. Hence, when the BNG receives an account-logoff request from the RADIUS CoA server, for both a user-initiated and an administrator-initiated request, the Acct-Terminate-Cause to be sent to the RADIUS server is always set as Admin-Reset.

- account-update — BNG parses and applies the attributes received as part of the CoA profile. Only subscriber-specific attributes are supported and applied on the user profile.
- activate-service — BNG starts a predefined service on a subscriber. The service settings can either be defined locally by a dynamic template, or downloaded from the RADIUS server.
- deactivate-service — BNG stops a previously started service on the subscriber, which is equivalent to deactivating a dynamic-template.

For a list of supported Vendor-Specific Attributes for account operations, see Vendor-Specific Attributes for Account Operations.

**Note** In order for BNG to enable interim accounting, it is mandatory for the CoA request to have both accounting method list from the dynamic-template and Acct-Interim-Interval attribute from the user profile. This behavior is applicable for accounting enabled through dynamic-template. Whereas, from Cisco IOS XR Software Release 5.3.0 and later, the CoA request needs to have only the Acct-Interim-Interval attribute in the user profile.

### Service Activate from CoA

BNG supports activating services through CoA requests. The CoA **service-activate** command is used for activating services. The CoA request for the service activate should contain these attributes:

- "subscriber:command=activate-service" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- Other attributes that are part of the service profile

The "<subscriber:sa=<service-name>" can also be used to activate services from CoA and through RADIUS.

Duplicate service activate requests can be sent to BNG from the CoA server. BNG does not take any action on services that are already activated. BNG sends a CoA ACK message to the CoA server under these scenarios:

- When a duplicate request with identical parameters comes from the CoA for a service that is already active.
- When a duplicate request with identical parameters comes from the CoA to apply a parameterized service.

BNG sends a CoA NACK message to the CoA server with an error code as an invalid attribute under these scenarios:

- When a request comes from the CoA to deactivate a non-parameterized service that is not applied to the session.
- When a request comes from the CoA to deactivate a parameterized service that is not applied to the session.
- When a duplicate request to apply a parameterized service is made with non-identical parameters from the CoA.
- When a request with non-identical parameters comes from CoA to deactivate a parameterized service.

### Service Update from CoA

The service update feature allows an existing service-profile to be updated with a new RADIUS attribute list representing the updated service. This impacts any subscriber who is already activated with the service and new subscriber who activate the service in the future. The new CoA **service-update** command is used for activating this feature. The CoA request for the service update should have these attributes:

- "subscriber:command=service-update" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- Other attributes that are part of the service profile

A service update CoA should have a minimum of these attributes:

- vsa cisco generic 1 string "subscriber:command=service-update"
- vsa cisco generic 1 string "subscriber:service-name=<service name>"

### Web Logon with RADIUS Based CoA

To support Web Logon, a set of Policy Rule Events need to be configured in an ordered manner. These events are as follows:

- session-start:
  - On the start of a session, a subscriber is setup to get internet connectivity. The service is activated to redirect HTTP traffic to a Web portal for web-based logon.
  - Start the timer with duration for the maximum waiting period for authentication.
- account-logon — The Web portal collects the user credentials such as username and password and triggers a CoA account-logon command. When this event is triggered, subscriber username and password are authenticated by the RADIUS server. Once the authentication is successful, the HTTP redirect service is deactivated, granting user access to already connected internet setup. Also, the timer established in session-start must be stopped. However, if the authentication fails during account-logon, BNG sends a NAK CoA request, allowing for further authentication attempts to take place.
- timer expiry — When the timer expires, the subscriber session is disconnected based on the configuration.

# Multi-Action Change of Authorization

BNG supports multi-action Change of Authorization (CoA) wherein service providers can activate and deactivate multiple services using a single CoA request. Multi-action CoA is supported for **Service-Logon** and **Service-Logoff** CoA commands. The Service-Logon command can contain one or more **Service-Activate** attributes, and optionally **Service-Deactivate** attributes, for multi-action CoA to specify service(s) to be activated or deactivated. Similarly, the **Service-Logoff** command can contain one or more **Service-Deactivate** attributes, and optionally **Service-Activate** attributes, for multi-action CoA to specify service(s) to be deactivated or activated.

MA-CoA supports up to a maximum of 10 service activations or deactivations per MA-CoA request, however, it is recommended to issue six activations or deactivations per MA-CoA request.

During the multi-action CoA request, if any of the COA requests fail to activate or deactivate, then any of the services which have been activated or deactivated as part of that CoA request is rolled back to its previous state. The session restores back to the its pre-MA-CoA state upon failure to activation or deactivation.

A rollback-failure event, exception, can be configured to specify what action to be taken when a service rollback fails following a failed MA-CoA request (that is, a case of a double-failure condition). The default action to be taken when the rollback fails is to preserve the session, however, you can configure to terminate the session.

