Authentication Authorization and Accounting Configuration Guide Cisco IOS Release 12.4

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Configuring Authentication

Authentication provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the selected security protocol, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Authentication

The Cisco IOS software implementation of authentication is divided into AAA Authentication and non-authentication methods. Cisco recommends that, whenever possible, AAA security services be used to implement authentication.

Restrictions for Configuring Authentication

- Effective with Cisco IOS Release 12.3, the number of AAA method lists that can be configured is 250.
- If you configure one RADIUS server with the nonstandard option and another RADIUS server without the nonstandard option, the RADIUS-server host with the nonstandard option does not accept a

predefined host. If you configure the same RADIUS server host IP address for a different UDP destination port for accounting requests using the **acct-port** keyword and a UDP destination port for authentication requests using the **auth-port**keyword with and without the nonstandard option, the RADIUS server does not accept the nonstandard option.

Information About Configuring Authentication

The following sections describe how AAA authentication is configured by defining a named list of authentication methods and then applying that list to various interfaces, and how AAA authentication is handled through RADIUS Change in Authorization (CoA):

- Named Method Lists for Authentication, page 2
- RADIUS Change of Authorization, page 5

Named Method Lists for Authentication

A named list of authentication methods must first be defined to configure AAA authentication, and then this named list is applied to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which is named “default”). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is a sequential list describing the authentication methods to be queried in order to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS software uses the first listed method to authenticate users. If that method fails to respond, the Cisco IOS software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or all methods defined in the method list are exhausted.

It is important to note that the Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle--meaning that the security server or local username database responds by denying the user access--the authentication process stops and no other authentication methods are attempted.

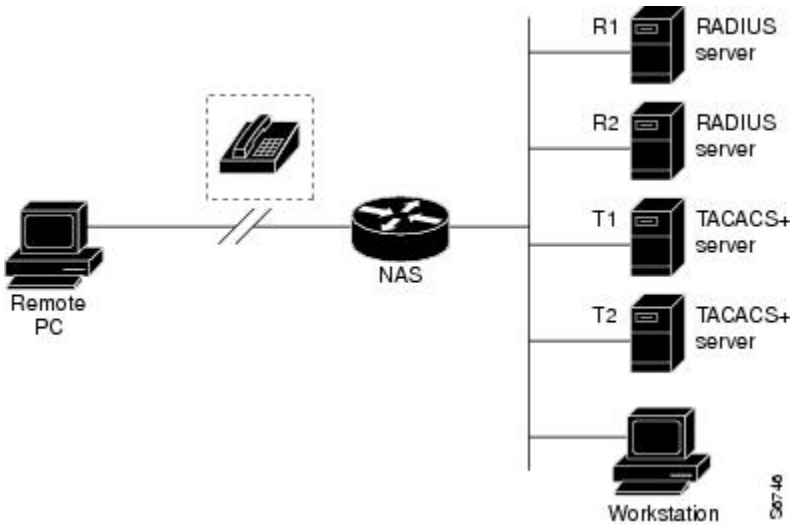
- Method Lists and Server Groups, page 2
- Method List Examples, page 3
- AAA Authentication General Configuration Procedure, page 4

Method Lists and Server Groups

A server group is a way to group existing Lightweight Directory Access Protocol (LDAP), RADIUS or TACACS+ server hosts for use in method lists. The figure below shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers and T1 and T2 are

TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Figure 1 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as a server group, and define T1 and T2 as a separate server group. For example, you can specify R1 and T1 in the method list for authentication login, while specifying R2 and T2 in the method list for PPP authentication.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service--for example, authentication--the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

See the Configuring LDAP, Configuring RADIUS, or Configuring TACACS+ feature modules for more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers.

Method List Examples

Suppose the system administrator has decided on a security solution where all interfaces will use the same authentication methods to authenticate PPP connections. In the RADIUS group, R1 is contacted first for authentication information, then if there is no response, R2 is contacted. If R2 does not respond, T1 in the TACACS+ group is contacted; if T1 does not respond, T2 is contacted. If all designated servers fail to respond, authentication falls to the local username database on the access server itself. To implement this solution, the system administrator would create a default method list by entering the following command:

```
aaa authentication ppp default group radius group tacacs+ local
```

In this example, “default” is the name of the method list. The protocols included in this method list are listed after the name, in the order they are to be queried. The default list is automatically applied to all interfaces.

When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern would continue through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated.

It is important to remember that a FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Suppose the system administrator wants to apply a method list only to a particular interface or set of interfaces. In this case, the system administrator creates a named method list and then applies this named list to the applicable interfaces. The following example shows how the system administrator can implement an authentication method that will be applied only to interface 3:

```
aaa authentication ppp default group radius group tacacs+ local
aaa authentication ppp apple group radius group tacacs+ local none
 interface async 3
 ppp authentication chap apple
```

In this example, “apple” is the name of the method list, and the protocols included in this method list are listed after the name in the order in which they are to be performed. After the method list has been created, it is applied to the appropriate interface. Note that the method list name (apple) in both the AAAand PPP authentication commands must match.

In the following example, the system administrator uses server groups to specify that only R2 and T2 are valid servers for PPP authentication. To do this, the administrator must define specific server groups whose members are R2 (172.16.2.7) and T2 (172.16.2.77), respectively. In this example, the RADIUS server group “rad2only” is defined as follows using the **aaa group server** command:

```
aaa group server radius rad2only
 server 172.16.2.7
```

The TACACS+ server group “tac2only” is defined as follows using the **aaa group server** command:

```
aaa group server tacacs+ tac2only
 server 172.16.2.77
```

The administrator then applies PPP authentication using the server groups. In this example, the default methods list for PPP authentication follows the order: **group rad2only**, **group tac2only**, and **local**:

```
aaa authentication ppp default group rad2only group tac2only local
```

AAA Authentication General Configuration Procedure

To configure AAA authentication, perform the following tasks:

1 Enable AAA by using the **aaa new-model** command in global configuration mode.

2 Configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos if you are using a security server. See Configuring RADIUS, Configuring TACACS+, and Configuring Kerberos, respectively for more information.
3 Define the method lists for authentication by using an AAA authentication command.
4 Apply the method lists to a particular interface or line, if required.

RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model in which the request originates from a network attached device and the response is sent from the queried servers. The Cisco IOS supports the RADIUS Change of Authorization (CoA) extensions defined in RFC 5176 that are typically used in a pushed model and allow for the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

Beginning with Cisco IOS Release 12.2(5) SXI, per-session CoA requests are supported in:

- Session reauthentication
- Session termination
- Session termination with port shutdown
- Session termination with port bounce
- Security and Password--see the Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices feature module for more information.
- Accounting--see the Configuring Accounting feature module for more information..

This section describes how RADIUS CoA messaging works:

- Change-of-Authorization Requests, page 5
- CoA Request Response Code, page 6
- CoA Request Commands, page 8

Change-of-Authorization Requests

Change of Authorization (CoA) requests, as described in RFC 5176, are used in a push model to allow for session identification, host reauthentication, and session termination. The model is comprised of one request (CoA-Request) and two possible response codes:

- CoA acknowledgement (ACK) [CoA-ACK]
- CoA non-acknowledgement (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the router that acts as a listener.

- RFC 5176 Compliance, page 5

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the router for session termination.

The table below shows the IETF attributes that are supported for this feature.

Table 1 Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

The table below shows the possible values for the Error-Cause attribute.