The following example details on the rollback failure exception.

```
policy-map type control subscriber PL1
 event session-start match-first
  class type control subscriber class-default do-all
   1 activate dynamic-template pkt-trig1
  !
 !
 event exception match-first
  class type control subscriber coa-rollback-failure do-all
   10 disconnect
  !
 !
!
```

### An Example of a Multi-Action Change of Authorization Use Case

The following example lists the sequence of events that occur in the case of a PTA session initiation.

1. PTA session's web traffic redirected to a service portal (HTTP Redirect)
2. The user activates the first level of service through the service portal. A multi-action COA request is initiated in the following sequence.
    1. Deactivate redirection
    2. Activate Turbo Button 1
    3. Activate VoIP with two channels
3. The user activates the second level of service through the service portal. A multi-action COA request is initiated in the following sequence.
    1. Deactivate Turbo Button 1
    2. Activate Turbo Button 2

3. Deactivate VoIP with two channels
4. Activate VoIP with 4 channels

#### Interworking with Service-Level Accounting

BNG supports Service-Level Accounting, where a service is a collection of features that are activated and deactivated as a group. Service-Level Accounting and MA-CoA features are independent, that is, they can be applied separately. However, MA-CoA accounts for services that are activated or deactivated that have Service-Level Accounting enabled through the dynamic template configuration.

## Generating Accounting Records

The following cases describes how the multi-action CoA records are generated for accounting purposes.

**MA-CoA ACK Case**

- If MA-CoA request contains only service activate commands, then START accounting record for those services are generated after the CoA Ack is sent out.
- If MA-CoA request contains only deactivate services or combination of activate and deactivate services, then for those services START or STOP accounting records are generated after the CoA Ack is sent out.

**MA-CoA NAK Case (Rollback scenario)**

- If MA-CoA request fails due to presence of invalid command formats or due to internal software failure or due to presence of invalid service names, that are not defined in the box, in such cases the accounting START or STOP messages are not generated upon rollback.
- If MA-CoA request fails due to internal feature programming failure, then the Service-START or Service-STOP accounting records may be generated for the services that were activated or deactivated before the failure. After the failure, the rollback is initiated and appropriate Service-START or Service-STOP records are generated for these services.

## High Availability for MA-CoA

If an high availability event other than a line card online insertion and removal (LC-OIR), such as a process restart or an RP failover occurs while an MA-CoA request is being processed, then the affected session is restored to its pre-MA-CoA state. The policy plane does not make an attempt to automatically recover the MA-CoA message or to resume processing. Instead, the CoA Client times out and re-sends the MA-CoA request to the BNG router.

## An Example with Verification Commands

The following example shows the profile of a subscriber with existing services, modified with a MA-CoA request, and the subscriber profile with the changed services invoked by the MA-CoA request.

#### Multi-Action Change of Authorization - Verification Commands

**Session with an Existing Service** ----------[1]

```
show subsscriber session all detail internal
```

```
Interface:                Bundle-Ether1.1.ip1
Circuit ID:               Unknown
Remote ID:                Unknown
Type:                     IP: DHCP-trigger
IPv4 State:               Up, Wed Jul  9 14:25:40 2014
IPv4 Address:             12.1.0.2, VRF: default
IPv4 Up helpers:          0x00000040 {IPSUB}
IPv4 Up requestors:       0x00000040 {IPSUB}
Mac Address:              0000.0c00.0001
Account-Session Id:       00000001
Nas-Port:                 Unknown
User name:                0000.0c00.0001
Outer VLAN ID:            10
Subscriber Label:         0x00000040
Created:                  Wed Jul  9 14:25:37 2014
State:                    Activated
Authentication:           unauthenticated
Authorization:            authorized
Ifhandle:                 0x020001a0
Session History ID:       1
Access-interface:         Bundle-Ether1.1
Policy Executed:

  event Session-Start match-first [at Wed Jul  9 14:25:37 2014]
    class type control subscriber ISN_CM do-all [Succeeded]
      1 activate dynamic-template ISN_TEMPLATE_1 [cerr: No error][aaa: Success]
      2 authorize aaa list default [cerr: No error][aaa: Success]
      1001 activate dynamic-template svcQoSAcct2 [cerr: No error][aaa: Success]
      1002 activate dynamic-template svcQoSAcct3 [cerr: No error][aaa: Success]
Session Accounting: disabled
Last COA request received: unavailable
User Profile received from AAA:
 Attribute List: 0x1000eb24
 1:  ipv4-mtu        len=  4  value= 1500(5dc)
Services:
  Name        : ISN_TEMPLATE_1
  Service-ID  : 0x4000002
  Type        : Template
  Status      : Applied
-------------------------
  Name        : svcQoSAcct1
  Service-ID  : 0x400000a
  Type        : Multi Template
  Status      : Applied
-------------------------
  Name        : svcQoSAcct2
  Service-ID  : 0x400000b
  Type        : Template
  Status      : Applied
-------------------------
  Name        : svcQoSAcct3
  Service-ID  : 0x400000c
  Type        : Template
  Status      : Applied
-------------------------
[Event History]
   Jul  9 14:29:41.056 IPv4 Start
   Jul  9 14:29:44.384 SUBDB produce done
   Jul  9 14:29:44.384 IPv4 Up

RP/0/RSP1/CPU0:BNG#show subscriber database association
```

```
Location 0/RSP1/CPU0

Bundle-Ether1.1.ip1, subscriber label 0x40
  Name                           Template Type
  --------                       -------------
  U00000040                      User profile
  svcQoSAcct3                    Service
  svcQoSAcct2                    Service
  svcQoSAcct1                    Service
  ISN_TEMPLATE_1                 IP subscriber
```