Table 2 Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

CoA Request Response Code

The CoA Request response code can be used to issue a command to the router. The supported commands are listed in the CoA Request Commands section.

- Session Identification, page 7
- CoA ACK Response Code, page 7
- CoA NAK Response Code, page 7

Session Identification

For disconnect and CoA requests targeted at a particular session, the router locates the session based on one or more of the following attributes:

- Calling-Station-Id (IETF attribute #31which contains the host MAC address)
- Audit-Session-Id (Cisco VSA)
- Acct-Session-Id (IETF attribute #44)

Unless all session identification attributes included in the CoA message match the session, the router returns a Disconnect-NAK or CoA-NAK with the “Invalid Attribute Value” error-code attribute.

For disconnect and CoA requests targeted to a particular session, any one of the following session identifiers can be used:

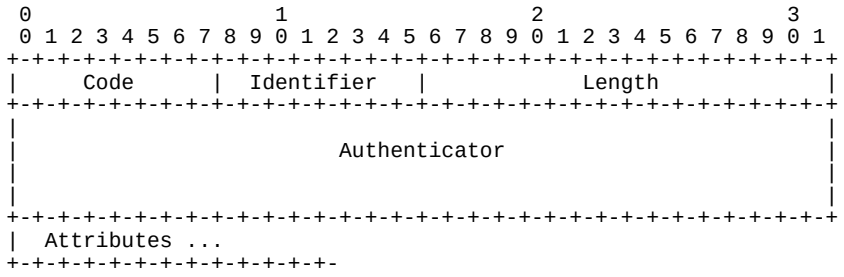
- Calling-Station-ID (IETF attribute #31, which contains the MAC address)
- Audit-Session-ID (Cisco vendor-specific attribute)
- Accounting-Session-ID (IETF attribute #44).

If more than one session identification attribute is included in the message, all of the attributes must match the session or the router returns a Disconnect- negative acknowledgement (NAK) or CoA-NAK with the error code “Invalid Attribute Value.”

CoA ACK Response Code

If the authorization state is changed successfully, a positive acknowledgement (ACK) is sent. The attributes returned within CoA ACK vary based on the CoA Request and are discussed in individual CoA Commands.

The packet format for a CoA Request code as defined in RFC 5176 consists of the fields: Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format.



The attributes field is used to carry Cisco VSAs.

CoA NAK Response Code

A negative acknowledgement (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure.

CoA Request Commands

The router supports the commands shown in the table below.

Table 3 CoA Commands Supported on the Router

Command[1]	Cisco VSA
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	This is a standard disconnect request that does not require a VSA
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"

Session Reauthentication

To initiate session authentication, the AAA server sends a standard CoA-Request message that contains a Cisco vendor-specific attribute (VSA) in this form: *Cisco:Avpair="subscriber:command=reauthenticate"* and one or more session identification attributes.

The current session state determines the router response to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1x, the router responds by sending an EAPoL[2]-RequestId message (see footnote below) to the server.
- If the session is currently authenticated by MAC authentication bypass (MAB), the router sends an access-request to the server, passing the same identity attributes used for the initial successful authentication.
- If session authentication is in progress when the router receives the command, the router terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request command terminates the session without disabling the host port. This command causes re-initialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network. If the session cannot be located, the router returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session *is* located, the

1 **All CoA commands must include the session identifier between the router and the CoA client.**

2 Extensible Authentication Protocol over LAN

router terminates the session. After the session has been completely removed, the router returns a Disconnect-ACK.

To restrict a host's access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to be causing problems on the network and network access needs to be immediately blocked for the host. When you want to restore network access on the port, re-enable it using a non-RADIUS mechanism.

CoA Request Disable Host Port

The RADIUS server CoA disable port command administratively shuts down the authentication port that is hosting a session, resulting in session termination. This command is carried in a standard CoA-Request message that has this new VSA:

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the Session Identification. If the router cannot locate the session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the router locates the session, it disables the hosting port and returns a CoA-ACK message.

If the router fails before returning a CoA-ACK to the client, the process is repeated on the new active router when the request is re-sent from the client. If the router fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is restarted on the new active router.

If the RADIUS server CoA disable port command needs to be ignored, see Configuring the Router to Ignore Bounce and Disable RADIUS CoA Requests for more information.

CoA Request Bounce-Port

A RADIUS server CoA bounce port command sent from a RADIUS server can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port. This incident can occur when there is a VLAN change and the endpoint is a device (such as a printer), that does not have a mechanism to detect a change on this authentication port. The CoA bounce port command is carried in a standard CoA-Request message that contains the following new VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the Session Identification. If the session cannot be located, the router returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the router disables the hosting port for a period of 10 seconds, re-enables it (port-bounce), and returns a CoA-ACK.

If the RADIUS server CoA bounce port command needs to be ignored, see Configuring the Router to Ignore Bounce and Disable RADIUS CoA Requests for more information.

How to Configure AAA Authentication Methods

Note AAA features are not available for use until you enable AAA globally by issuing the **aaa new-model** command.

- Configuring Login Authentication Using AAA, page 10

Configuring Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login**command to enable AAA authentication regardless of which of the supported login authentication methods you decide to use. With the **aaa authentication login**command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication line** command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication login**{**default** | *list-name*} *method1*[*method2*...]
3. Router(config)# **line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**]
4. Router(config-line)# **login authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA globally.
Step 2	Router(config)# **aaa authentication login**{**default** | *list-name*} *method1*[*method2*...]	Creates a local authentication list.
Step 3	Router(config)# **line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**]	Enters line configuration mode for the lines to which you want to apply the authentication list.

	Command or Action	Purpose
Step 4	Router(config-line)# **login authentication** **Example:** {**default** | *list-name*}	Applies the authentication list to a line or set of lines.

The *list-name* is a character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the LDAP server returns an error, enter the following command:

```
aaa authentication login default group ldap none
```

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication login default group tacacs+ none
```

Note Because the **none** keyword enables *any* user logging in to successfully authenticate, it should be used only as a backup method of authentication.

To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa authentication login default group radius
```

The table below lists the supported login authentication methods.

Table 4 AAA Authentication Login Methods

Keyword	Description
enable	Uses the enable password for authentication.
krb5	Uses Kerberos 5 for authentication.

Keyword		Description
	krb5-telnet	Uses Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. If selected, this keyword must be listed as the first method in the method list.
	line	Uses the line password for authentication.
	local	Uses the local username database for authentication.
	local-case	Uses case-sensitive local username authentication.
	none	Uses no authentication.
`group ldap`		Uses the list of all LDAP servers for authentication.
	group radius	Uses the list of all RADIUS servers for authentication.
	group tacacs+	Uses the list of all TACACS+ servers for authentication.
	group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Note The **login** command only changes username and privilege level but does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

- Preventing an Access Request with an Expired Username from Being Sent to the RADIUS Server, page 13
- Login Authentication Using Enable Password, page 14
- Login Authentication Using Kerberos, page 14
- Login Authentication Using Line Password, page 15

Preventing an Access Request with an Expired Username from Being Sent to the RADIUS Server

The following task is used to prevent an access request with an expired username from being sent to the RADIUS server. The Easy VPN client is notified by the RADIUS server that its password has expired. The password-expiry feature also provides a generic way for the user to change the password.