**MA-CoA Request Initiated From RADIUS Client** ----------
[2]

```
exec  /bin/echo
"Cisco-AVPair='subscriber:sd=svcQoSAcct1',Cisco-AVPair='subscriber:sd=svcQoSAcct2',
Cisco-AVPair='subscriber:sd=svcQoSAcct3',Cisco-AVPair='subscriber:sa=qosin_coa',
Cisco-AVPair='subscriber:sa=qosout_coa',Acct-Session-Id=00000001" | /usr/local/bin/radclient
 -r 1 -x 5.11.17.31:1700 coa coa


RP/0/RSP1/CPU0:BNG#show subscriber manager statistics AAA COA location 0/rsp1/cpu0


[ CHANGE OF AUTHORIZATION STATISTICS ]

CoA Requests:
                 Type     Received        Acked        NAKed
                 ====     ========        =====        =====
        Account Logon            0            0            0
       Account Logoff            0            0            0
       Account Update            0            0            0
           Disconnect            0            0            0
 Single Service Logon            0            0            0
Single Service Logoff            0            0            0
Single Service Modify            0            0            0
     Multiple Service            1            1            0

Errors:
  None

RP/0/RSP1/CPU0:BNG#show subscriber session all detail internal

Interface:               Bundle-Ether1.1.ip1
Circuit ID:              Unknown
Remote ID:               Unknown
Type:                    IP: DHCP-trigger
IPv4 State:              Up, Wed Jul  9 14:25:40 2014
IPv4 Address:            12.1.0.2, VRF: default
IPv4 Up helpers:         0x00000040 {IPSUB}
IPv4 Up requestors:      0x00000040 {IPSUB}
Mac Address:             0000.0c00.0001
Account-Session Id:      00000001
Nas-Port:                Unknown
User name:               0000.0c00.0001
Outer VLAN ID:           10
Subscriber Label:        0x00000040
Created:                 Wed Jul  9 14:25:37 2014
State:                   Activated
Authentication:          unauthenticated
Authorization:           authorized
```

```
Ifhandle:                0x020001a0
Session History ID:      1
Access-interface:        Bundle-Ether1.1
Policy Executed:

  event Session-Start match-first [at Wed Jul  9 14:25:37 2014]
    class type control subscriber ISN_CM do-all [Succeeded]
      1 activate dynamic-template ISN_TEMPLATE_1 [cerr: No error][aaa: Success]
      2 authorize aaa list default [cerr: No error][aaa: Success]
      1001 activate dynamic-template svcQoSAcct2 [cerr: No error][aaa: Success]
      1002 activate dynamic-template svcQoSAcct3 [cerr: No error][aaa: Success]
Session Accounting: disabled
Last COA request: Wed Jul  9 14:27:37 2014
COA Request  Attribute List: 0x1000f0c4
 1:  sd              len= 11  value= svcQoSAcct1
 2:  command         len= 18  value= deactivate-service
 3:  service-info    len= 11  value= svcQoSAcct1
 4:  service-name    len= 11  value= svcQoSAcct1
 5:  sd              len= 11  value= svcQoSAcct2
 6:  command         len= 18  value= deactivate-service
 7:  service-info    len= 11  value= svcQoSAcct2
 8:  service-name    len= 11  value= svcQoSAcct2
 9:  sd              len= 11  value= svcQoSAcct3
 10:  command         len= 18  value= deactivate-service
 11:  service-info    len= 11  value= svcQoSAcct3
 12:  service-name    len= 11  value= svcQoSAcct3
 13:  sa              len=  9  value= qosin_coa
 14:  command         len= 16  value= activate-service
 15:  service-info    len=  9  value= qosin_coa
 16:  service-name    len=  9  value= qosin_coa
 17:  sa              len= 10  value= qosout_coa
 18:  command         len= 16  value= activate-service
 19:  service-info    len= 10  value= qosout_coa
 20:  service-name    len= 10  value= qosout_coa
Last COA response: Result ACK
COA Response  Attribute List: 0x1000f4e4
 1:  sd              len= 11  value= svcQoSAcct1
 2:  sd              len= 11  value= svcQoSAcct2
 3:  sd              len= 11  value= svcQoSAcct3
 4:  sa              len=  9  value= qosin_coa
 5:  sa              len= 10  value= qosout_coa
User Profile received from AAA:
 Attribute List: 0x1000f6f4
 1:  ipv4-mtu        len=  4  value= 1500(5dc)
Services:
  Name        : ISN_TEMPLATE_1
  Service-ID  : 0x4000002
  Type        : Template
  Status      : Applied
-------------------------
  Name        : qosin_coa
  Service-ID  : 0x4000006
  Type        : Multi Template
  Status      : Applied
-------------------------
  Name        : qosout_coa
  Service-ID  : 0x4000008
  Type        : Multi Template
  Status      : Applied
-------------------------
[Event History]
   Jul  9 14:29:41.056 IPv4 Start
   Jul  9 14:29:44.384 IPv4 Up
```

```
   Jul  9 14:31:41.504 CoA request
   Jul  9 14:31:41.632 SUBDB produce done [many]
```

**Changed Subscriber Profile after the MA-CoA Request is Processed from RADIUS**
----------[3]

```
RP/0/RSP1/CPU0:BNG#show subscriber database association


Location 0/RSP1/CPU0

Bundle-Ether1.1.ip1, subscriber label 0x40
  Name                            Template Type
  --------                        -------------
  U00000040                       User profile
  qosout_coa                      Service
  qosin_coa                       Service
  ISN_TEMPLATE_1                  IP subscriber
```

In the above example, the subscriber profile existing services are defined by [1], the changes initiated by the MA-CoA request is represented by [2], and the changes that are impacted by the MA-CoA request is shown in [3].

## Restrictions in Multi-Action Change of Authorization

Multi-Action Change of Authorization is subjected to the following restrictions:

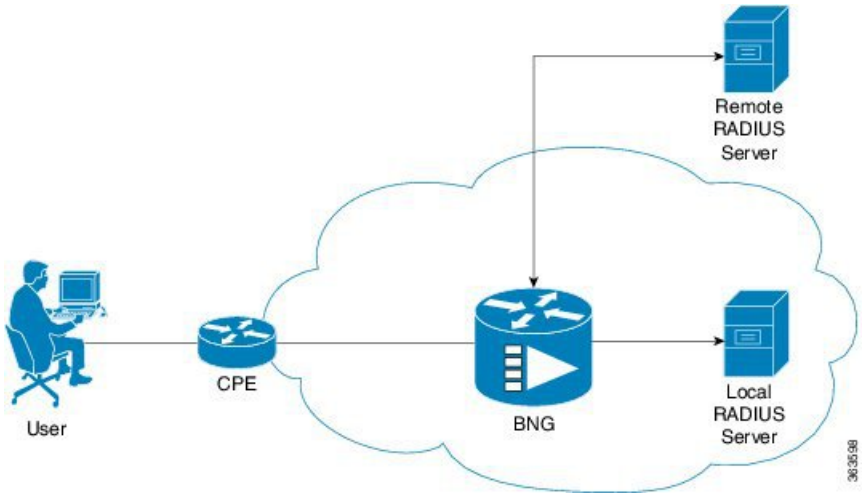
- **Service-Activate and Service-Deactivate commands only**: Only the Service-Activate and Service-Deactivate commands are supported in the MA-CoA requests. If a MA-CoA request containing account-logon, account-logoff, account-update, session-query, or disconnect-request commands is received, the request is rejected.
- Cisco VSAs of format **"subscriber:command= activate-service"** and **"subscriber:service-name=Svc1"** are not supported in MA-COA. If requests containing these VSA formats are received, a NAK is sent. Only formats of the **"subscriber:sa/sd=svcname"** type is supported.
- Event **service-logon** and **service-logoff** actions are not supported under policy map for services activated or deactivated through MA-CoA (same as service activation done as part of Access-Accept).
- MA-CoA with QoS Shaper Parameterization is not supported.
- **Dynamic Template services only**: MA-CoA is supported on services that are defined through dynamic templates configured on the router. MA-CoA design does not preclude support for services downloaded through RADIUS server, that is, the service profiles.
- **CoA Account-Update messages must not contain any Service-Activate or Deactivate VSAs**: MA-CoA does not restrict or detect Service-Activate or Service-Deactivate VSAs within the CoA Account-Update messages, however, the support is not available.
- **Bundle Subscribers only**: MA-COA is currently supported on RP-based subscribers (bundle-access interfaces) only.
- **Scale, Performance, Boundary Conditions**: The following are the conditions for MA-CoA:
  - MA-CoA does not to impose any significant limitations on scaling, in terms of the total number of sessions or the number of services applied per session.

- MA-CoA supports up to a maximum of 10 service activations or deactivations per MA-CoA request. If the number of action requests exceeds the limit of 10, a NAK is initiated for the last request received.
- MA-CoA can handle a maximum of 30 CoA messages per second.

# User Authentication and Authorization in the Local Network

The user authentication and authorization in the local network feature in BNG provides the option to perform subscriber authorization locally (in a subscriber's network), instead of both remote authentication and authorization that occurs in RADIUS servers. With the User Authentication and Authorization in the Local Network feature, you can run the RADIUS server locally in your network, manage, and configure the RADIUS server locally in your network to the profile that is required for the environment. In the case of a remote RADIUS server, the RADIUS server is maintained by an external regulatory body (not within the subscriber's network) and subscriber will not be able to manage or configure the server.