Note The **radius-server vsa send authentication** command must be configured to make the password-expiry feature work.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} **passwd-expiry** *method1* [*method2...*]
5. **radius-server vsa send authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA.
Step 4	**aaa authentication login** {**default** | *list-name*} **passwd-expiry** *method1* [*method2...*] **Example:** `Router(config)# aaa authentication login userauthen passwd-expiry group radius`	The **default** keyword uses the listed authentication methods that follow this keyword as the default list of methods when a user logs in. The *list-name* argument is a character string used to name the list of authentication methods activated when a user logs in. The **password-expiry** keyword enables password aging on a local authentication list. The *method* argument identifies the list of methods that the authentication algorithm tries in the given sequence. You must enter at least one method; you may enter up to four methods. The example configures password aging by using AAA with a crypto client.
Step 5	**radius-server vsa send authentication** **Example:** `Router(config)# radius-server vsa send authentication`	Sends vendor-specific attributes in access requests

Login Authentication Using Enable Password

Use the **aaa authentication login**command with the **enable** keyword to specify the enable password as the login authentication method. For example, to specify the enable password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default enable
```

Before you can use the enable password as the login authentication method, you need to define the enable password. For more information about defining enable passwords, refer to “Configuring Passwords and Privileges.”

Login Authentication Using Kerberos

Authentication via Kerberos is different from most other authentication methods: the user’s password is never sent to the remote access server. Remote users logging in to the network are prompted for a username. If the key distribution center (KDC) has an entry for that user, it creates an encrypted ticket granting ticket (TGT) with the password for that user and sends it back to the router. The user is then prompted for a password, and the router attempts to decrypt the TGT with that password. If it succeeds, the user is authenticated and the TGT is stored in the user’s credential cache on the router.

While **krb5** does use the KINIT program, a user does not need to run the KINIT program to get a TGT to authenticate to the router. This is because KINIT has been integrated into the login procedure in the Cisco IOS implementation of Kerberos.

Use the **aaa authentication login**command with the **krb5** keyword to specify Kerberos as the login authentication method. For example, to specify Kerberos as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default krb5
```

Before you can use Kerberos as the login authentication method, you need to enable communication with the Kerberos security server. See Configuring Kerberos for more information about establishing communication with a Kerberos server.

Login Authentication Using Line Password

Use the **aaa authentication login**command with the **line** keyword to specify the line password as the login authentication method. For example, to specify the line password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default line
```

Before you can use a line password as the login authentication method, you need to define a line password. For more information about defining line passwords, see Configuring Line Password Protection.

Login Authentication Using Local Password

Use the **aaa authentication login**command with the **local** keyword to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default local
```

For information about adding users into the local username database, see Establishing Username Authentication.

Login Authentication Using Group LDAP

Use the **aaa authentication login**command with the **group ldap** method to specify ldap as the login authentication method. For example, to specify ldap as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group ldap
```

Login Authentication Using Group RADIUS

Use the **aaa authentication login**command with the **group radius** method to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group radius
```

Before you can use RADIUS as the login authentication method, you need to enable communication with the RADIUS security server. See Configuring RADIUS for more information about establishing communication with a RADIUS server.

Configuring RADIUS Attribute 8 in Access Requests

Once you have used the **aaa authentication login**command to specify RADIUS and your login host has been configured to request its IP address from the NAS, you can send attribute 8 (Framed-IP-Address) in access-request packets by using the **radius-server attribute 8 include-in-access-req** command in global configuration mode. This command makes it possible for a NAS to provide the RADIUS server with a hint of the user IP address in advance of user authentication. For more information about attribute 8, refer to the appendix “RADIUS Attributes” at the end of the book.

Login Authentication Using Group TACACS

Use the **aaa authentication login**command with the **group tacacs+** method to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group tacacs+
```

Before you can use TACACS+ as the login authentication method, you need to enable communication with the TACACS+ security server. See Configuring TACACS+ more information about establishing communication with a TACACS+ server.

Login Authentication Using group group-name

Use the **aaa authentication login**command with the **group** *group-name* method to specify a subset of LDAP, RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group loginrad
```

Before you can use a group name as the login authentication method, you need to enable communication with the RADIUS or TACACS+ security server. See Configuring RADIUS more information about establishing communication with a RADIUS server. See Configuring TACACS+ for more information about establishing communication with a TACACS+ server.

Configuring PPP Authentication Using AAA

Many users access network access servers through dialup via async or ISDN. Dialup via async or ISDN bypasses the CLI completely; instead, a network protocol (such as PPP or ARA) starts as soon as the connection is established.

The AAA security services facilitate a variety of authentication methods for use on serial interfaces running PPP. Use the **aaa authentication ppp**command to enable AAA authentication regardless of which of the supported PPP authentication methods you decide to use.

To configure AAA authentication methods for serial lines using PPP, use the following commands in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication ppp**{**default** | *list-name*} *method1*[*method2...*]
3. Router(config)# **interface** *interface-type interface-number*
4. Router(config-if)# **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**][**one-time**][**optional**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA globally.
Step 2	Router(config)# **aaa authentication ppp**{**default** | *list-name*} *method1*[*method2...*]	Creates a local authentication list.
Step 3	Router(config)# **interface** *interface-type interface-number*	Enters interface configuration mode for the interface to which you want to apply the authentication list.
Step 4	Router(config-if)# **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**][**one-time**][**optional**]	Applies the authentication list to a line or set of lines. In this command, *protocol1* and *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, specified by *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

With the **aaa authentication ppp**command, you create one or more lists of authentication methods that are tried when a user tries to authenticate via PPP. These lists are applied using the **ppp authentication** line configuration command.

To create a default list that is used when a named list is *not* specified in the **ppp authentication** command, use the **default** keyword followed by the methods you want used in default situations.

For example, to specify the local username database as the default method for user authentication, enter the following command:

```
aaa authentication ppp default local
```

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication ppp default group tacacs+ none
```

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

The table below lists the supported login authentication methods.

Table 5 AAA Authentication PPP Methods

Keyword	Description
if-needed	Does not authenticate if user has already been authenticated on a TTY line.
krb5	Uses Kerberos 5 for authentication (can only be used for PAP authentication).
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

- PPP Authentication Using Kerberos, page 18
- PPP Authentication Using Local Password, page 19
- PPP Authentication Using Group RADIUS, page 19
- Configuring RADIUS Attribute 44 in Access Requests, page 19
- PPP Authentication Using Group TACACS, page 19
- PPP Authentication Using group group-name, page 19

PPP Authentication Using Kerberos

Use the **aaa authentication ppp**command with the **krb5***method* keyword to specify Kerberos as the authentication method for use on interfaces running PPP. For example, to specify Kerberos as the method of user authentication when no other method list has been defined, enter the following command:

```
aaa authentication ppp default krb5
```

Before you can use Kerberos as the PPP authentication method, you need to enable communication with the Kerberos security server. See Configuring Kerberos for more information about establishing communication with a Kerberos server.

Note Kerberos login authentication works only with PPP PAP authentication.

PPP Authentication Using Local Password

Use the **aaa authentication ppp**command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of authentication for use on lines running PPP when no other method list has been defined, enter the following command:

```
aaa authentication ppp default local
```

For information about adding users into the local username database, see Establishing Username Authentication.

PPP Authentication Using Group RADIUS

Use the **aaa authentication ppp**command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group radius
```

Before you can use RADIUS as the PPP authentication method, you need to enable communication with the RADIUS security server. See Configuring RADIUS for more information about establishing communication with a RADIUS server.