***Figure 1: User Authentication and Authorization in the Local Network***




User Authentication and Authorization in the Local Network feature is used in a case when a user wants to perform a two-level authentication or authorization, first, a remote authentication (or authorization) followed by a local authorization (or authentication).

**Note** All the debug commands applicable to AAA server are applicable on User Authentication and Authorization in the Local Network feature.

For IPoE subscribers, User Authentication and Authorization in the Local Network is a two-level authorization process as a part of the session-start event. For PTA subscribers, User Authentication and Authorization in the Local Network is a remote server authentication process, followed by a local server authorization process.

# Policy Configurations for IPoE Sessions

The following policy configuration explains how the authentication and authorization process occurs in IPoE subscriber sessions. The authentication and authorization processes are performed using two RADIUS servers (one located remotely and the other located locally). At first, the authentication request is routed to the remotely located RADIUS server, which is not in the user's control. Then, to authorize the session, the authorization request is routed to the local RADIUS server, where the subscriber profile for the service provider is maintained.

As a first step in the authorization process, you can configure the authentication process to download the authorization profile from the local RADIUS server. However, when both RADIUS servers have the same authorization profiles, either partially or completely, that part of the authorization profile that is the same is overridden by the one downloaded from the local RADIUS server, and the other part of the authorization profile is merged.

Case 1: Subscriber session created by applying the user profile downloaded from the local RADIUS server.

```
Radius Server1 (located remotely, profile not controlled by the operator

              0000.0000.0001  Cleartext-Password := "shootme"
             Fall-Through = no

Radius Server2 (located locally, profile controlled by the operator)

              0000.0000.0001  Cleartext-Password := "shootme"
             Class = "IPSUB",
             Cisco-avpair += "ip:sub-qos-policy-in=12MUp",
             Cisco-avpair += "ip:sub-qos-policy-out=12MDown",
             Fall-Through = no
```

Case 2: Subscriber session created by applying the user profile downloaded from the remote RADIUS server, and in this case, the policy attribute values are overridden by the local RADIUS server profile.

```
Radius Server1 (located remotely, profile not controlled by the operator)

             0000.0000.0001  Cleartext-Password := "shootme"
             Cisco-avpair += "ip:sub-qos-policy-in=6MUp",
             Cisco-avpair += "ip:sub-qos-policy-out=6MDown",
             Fall-Through = no

Radius Server2 (located locally, profile controlled by the operator)

              0000.0000.0001  Cleartext-Password := "shootme"
             Class = "IPSUB",
             Cisco-avpair += "ip:sub-qos-policy-in=12MUp",
             Cisco-avpair += "ip:sub-qos-policy-out=12MDown",
             Fall-Through = no
```