Configuring RADIUS Attribute 44 in Access Requests

Once you have used the **aaa authentication ppp**command with the **group radius** *method* to specify RADIUS as the login authentication method, you can configure your router to send attribute 44 (Acct-Session-ID) in access-request packets by using the **radius-server attribute 44 include-in-access-req** command in global configuration mode. This command allows the RADIUS daemon to track a call from the beginning of the call to the end of the call. For more information on attribute 44, refer to the appendix "RADIUS Attributes" at the end of the book.

PPP Authentication Using Group TACACS

Use the **aaa authentication ppp**command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group tacacs+
```

Before you can use TACACS+ as the PPP authentication method, you need to enable communication with the TACACS+ security server. See Configuring TACACS+ for more information about establishing communication with a TACACS+ server.

PPP Authentication Using group group-name

Use the **aaa authentication ppp**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group

name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group ppprad**:

```
aaa group server radius ppprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *ppprad*.

To specify **group ppprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group ppprad
```

Before you can use a group name as the PPP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. See Configuring RADIUS for more information about establishing communication with a RADIUS server, and Configuring TACACS+ for more information about establishing communication with a TACACS+ server.

Configuring AAA Scalability for PPP Requests

You can configure and monitor the number of background processes allocated by the PPP manager in the network access server (NAS) to deal with AAA authentication and authorization requests. In previous Cisco IOS releases, only one background process was allocated to handle all AAA requests for PPP. This meant that parallelism in AAA servers could not be fully exploited. The AAA Scalability feature enables you to configure the number of processes used to handle AAA requests for PPP, thus increasing the number of users that can be simultaneously authenticated or authorized.

To allocate a specific number of background processes to handle AAA requests for PPP, use the following command in global configuration mode:

Command or Action	Purpose
`Router(config)#` **aaa processes** *number*	Allocates a specific number of background processes to handle AAA authentication and authorization requests for PPP.

The argument *number* defines the number of background processes earmarked to process AAA authentication and authorization requests for PPP and can be configured for any value from 1 to 2147483647. Because of the way the PPP manager handles requests for PPP, this argument also defines the number of new users that can be simultaneously authenticated. This argument can be increased or decreased at any time.

Note Allocating additional background processes can be expensive. You should configure the minimum number of background processes capable of handling the AAA requests for PPP.

Configuring ARAP Authentication Using AAA

With the **aaa authentication arap** command, you create one or more lists of authentication methods that are tried when AppleTalk Remote Access Protocol (ARAP) users attempt to log in to the router. These lists are used with the **arap authentication** line configuration command.

Use the following commands starting in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication arap**
3. Router(config)# **line** *number*
4. Router(config-line)# **autoselect arap**
5. Router(config-line)# **autoselect during-login**
6. Router(config-line)# **arap authentication** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA globally.
Step 2	Router(config)# **aaa authentication arap** **Example:** {**default** | *list-name*} *method1* [*method2...*]	Enables authentication for ARAP users.
Step 3	Router(config)# **line** *number*	(Optional) Changes to line configuration mode.
Step 4	Router(config-line)# **autoselect arap**	(Optional) Enables autoselection of ARAP.
Step 5	Router(config-line)# **autoselect during-login**	(Optional) Starts the ARAP session automatically at user login.
Step 6	Router(config-line)# **arap authentication** *list-name*	(Optional--not needed if **default** is used in the **aaa authentication arap** command) Enables TACACS+ authentication for ARAP on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **arap authentication** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

The following table lists the supported login authentication methods.

Table 6 AAA Authentication ARAP Methods

Keyword	Description
auth-guest	Allows guest logins only if the user has already logged in to EXEC.
guest	Allows guest logins.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

For example, to create a default AAA authentication method list used with ARAP, enter the following command:

```
aaa authentication arap default if-needed none
```

To create the same authentication method list for ARAP but name the list *MIS-access*,enter the following command:

```
aaa authentication arap MIS-access if-needed none
```

This section includes the following sections:

ARAP Authentication Allowing Authorized Guest Logins

Use the **aaa authentication arap**command with the **auth-guest** keyword to allow guest logins only if the user has already successfully logged in to the EXEC. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all authorized guest logins--meaning logins by users who have already successfully logged in to the

EXEC--as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default auth-guest group radius
```

For more information about ARAP authorized guest logins, refer to the chapter “Configuring AppleTalk” in the *CiscoIOS AppleTalk and Novell IPX Configuration Guide* .

Note By default, guest logins through ARAP are disabled when you initialize AAA. To allow guest logins, you must use the **aaa authentication arap** command with either the **guest** or the **auth-guest** keyword.

ARAP Authentication Allowing Guest Logins

Use the **aaa authentication arap**command with the **guest** keyword to allow guest logins. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all guest logins as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default guest group radius
```

For more information about ARAP guest logins, refer to the chapter “Configuring AppleTalk” in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide* .

ARAP Authentication Using Line Password

Use the **aaa authentication arap**command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default line
```

Before you can use a line password as the ARAP authentication method, you need to define a line password. For more information about defining line passwords, refer to the section Configuring Line Password Protection.

ARAP Authentication Using Local Password

Use the **aaa authentication arap**command with the *method*keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default local
```

For information about adding users to the local username database, refer to the section Establishing Username Authentication.

ARAP Authentication Using Group RADIUS

Use the **aaa authentication arap**command with the **group radius** *method* to specify RADIUS as the ARAP authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group radius
```

Before you can use RADIUS as the ARAP authentication method, you need to enable communication with the RADIUS security server..

ARAP Authentication Using Group TACACS

Use the **aaa authentication arap**command with the **group tacacs+** *method* to specify TACACS+ as the ARAP authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group tacacs+
```

Before you can use TACACS+ as the ARAP authentication method, you need to enable communication with the TACACS+ security server. See Configuring TACACS+ for more information about establishing communication with a TACACS+ server.

ARAP Authentication Using Group group-name

Use the **aaa authentication arap**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the ARAP authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server**command to first define the members of **group araprad**:

```
aaa group server radius araprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *araprad*.

To specify **group araprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group araprad
```

Before you can use a group name as the ARAP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. See Configuring RADIUS for more information about establishing communication with a RADIUS server, and Configuring TACACS+ for more information about establishing communication with a TACACS+ server.

Configuring NASI Authentication Using AAA

With the **aaa authentication nasi** command, you create one or more lists of authentication methods that are tried when NetWare Asynchronous Services Interface (NASI) users attempt to log in to the router. These lists are used with the **nasi authentication line** configuration command.

To configure NASI authentication using AAA, use the following commands starting in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication nasi**
3. Router(config)# **line** *number*
4. Router(config-line)# **nasi authentication** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA globally.
Step 2	Router(config)# **aaa authentication nasi** **Example:** {**default** | *list-name*} *method1* [*method2...*]	Enables authentication for NASI users.
Step 3	Router(config)# **line** *number*	(Optional--not needed if **default** is used in the **aaa authentication nasi** command) Enters line configuration mode.
Step 4	Router(config-line)# **nasi authentication** *list-name*	(Optional--not needed if **default** is used in the **aaa authentication nasi** command) Enables authentication for NASI on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **aaa authentication nasi**command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

The table below lists the supported NASI authentication methods.

Table 7 AAA Authentication NASI Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.

Keyword	Description
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

NASI Authentication Using Enable Password

Use the **aaa authentication nasi**command with the *method* keyword **enable** to specify the enable password as the authentication method. For example, to specify the enable password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default enable
```

Before you can use the enable password as the authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter “Configuring Passwords and Privileges.”

NASI Authentication Using Line Password

Use the **aaa authentication nasi**command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default line
```

Before you can use a line password as the NASI authentication method, you need to define a line password. For more information about defining line passwords, refer to Configuring Line Password Protection.

NASI Authentication Using Local Password

Use the **aaa authentication nasi**command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication information. For example, to

specify the local username database as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default local
```

For information about adding users to the local username database, refer to Establishing Username Authentication.

NASI Authentication Using Group RADIUS

Use the **aaa authentication nasi**command with the **group radius** *method* to specify RADIUS as the NASI authentication method. For example, to specify RADIUS as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group radius
```

Before you can use RADIUS as the NASI authentication method, you need to enable communication with the RADIUS security server. See Configuring RADIUS for more information about establishing communication with a RADIUS server.

NASI Authentication Using Group TACACS

Use the **aaa authentication nasi**command with the **group tacacs+** *method* keyword to specify TACACS+ as the NASI authentication method. For example, to specify TACACS+ as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group tacacs+
```

Before you can use TACACS+ as the authentication method, you need to enable communication with the TACACS+ security server. See Configuring TACACS+ for more information about establishing communication with a TACACS+ server."

NASI Authentication Using group group-name

Use the **aaa authentication nasi**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the NASI authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server**command to first define the members of **group nasirad**:

```
aaa group server radius nasirad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *nasirad*.

To specify **group nasirad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group nasirad
```

Before you can use a group name as the NASI authentication method, you need to enable communication with the RADIUS or TACACS+ security server. See Configuring RADIUS for more information about establishing communication with a RADIUS server and Configuring TACACS+ for more information about establishing communication with a TACACS+ server.

Specifying the Amount of Time for Login Input

The **timeout login response**command allows you to specify how long the system will wait for login input (such as username and password) before timing out. The default login value is 30 seconds; with the **timeout login response** command, you can specify a timeout value from 1 to 300 seconds. To change the login timeout value from the default of 30 seconds, use the following command in line configuration mode:

Command or Action	Purpose
`Router(config-line)#` **timeout login response** *seconds*	Specifies how long the system will wait for login information before timing out.

Enabling Password Protection at the Privileged Level

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine whether a user can access the privileged EXEC command level. You can specify up to four authentication methods. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Use the following command in global configuration mode:

Command or Action	Purpose
`Router(config)#` **aaa authentication enable default** *method1* [*method2...*]	Enables user ID and password checking for users requesting privileged EXEC level. **Note** All **aaa authentication enable default** requests sent by the router to a RADIUS server include the username "$enab15$." Requests sent to a TACACS+ server will include the username that is entered for login authentication.

The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered. The table below lists the supported enable authentication methods.

Table 8 AAA Authentication Enable Default Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS hosts for authentication. **Note** The RADIUS method does not work on a per-username basis.

Keyword	Description
group tacacs+	Uses the list of all TACACS+ hosts for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Changing the Text Displayed at the Password Prompt

Use the **aaa authentication password-prompt** command to change the default text that the Cisco IOS software displays when prompting a user to enter a password. This command changes the password prompt for the enable password as well as for login passwords that are not supplied by remote security servers. The **no** form of this command returns the password prompt to the following default value:

```
Password:
```

The **aaa authentication password-prompt** command does not change any dialog that is supplied by a remote TACACS+ or RADIUS server.

The **aaa authentication password-prompt** command works when RADIUS is used as the login method. You will be able to see the password prompt defined in the command shown even when the RADIUS server is unreachable. The **aaa authentication password-prompt** command does not work with TACACS +. TACACS+ supplies the NAS with the password prompt to display to the users. If the TACACS+ server is reachable, the NAS gets the password prompt from the server and uses that prompt instead of the one defined in the **aaa authentication password-prompt** command. If the TACACS+ server is not reachable, the password prompt defined in the **aaa authentication password-prompt** command may be used.

Use the following command in global configuration mode:

Command or Action	Purpose
`Router(config)#` **aaa authentication password-prompt** *text-string*	Changes the default text displayed when a user is prompted to enter a password.

Preventing an Access Request with a Blank Username from Being Sent to the RADIUS Server

The following configuration steps provide the ability to prevent an Access Request with a blank username from being sent to the RADIUS server. This functionality ensures that unnecessary RADIUS server interaction is avoided, and RADIUS logs are kept short.

Note The **aaa authentication suppress null-username** command is available only in Cisco IOS XE Release 2.4 and Cisco IOS Release 12.2(33)SRD.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication suppress null-username**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# configure terminal`	Enables AAA globally.
Step 4	**aaa authentication suppress null-username** **Example:** `Router(config)# aaa authentication suppress null-username`	Prevents an Access Request with a blank username from being sent to the RADIUS server.

Configuring Message Banners for AAA Authentication

AAA supports the use of configurable, personalized login and failed-login banners. You can configure message banners that will be displayed when a user logs in to the system to be authenticated using AAA and when, for whatever reason, authentication fails.

- Configuring a Login Banner, page 30
- Configuring a Failed-Login Banner, page 31

Configuring a Login Banner

To create a login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character

is repeated at the end of the text string to signify the end of the banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a banner that will be displayed whenever a user logs in (replacing the default message for login), use the following commands in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication banner** *delimiter string delimiter*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA.
Step 2	Router(config)# **aaa authentication banner** *delimiter string delimiter*	Creates a personalized login banner.

The maximum number of characters that can be displayed in the login banner is 2996 characters.

Configuring a Failed-Login Banner

To create a failed-login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the failed-login banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a message that will be displayed whenever a user fails login (replacing the default message for failed login), use the following commands in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication fail-message** *delimiter string delimiter*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA.
Step 2	Router(config)# **aaa authentication fail-message** *delimiter string delimiter*	Creates a message to be displayed when a user fails login.

The maximum number of characters that can be displayed in the failed-login banner is 2996 characters.

Configuring AAA Packet of Disconnect

Packet of disconnect (POD) terminates connections on the network access server (NAS) when particular session attributes are identified. By using session information obtained from AAA, the POD client residing on a UNIX workstation sends disconnect packets to the POD server running on the network access server.

The NAS terminates any inbound user session with one or more matching key attributes. It rejects requests when required fields are missing or when an exact match is not found.

To configure POD, perform the following tasks in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa accounting network default**
2. Router(config)# **aaa accounting delay-start**
3. Router(config)# **aaa pod server server-key** *string*
4. Router(config)# **radius-server host** *IP address* **non-standard**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa accounting network default** **Example:** `start-stop radius`	Enables AAA accounting records.
Step 2	Router(config)# **aaa accounting delay-start**	(Optional) Delays generation of the start accounting record until the Framed-IP-Address is assigned, allowing its use in the POD packet.
Step 3	Router(config)# **aaa pod server server-key** *string*	Enables POD reception.
Step 4	Router(config)# **radius-server host** *IP address* **non-standard**	Declares a RADIUS host that uses a vendor-proprietary version of RADIUS.

Enabling Double Authentication

Previously, PPP sessions could only be authenticated by using a single authentication method: either PAP or CHAP. Double authentication requires remote users to pass a second stage of authentication--after CHAP or PAP authentication--before gaining network access.

This second (“double”) authentication requires a password that is known to the user but *not* stored on the user’s remote host. Therefore, the second authentication is specific to a user, not to a host. This provides an additional level of security that will be effective even if information from the remote host is stolen. In addition, this also provides greater flexibility by allowing customized network privileges for each user.