### Profile Created by the Attribute Merging of both the Local and Remote Server Profiles

```
RP/0/RSP0/CPU0:BNG#sh run aaa
radius-server host 10.105.236.46 auth-port 1812 acct-port 1813
key 7 111B1801464058
!
radius-server host 10.105.236.237 auth-port 1812 acct-port 1813
key 7 095E4F0D485744
```

```
!
aaa group server radius local_server
server 10.105.236.237 auth-port 1812 acct-port 1813
!
aaa group server radius remote_server
server 10.105.236.46 auth-port 1812 acct-port 1813
!
aaa accounting subscriber acct_meth broadcast group local_server group remote_server
aaa authorization subscriber local_server group local_server
aaa authorization subscriber remote_server group remote_server

RP/0/RSP0/CPU0:BNG#

RP/0/RSP0/CPU0:BNG#sh run policy-map type control subscriber ISN_CNTRL_1
policy-map type control subscriber ISN_CNTRL_1
event session-start match-all
  class type control subscriber ISN_CM do-all
   10 activate dynamic-template ISN_TEMPLATE_1
   11 authorize aaa list remote_server identifier source-address-mac password shootme
   12 authorize aaa list local_server identifier source-address-mac password shootme
  !
!
end-policy-map
!

RP/0/RSP0/CPU0:BNG#
Remote User Profile
0000.0c00.0001 Cleartext-Password := "shootme"
    cisco-avpair += "subscriber:accounting-list=acct_meth", -- [(A) Same attribute on both
 profile]
    Session-Timeout += 1000, ---------------------------------- [(B) Attribute defined in
remote profile only]
    Acct-Interim-Interval = 3600 ----------------------------- [(C) Same attribute on both
 profiles with diff value}


Local User profile
0000.0c00.0001 Cleartext-Password := "shootme"
    cisco-avpair += "subscriber:accounting-list=acct_meth", -- [(A) Same attribute on both
 profile]
    cisco-avpair += "sub-qos-policy-in=12MUp",---------------- [(D) Attribute defined in
local profile only]
    cisco-avpair += "sub-qos-policy-out=12MDown",------------- [(E) Attribute defined in
local profile only]
    cisco-avpair += "ipv4:inacl=innet", ----------------------- [(F) Attribute defined in
local profile only]
    cisco-avpair += "ipv4:outacl=outnet", --------------------- [(G) Attribute defined in
local profile only]
   Acct-Interim-Interval = 3000 ----------------------------- [(H) Same attributes on both
 profiles with diff value]


RP/0/RSP0/CPU0:BNG#sh subscriber session all detail internal
Interface:               Bundle-Ether1.1.ip22
Circuit ID:              Unknown
Remote ID:               Unknown
Type:                    IP: DHCP-trigger
IPv4 State:              Up, Wed Jun 18 16:56:25 2014
IPv4 Address:            12.16.0.24, VRF: default
IPv4 Up helpers:         0x00000040 {IPSUB}
IPv4 Up requestors:      0x00000040 {IPSUB}
Mac Address:             0000.0c00.0001
Account-Session Id:      000000bb
Nas-Port:                Unknown
```

```
User name:                0000.0c00.0001
Outer VLAN ID:            10
Subscriber Label:         0x00000075
Created:                  Wed Jun 18 16:56:15 2014
State:                    Activated
Authentication:           unauthenticated
Authorization:            authorized
Ifhandle:                 0x000012a0
Session History ID:       11
Access-interface:         Bundle-Ether1.1
Policy Executed:


  event Session-Start match-all [at Wed Jun 18 16:56:15 2014]
    class type control subscriber ISN_CM do-all [Succeeded]
      10 activate dynamic-template ISN_TEMPLATE_1 [cerr: No error][aaa: Success]
      11 authorize aaa list remote_server [cerr: No error][aaa: Success]
      12 authorize aaa list local_server [cerr: No error][aaa: Success]
Session Accounting:
  Acct-Session-Id:          000000bb
  Method-list:              acct_meth
  Accounting started:       Wed Jun 18 16:56:25 2014
  Interim accounting:       On, interval 50 mins
    Last successful update: Never
    Next update in:         00:46:48 (dhms)
    Last update sent:       Never
    Updates sent:           0
    Updates accepted:       0
    Updates rejected:       0
    Update send failures:   0
Last COA request received: unavailable
User Profile received from AAA:
Attribute List: 0x1000e764
1:  session-timeout len=  4  value= 1000(3e8)  ------- [(B) Attribute value fetched from
the remote profile]
2:  accounting-list len=  9  value= acct_meth  ------- [(A) Attribute common to both the
profiles]
3:  sub-qos-policy-in len=  5  value= 12MUp    ------- [(D) Attribute defined in the local
 profile]
4:  sub-qos-policy-out len=  7  value= 12MDown ------- [(E) Attribute defined in the local
 profile]
5:  inacl           len=  5  value= innet ------------ [(F) Attribute defined in the local
 profile]
6:  outacl          len=  6  value= outnet ----------- [(G) Attribute defined in the local
 profile]
7:  acct-interval   len=  4  value= 3000(bb8) -------- [(I) Attribute value fetched from
the local profile]
Services:
  Name        : ISN_TEMPLATE_1
  Service-ID  : 0x4000002
  Type        : Template
  Status      : Applied
-------------------------
```

In the above example, the server profile attributes are defined in both the Local RADIUS and the Remote RADIUS servers. Attributes (A), (B), and (C) are defined in remote RADIUS server profile, and attributes (A), (D), (E), (F), (G), and (H) are defined in the local RADIUS server profile. The subscriber session created by applying the user profile downloaded from the local RADIUS server contains attributes (B), (A), (D), (E), (F), (G), and (I), where the attribute (B) is fetched from the remote RADIUS server profile; the attribute (A) is common to both the RADIUS server profiles; the attributes (D), (E), (F), and (G) are the attributes fetched from the local RADIUS server profile; and

attribute (I) is common to both the profiles, however, the attribute value differs on both the profiles. In this case, the value of the attribute (I) is fetched from the local RADIUS server profile.

# Policy Configurations for PTA Sessions

The following policy configuration explains how the authentication and authorization processes occur in PTA subscriber sessions. In the case of PTA subscriber sessions, the authentication and authorization processes consists of two steps:

1. Domain Authorization on LAC: Achieved through the local RADIUS server where the domain authorization occurs.

```
policy-map type control subscriber vpdn_ipv4_pmap
event session-start match-first
 class type control subscriber vpdn_ipv4_cmap do-until-failure
  ! activate dynamic-template vpdn_v4
 !
event session-activate match-first
 class type control subscriber vpdn_ipv4_cmap do-until-failure
  10 authorize aaa list vpdn-author-list format vpdn_domain password cisco
  20 authenticate aaa list vpdn-authen-list
 !
```

2. User Authentication before forwarding on LAC: Achieved using two RADIUS servers: one local RADIUS server for domain authorization and another remote RADIUS server for user authentication.

```
radius-server vsa attribute ignore unknown

radius-server host 5.8.23.156 auth-port 1812 acct-port 1813
key 7 02050D480809
!
radius-server host 5.8.23.160 auth-port 1812 acct-port 1813
key 7 030752180500
!
radius-server key 7 0214055F5A545C
aaa attribute format vpdn_domain
username-strip prefix-delimiter @
!
aaa accounting network default start-stop group radius
aaa group server radius vpdn-authen
server 5.8.23.160 auth-port 1812 acct-port 1813
!
aaa group server radius vpdn-author
server 5.8.23.156 auth-port 1812 acct-port 1813
!
aaa accounting subscriber default group radius
aaa authorization subscriber vpdn-author-list group vpdn-author
aaa authentication subscriber vpdn-authen-list group vpdn-authen
!
```

# Service Accounting

Accounting records for each service enabled on a subscriber can be sent to the configured RADIUS server. These records can include service-start, service-stop, and service-interim records containing the current state of the service and any associated counters. This feature is the Service Accounting feature. Service accounting records are consolidated accounting records that represent the collection of features that make up a service as part of a subscriber session.

Service accounting starts when a subscriber session comes up with a service enabled on it. This can happen through a dynamic template applied through a control policy, through access-accept (AA) messages when the session is authorized, or through a change of authorization (CoA), when a new service is applied on a subscriber session. Service accounting stops either when the session is terminated, or a service is removed from the session through CoA, or some other event that deactivates the service. Start records have no counters; interim and stop records with QoS counters are generated when service accounting is enabled for QoS. Interim accounting records can be generated, in between start and stop accounting, as an option with a pre-defined periodic interval. When the interim period is zero, interim accounting records are not created. Different interim intervals are based on every service for each session. Service accounting is enabled on each template, based on the configuration.

Service Accounting is supported on bundle subscriber interfaces as well as line card subscriber interfaces.

**Note** The policy-map associated to a dynamic template can be edited to change the service parameters. However, this does not update the accounting records. Therefore, to generate all the accounting records accurately, it is recommended that a new service with all the required service parameters be created and associated to the new service, through a CoA.

For service accounting, statistics for ingress and egress QoS policies, which are applied under each service for a given subscriber, may need to be reported as part of the accounting interim and stop records. For each service, these QoS counters can be reported as part of the accounting records:

- BytesIn — Aggregate of bytes matching all classes of the ingress QoS policy for the service minus the policer drops.
- PacketsIn — Aggregate of packets matching all classes of the ingress QoS policy for the service minus the policer drops.
- BytesOut — Aggregate of bytes matching all classes of the egress QoS policy for the service minus the queuing drops.
- PacketsOut — Aggregate of packets matching all classes of the egress QoS policy for the service minus the queuing drops

Dynamic template features that support accounting statistic collection and require that their statistics be reported in the AAA service accounting records can enable accounting statistics on their features using the newly-introduced optional **acct-stats** configuration option. This option is not available for the features that do not support statistic collection. By default, QoS accounting statistics are disabled to optimize performance.

**Note** The QoS counters for each direction is reported only if a QoS policy is applied for that service in the given direction. For example, if a service does not have an ingress policy applied, BytesIn and PacketsIn counters are reported as being 0.

### Pre-requisites

- Subscriber accounting, the parent accounting record for service accounting, must be configured to enable the service accounting feature to work.
- The keyword **acct-stats** must be configured in service-policy configuration to enable the service accounting feature to report feature counter information as part of the records.

### Restriction

- IPv4 and IPv6 subscriber sessions has a single set of service accounting records. They are merged into one set of bytes_in, bytes_out, packets_in, packets_out counters.
- Service accounting is not supported for static sessions.

# Configuring Service Accounting

Perform this task to configure service accounting through the dynamic template:

### Before you begin

You must configure subscriber accounting before performing this task. Refer Creating Dynamic Template for IPv4 or IPv6 Subscriber Session for configuring procedure.

## SUMMARY STEPS

1. **configure**
2. **aaa accounting service** {*list_name* | **default**} {**broadcast group** {*group_name* | **radius**} | **group** {*group_name* | **radius**} }
3. **aaa service-accounting** [**extended** | **brief**]
4. **dynamic-template**
5. **type service** *dynamic-template-name*
6. **accounting aaa list** {*method_list_name* | **default**} **type service** [**periodic-interval** *time*]
7. {**ipv4** | **ipv6**} **access-group** *access-list-name*
8. **service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**]
9. **commit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **aaa accounting service** {*list_name* \| **default**} {**broadcast group** {*group_name* \| **radius**} \|**group** {*group_name* \| **radius**} } **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting service l1 group srGroup1` | Creates an accounting list for service accounting |
| **Step 3** | **aaa service-accounting** [**extended** \| **brief**] **Example:** `RP/0/RSP0/CPU0:router(config)# aaa service-accounting brief` | (Optional) Sets accounting parameters for service to select the level of subscriber accounting state and to identity attribute reporting in brief or extended form. **Note** The default setting is extended. |
| **Step 4** | **dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template` | Enters the dynamic-template configuration mode. |
| **Step 5** | **type service** *dynamic-template-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service s1` | Creates a dynamic-template with a user-defined name for a service. |
| **Step 6** | **accounting aaa list** {*method_list_name* \| **default**} **type service** [**periodic-interval** *time*] **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# accounting aaa list l1 type service periodic-interval 1000` | Configures the service accounting feature. |
| **Step 7** | {**ipv4** \| **ipv6**} **access-group** *access-list-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 access-group ACL1` `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 access-group ACL2` | Sets IPv4 or IPv6 access list to an interface. |
| **Step 8** | **service-policy** {**input** \| **output** \| **type**} *service-policy_name* [**acct-stats**] **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input QoS1 acct-stats` `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy output QoS2 acct-stats` | Associates a service-policy to the dynamic template, and enables service accounting feature using **acct-stats** keyword. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | **commit** | |