The second stage authentication can use one-time passwords such as token card passwords, which are not supported by CHAP. If one-time passwords are used, a stolen user password is of no use to the perpetrator.

- How Double Authentication Works, page 32
- Configuring Double Authentication, page 33
- Accessing the User Profile After Double Authentication, page 34

How Double Authentication Works

With double authentication, there are two authentication/authorization stages. These two stages occur after a remote user dials in and a PPP session is initiated.

In the first stage, the user logs in using the remote host name; CHAP (or PAP) authenticates the remote host, and then PPP negotiates with AAA to authorize the remote host. In this process, the network access privileges associated with the remote host are assigned to the user.

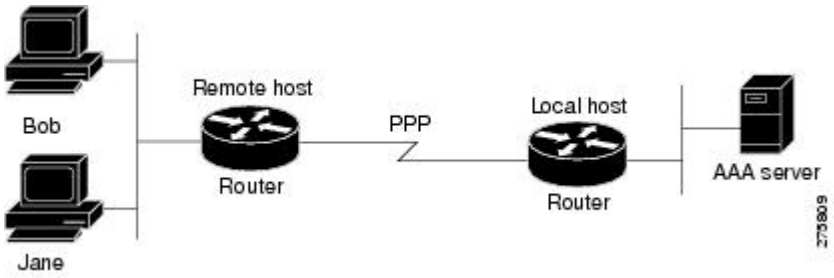
Note We suggest that the network administrator restrict authorization at this first stage to allow only Telnet connections to the local host.

In the second stage, the remote user must Telnet to the network access server to be authenticated. When the remote user logs in, the user must be authenticated with AAA login authentication. The user then must enter the **access-profile** command to be reauthorized using AAA. When this authorization is complete, the user has been double authenticated, and can access the network according to per-user network privileges.

The system administrator determines what network privileges remote users will have after each stage of authentication by configuring appropriate parameters on a security server. To use double authentication, the user must activate it by issuing the **access-profile**command.

Caution Double authentication can cause certain undesirable events if multiple hosts share a PPP connection to a network access server, as shown in the figure below. First, if a user, Bob, initiates a PPP session and activates double authentication at the network access server (per the figure below), any other user will automatically have the same network privileges as Bob until Bob's PPP session expires. This happens because Bob's authorization profile is applied to the network access server's interface during the PPP session and any PPP traffic from other users will use the PPP session Bob established. Second, if Bob initiates a PPP session and activates double authentication, and then--before Bob's PPP session has expired--another user, Jane, executes the **access-profile** command (or, if Jane Telnets to the network access server and **autocommand access-profile**is executed), a reauthorization will occur and Jane's authorization profile will be applied to the interface--replacing Bob's profile. This can disrupt or halt Bob's PPP traffic, or grant Bob additional authorization privileges Bob should not have.

Figure 2 Possibly Risky Topology: Multiple Hosts Share a PPP Connection to a Network Access Server



Configuring Double Authentication

To configure double authentication, you must complete the following steps:

1. Enable AAA by using the **aaa-new model**global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.

3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the "Configuring Authorization" chapter.
4. Configure security protocol parameters (for example, RADIUS or TACACS+). See Configuring RADIUS for more information about RADIUS and Configuring TACACS+ for more information about TACACS+.
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. (Optional) Configure the **access-profile**command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Technologies Command Reference: Network Services.*

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the chapter "Authentication Commands" in the *CiscoIOS Security Command Reference* .
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration or they can *replace* the existing interface configuration--depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *CiscoIOS Debug Command Reference* .

Accessing the User Profile After Double Authentication

In double authentication, when a remote user establishes a PPP link to the local host using the local host name, the remote host is CHAP (or PAP) authenticated. After CHAP (or PAP) authentication, PPP negotiates with AAA to assign network access privileges associated with the remote host to the user. (We suggest that privileges at this stage be restricted to allow the user to connect to the local host only by establishing a Telnet connection.)

When the user needs to initiate the second phase of double authentication, establishing a Telnet connection to the local host, the user enters a personal username and password (different from the CHAP or PAP username and password). This action causes AAA reauthentication to occur according to the personal username/password. The initial rights associated with the local host, though, are still in place. By using the **access-profile** command, the rights associated with the local host are replaced by or merged with those defined for the user in the user's profile.

To access the user profile after double authentication, use the following command in EXEC configuration mode:

Command or Action	Purpose
Router> access-profile [**merge** | **replace**] [**ignore-sanity-checks**]	Accesses the rights associated for the user after double authentication.

If you configured the **access-profile** command to be executed as an autocommand, it will be executed automatically after the remote user logs in.

Enabling Automated Double Authentication

You can make the double authentication process easier for users by implementing automated double authentication. Automated double authentication provides all of the security benefits of double authentication, but offers a simpler, more user-friendly interface for remote users. With double authentication, a second level of user authentication is achieved when the user Telnets to the network access server or router and enters a username and password. With automated double authentication, the user does not have to Telnet to the network access server; instead the user responds to a dialog box that requests a username and password or personal identification number (PIN). To use the automated double authentication feature, the remote user hosts must be running a companion client application. As of Cisco IOS Release 12.0, the only client application software available is the Glacier Bay application server software for PCs.

Note Automated double authentication, like the existing double authentication feature, is for Multilink PPP ISDN connections only. Automated double authentication cannot be used with other protocols such as X.25 or SLIP.

Automated double authentication is an enhancement to the existing double authentication feature. To configure automated double authentication, you must first configure double authentication by completing the following steps:

1. Enable AAA by using the **aaa-new model**global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the chapter "Configuring Authorization."
4. Configure security protocol parameters (for example, RADIUS or TACACS+). See Configuring RADIUS for more information about RADIUS and Configuring TACACS+ for more information about TACACS+.
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. Configure the **access-profile**command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *CiscoIOS Dial Technologies Command Reference*

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the Authentication, Authorization, and Accounting (AAA) part of the *Cisco IOSSecurity Command Reference* .
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration, or *replace* the existing interface configuration--depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference* .

After you have configured double authentication, you are ready to configure the automation enhancement.

- Configuring Automated Double Authentication, page 36
- Troubleshooting Automated Double Authentication, page 37

Configuring Automated Double Authentication

To configure automated double authentication, use the following commands, starting in global configuration mode:

SUMMARY STEPS

1. **Router(config)# ip trigger-authentication**
2. Do one of the following:
   - **Router(config)# interface bri** *number*
   -
   -
   - **Router(config)# interface serial** *number* **:23**
3. **Router(config-if)# ip trigger-authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Router(config)# ip trigger-authentication** **Example:** [**timeout** *seconds*] [**port** *number*]	Enables automation of double authentication.
Step 2	Do one of the following: • **Router(config)# interface bri** *number* • • • **Router(config)# interface serial** *number* **:23**	Selects an ISDN BRI or ISDN PRI interface and enters interface configuration mode.
Step 3	**Router(config-if)# ip trigger-authentication**	Applies automated double authentication to the interface.

Troubleshooting Automated Double Authentication

To troubleshoot automated double authentication, use the following commands in privileged EXEC mode:

SUMMARY STEPS

1. **Router# show ip trigger-authentication**
2. **Router# clear ip trigger-authentication**
3. **Router# debug ip trigger-authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Router# show ip trigger-authentication**	Displays the list of remote hosts for which automated double authentication has been attempted (successfully or unsuccessfully).
Step 2	**Router# clear ip trigger-authentication**	Clears the list of remote hosts for which automated double authentication has been attempted. This clears the table displayed by the **show ip trigger-authentication** command.
Step 3	**Router# debug ip trigger-authentication**	Displays **debug** output related to automated double authentication.

Configuring the Dynamic Authorization Service for RADIUS CoA

Use the following procedure to enable the router as an authentication, authorization, and accounting (AAA) server for dynamic authorization service to support the CoA functionality that pushes the policy map in an input and output direction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip_addr* | *hostname*} [**server-key** [**0** | **7**] *string*]
6. **domain** {**delimiter** *character*| **stripping [right-to-left]**}
7.
8. **port** {*port-num*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA.
Step 4	**aaa server radius dynamic-author** **Example:** `Router(config)# aaa server radius dynamic-author`	Sets up the local AAA server for dynamic authorization service, which must be enabled to support the CoA functionality to push the policy map in an input and output direction and enter dynamic authorization local server configuration mode. In this mode, the RADIUS application commands are configured.
Step 5	**client** {*ip_addr* | *hostname*} [**server-key** [**0** | **7**] *string*] **Example:** `Router(config-locsvr-da-radius)#client 192.168.0.5 server-key cisco1`	Configures the IP address or hostname of the AAA server client. Use the optional **server-key** keyword and *string* argument to configure the server key at the "client" level. **Note** Configuring the server key at the client level overrides the server key configured at the global level.

	Command or Action	Purpose

Step 6	**domain** {**delimiter** *character*| **stripping [right-to-left]**}
	
	Example:
	`Router(config-locsvr-da-radius)# domain stripping right-to-left`
	
	Example:
	`Router(config-locsvr-da-radius)# domain delimiter @`
		(Optional) Configures username domain options for the RADIUS application.
		• The **delimiter** keyword specifies the domain delimiter. One of the following options can be specified for the *character* argument: **@**, /, **$**, **%**, \, # or **-**
		• The **stripping** keyword compares the incoming username with the names oriented to the left of the **@** domain delimiter.
		• The **right-to-left** keyword terminates the string at the first delimiter going from right to left.

Step 7		
Step 8	**port** {*port-num*} **Example:** `Router(config-locsvr-da-radius)# port 3799`	Configures UDP port 3799 for CoA requests.

Configuring the Router to Ignore Bounce and Disable RADIUS CoA Requests

Use the following procedure to configure the router to ignore RADIUS server CoA requests in the form of a bounce port command or disable port command.

When an authentication port is authenticated with multiple hosts and there is a CoA request for one host to flap on this port or one host session to be terminated on this port, the other hosts on this port are also affected. This can trigger a DHCP renegotiation from one or more hosts in the case of a flap, or the administratively shut down the authentication port hosting the session for one or more hosts, which may be undesirable.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **authentication command bounce-port ignore**
5. **authentication command disable-port ignore**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA.
Step 4	**authentication command bounce-port ignore** **Example:** `Router(config)#` **authentication command bounce-port ignore**	(Optional) Configures the router to ignore a RADIUS server bounce port command that causes a host to link flap on an authentication port, which causes DHCP renegotiation from one or more hosts connected to this port.
Step 5	**authentication command disable-port ignore** **Example:** `Router(config)#` **authentication command disable-port ignore**	(Optional) Configures the router to ignore a RADIUS server CoA disable port command that administratively shuts down the authentication port that hosts one or more host sessions. The shutting down of the port leads to session termination.

Non-AAA Authentication Methods

Configuring Line Password Protection

This task is used to provide access control on a terminal line by entering the password and establishing password checking.

Note If you configure line password protection and then configure TACACS or extended TACACS, the TACACS username and password take precedence over line passwords. If you have not yet implemented a security policy, we recommend that you use AAA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
4. **password** *password*
5. **login**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line** [**aux**	**console**	**tty**	**vty**] *line-number* [*ending-line-number*] **Example:** `Router(config)# line console 0`	Enters line configuration mode.
Step 4	**password** *password* **Example:** `Router(config-line)# secret word`	Assigns a password to a terminal or other device on a line. The password is case sensitive and can include spaces. For example, the password “Secret” is different than the password “secret,” and “two words” is an acceptable password.

	Command or Action	Purpose
Step 5	**login** **Example:** `Router(config-line)# login`	Enables password checking at login. Line password verification can be disabled by using the **no** version of this command. **Note** The **login** command only changes the username and privilege level. It does not execute a shell; therefore autocommands are not executed. To execute autocommands under this circumstance, a Telnet session needs to be established to the router. Ensure the router is configured for secure Telnet sessions if autocommands are implemented this way.

Establishing Username Authentication

You can create a username-based authentication system, which is useful in the following situations:

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS
- To provide special-case logins: for example, access list verification, no password verification, autocommand execution at login, and “no escape” situations

To establish username authentication, use the following commands in global configuration mode as needed for your system configuration:

SUMMARY STEPS

1. Do one of the following:
   - **Router(config)# username** *name* [**nopassword** | **password** *password* | **password** *encryption-type encrypted password*]
   -
   -
   - **Router(config)# username** *name* [**access-class** *number*]
2. **Router(config)# username** *name* [**privilege** *level*]
3. **Router(config)# username** *name* [**autocommand** *command*]
4. **Router(config)# username** *name* [**noescape**] [**nohangup**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Do one of the following: • **Router(config)# username** *name* [**nopassword**	**password** *password*	**password** *encryption-type encrypted password*] • • • **Router(config)# username** *name* [**access-class** *number*]	Establishes username authentication with encrypted passwords. or (Optional) Establishes username authentication by access list.

	Command or Action	Purpose
Step 2	**Router(config)# username** *name* [**privilege** *level*]	(Optional) Sets the privilege level for the user.
Step 3	**Router(config)# username** *name* [**autocommand** *command*]	(Optional) Specifies a command to be executed automatically.
Step 4	**Router(config)# username** *name* [**noescape**] [**nohangup**]	(Optional) Sets a "no escape" login environment.

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected. The **nohangup** feature does not disconnect after using the autocommand.

Caution Passwords will be displayed in clear text in your configuration unless you enable the **service password-encryption** command. For more information about the **service password-encryption** command, refer to the chapter "Passwords and Privileges Commands" in the *CiscoIOS Security Command Reference* .

Enabling CHAP or PAP Authentication

One of the most common transport protocols used in Internet service providers' (ISPs') dial solutions is the Point-to-Point Protocol (PPP). Traditionally, remote users dial in to an access server to initiate a PPP session. After PPP has been negotiated, remote users are connected to the ISP network and to the Internet.

Because ISPs want only customers to connect to their access servers, remote users are required to authenticate to the access server before they can start up a PPP session. Normally, a remote user authenticates by typing in a username and password when prompted by the access server. Although this is a workable solution, it is difficult to administer and awkward for the remote user.

A better solution is to use the authentication protocols built into PPP. In this case, the remote user dials in to the access server and starts up a minimal subset of PPP with the access server. This does not give the remote user access to the ISP's network--it merely allows the access server to talk to the remote device.

PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection.

PPP (with or without PAP or CHAP authentication) is also supported in dialout solutions. An access server utilizes a dialout feature when it initiates a call to a remote device and attempts to start up a transport protocol such as PPP.