### Configuring Service Accounting: Example

```
configure
 aaa accounting service S1 group SG1
 aaa service-accounting brief
 dynamic-template
  type service s1
   accounting aaa list S1 type service periodic-interval 600
   ipv4 access-group ACL1
   service-policy input QOS1 acct-stats
   service-policy output QOS2 acct-stats
!
!
end
```

# Statistics Infrastructure

The accounting counters are maintained by the service accounting statistics IDs (statsD) infrastructure. Service accounting interacts with the statistics infrastructure in this manner:

- Each feature has a statistics collector process that is responsible for returning statistics counters for that feature.
- A single collector can handle counters for multiple features.
- An accounting process, the service accounting management agent, uses the access library to register for notifications and request statistics, and pushes to a radius server.

There is a polling period to pull the data from statsD. To support sub-second accuracy on stop records, the statistics are immediately pulled when the session is terminated, without waiting for any polling method to get accurate data. The same method is followed by session accounting and service accounting. Sub-second accuracy is not supported for data reported in interim records, because no data is pulled while sending interim accounting records.

# Configuring Statistics IDs (statsD)

The statsD is configured to poll feature statistics by default every 900 seconds (that is, every 15 minutes). Perform this task to change the default figure to either increase or decrease the polling interval.

### SUMMARY STEPS

1. **configure**
2. **statistics period service-accounting** {*period* | **disable**}
3. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure** | |
| **Step 2** | **statistics period service-accounting** {*period* \| **disable**}<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# statistics period service-accounting 1800` | Sets collection period for statistics collectors for the service accounting feature. |
| **Step 3** | **commit** | |

**Configuring Service Accounting: Example**

```
configure
 statistics period service-accounting 1800
end
```

# Understanding Per-VRF AAA Function

The Per VRF AAA function allows authentication, authorization, and accounting (AAA) on the basis of virtual routing and forwarding (VRF) instances. This feature permits the Provider Edge (PE) or Virtual Home Gateway (VHG) to communicate directly with the customer's RADIUS server, (which is associated with the customer's Virtual Private Network (VPN)), without having to go through a RADIUS proxy.

ISPs must be able to define operational parameters such as AAA server groups, method lists, system accounting, and protocol-specific parameters, and associate those parameters to a particular VRF instance.

The Per VRF AAA feature is supported with VRF extensions to server-group, RADIUS, and system accounting commands. The list of servers in server groups is extended to include definitions of private servers, in addition to references to the hosts in the global configuration. This allows simultaneous access to both customer servers and global service provider servers. The syntax for the command used to configure per-vrf AAA globally is:

```
radius source-interface subinterface-name [vrf vrf-name]
```

## RADIUS Double-Dip Feature

BNG supports the RADIUS double-dip feature, where BNG sends the first authentication or authorization request to a service provider's RADIUS server, which in turn responds with the correct VRF associated with the subscriber session. Subsequently, the BNG redirects the original request, and sends it as a second request, to the correct RADIUS server that is associated with the designated VRF.

# RADIUS over IPv6

From Cisco IOS XR Software Release 5.3.1 and later, RADIUS over IPv6 is supported in BNG, thereby allowing IPv6 address also for various RADIUS configurations and CoA client configurations.

These commands are extended to support IPv6 address:

- **radius-server host** (global configuration mode)
- **radius server** (radius server group configuration mode)
- **radius server-private** (radius server group configuration mode)
- **aaa server radius dynamic-author client** (global configuration mode)

For details on configuring RADIUS server group and settings, see Configuring RADIUS Server Group, on page 3 and Configuring RADIUS Server Settings, on page 18.

# Additional References

These sections provide references related to implementing RADIUS.

### RFCs

| Standard/RFC - AAA | |
|---|---|
| RFC-2865 | Remote Authentication Dial In User Service (RADIUS) |
| RFC-2866 | RADIUS Accounting |
| RFC-2867 | RADIUS Accounting Modifications for Tunnel Protocol Support |
| RFC-2868 | RADIUS Attributes for Tunnel Protocol Support |
| RFC-2869 | RADIUS Extensions |
| RFC-3575 | IANA Considerations for RADIUS |
| RFC-4679 | DSL Forum Vendor-Specific RADIUS Attributes |
| RFC-5176 | Dynamic Authorization Extensions to RADIUS |

### MIBs

| MIBs | MIBs Link |
|---|---|
| — | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. | http://www.cisco.com/cisco/web/support/index.html |