See "Configuring Interfaces" in the *CiscoIOS Configuration Fundamentals Configuration Guide* for more information about CHAP and PAP.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

When the remote device receives the challenge packet, it concatenates the ID, the remote device's password, and the random number, and then encrypts all of it using the remote device's password. The

remote device sends the results back to the access server, along with the name associated with the password used in the encryption process.

When the access server receives the response, it uses the name it received to retrieve a password stored in its user database. The retrieved password should be the same password the remote device used in its encryption process. The access server then encrypts the concatenated information with the newly retrieved password--if the result matches the result sent in the response packet, authentication succeeds.

The benefit of using CHAP authentication is that the remote device's password is never transmitted in clear text. This prevents other devices from stealing it and gaining illegal access to the ISP's network.

CHAP transactions occur only at the time a link is established. The access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the access server will require authentication from remote devices dialing in to the access server. If the remote device does not support the enabled protocol, the call will be dropped.

To use CHAP or PAP, you must perform the following tasks:

1. Enable PPP encapsulation.
2. Enable CHAP or PAP on the interface.
3. For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

Enabling PPP Encapsulation

To enable PPP encapsulation, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# encapsulation ppp	Enables PPP on an interface.

Enabling PAP or CHAP

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# ppp authentication {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**] [**one-time**]	Defines the authentication protocols supported and the order in which they are used. In this command, *protocol1*, *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, which is *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

If you configure **ppp authentication chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using CHAP; likewise, if you configure **ppp authentication pap**, all incoming calls that start a PPP connection will have to be authenticated via PAP. If you configure **ppp authentication chap pap**, the access server will attempt to authenticate all incoming calls that start a PPP session with CHAP. If the remote device does not support CHAP, the access server will try to authenticate the call using PAP. If the remote device does not support either CHAP or PAP, authentication will fail and the call will be dropped. If you configure **ppp authentication pap chap**, the access server will attempt to authenticate all incoming calls that start a PPP session with PAP. If the remote device does not support PAP, the access server will try to authenticate the call using CHAP. If the remote device does not support either protocol, authentication will fail and the call will be dropped. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA--they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication**command with the **if-needed** keyword means that PPP will only authenticate the remote device via PAP or CHAP if they have not yet authenticated during the life of the current call. If the remote device authenticated via a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate via CHAP if **ppp authentication chap if-needed** is configured on the interface.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp**command, you disable PPP on the line.

For information about adding a **username** entry for each remote system from which the local router or access server requires authentication, see Establishing Username Authentication, page 42.

Inbound and Outbound Authentication

PPP supports two-way authentication. Normally, when a remote device dials in to an access server, the access server requests that the remote device prove that it is allowed access. This is known as inbound authentication. At the same time, the remote device can also request that the access server prove that it is who it says it is. This is known as outbound authentication. An access server also does outbound authentication when it initiates a call to a remote device.

Enabling Outbound PAP Authentication

To enable outbound PAP authentication, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# ppp pap sent-username *username* **password** *password*	Enables outbound PAP authentication.

The access server uses the username and password specified by the **ppp pap sent-username** command to authenticate itself whenever it initiates a call to a remote device or when it has to respond to a remote device's request for outbound authentication.

Refusing PAP Authentication Requests

To refuse PAP authentication from peers requesting it, meaning that PAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# ppp pap refuse	Refuses PAP authentication from peers requesting PAP authentication.

If the **refuse** keyword is not used, the router will not refuse any PAP authentication challenges received from the peer.

Creating a Common CHAP Password

For remote CHAP authentication only, you can configure your router to create a common CHAP secret password to use in response to challenges from an unknown peer; for example, if your router calls a rotary of routers (either from another vendor, or running an older version of the Cisco IOS software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface.

To enable a router calling a collection of routers to configure a common CHAP secret password, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# ppp chap password *secret*	Enables a router calling a collection of routers to configure a common CHAP secret password.

Refusing CHAP Authentication Requests

To refuse CHAP authentication from peers requesting it, meaning that CHAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# ppp chap refuse [**callin**]	Refuses CHAP authentication from peers requesting CHAP authentication.

If the **callin** keyword is used, the router will refuse to answer CHAP authentication challenges received from the peer, but will still require the peer to answer any CHAP challenges the router sends.

If outbound PAP has been enabled (using the **ppp pap sent-username** command), PAP will be suggested as the authentication method in the refusal packet.

Delaying CHAP Authentication Until Peer Authenticates

To specify that the router will not authenticate to a peer requesting CHAP authentication until after the peer has authenticated itself to the router, use the following command in interface configuration mode:

Command or Action	Purpose
Router(config-if)# ppp chap wait *secret*	Configures the router to delay CHAP authentication until after the peer has authenticated itself to the router.

This command (which is the default) specifies that the router will not authenticate to a peer requesting CHAP authentication until the peer has authenticated itself to the router. The **no ppp chap wait**commandspecifies that the router will respond immediately to an authentication challenge.

Using MS-CHAP

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension of RFC 1994. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.

MS-CHAP differs from the standard CHAP as follows:

- MS-CHAP is enabled by negotiating CHAP Algorithm 0x80 in LCP option 3, Authentication Protocol.
- The MS-CHAP Response packet is in a format designed to be compatible with Microsoft Windows NT 3.5 and 3.51, Microsoft Windows 95, and Microsoft LAN Manager 2.*x*. This format does not require the authenticator to store a clear or reversibly encrypted password.
- MS-CHAP provides an authenticator-controlled authentication retry mechanism.
- MS-CHAP provides an authenticator-controlled change password mechanism.
- MS-CHAP defines a set of “reason-for failure” codes returned in the Failure packet message field.

Depending on the security protocols you have implemented, PPP authentication using MS-CHAP can be used with or without AAA security services. If you have enabled AAA, PPP authentication using MS-CHAP can be used in conjunction with both TACACS+ and RADIUS. The table below lists the vendor-specific RADIUS attributes (IETF Attribute 26) that enable RADIUS to support MS-CHAP.

Table 9 Vendor-Specific RADIUS Attributes for MS-CHAP

Vendor-ID Number	Vendor-Type Number	Vendor-Proprietary Attribute	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier.

- Defining PPP Authentication using MS-CHAP, page 48

Defining PPP Authentication using MS-CHAP

To define PPP authentication using MS-CHAP, use the following commands in interface configuration mode:

SUMMARY STEPS

1. **Router(config-if)# encapsulation ppp**
2. **Router(config-if)# ppp authentication ms-chap** [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Router(config-if)# encapsulation ppp**	Enables PPP encapsulation.
Step 2	**Router(config-if)# ppp authentication ms-chap** [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]	Defines PPP authentication using MS-CHAP.

If you configure **ppp authentication ms-chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using MS-CHAP. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA--they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is

defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication**command with the **if-needed** keyword means that PPP will only authenticate the remote device via MS-CHAP if that device has not yet authenticated during the life of the current call. If the remote device authenticated through a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate through MS-CHAP if **ppp authentication chap if-needed** is configured.

Note If PPP authentication using MS-CHAP is used with username authentication, you must include the MS-CHAP secret in the local username/password database. For more information about username authentication, refer to the “Establish Username Authentication” section.

Authentication Examples

RADIUS Authentication Examples

This section provides two sample configurations using RADIUS.

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login radius-login group radius local
aaa authentication ppp radius-ppp if-needed group radius
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
line 3
login authentication radius-login
interface serial 0
ppp authentication radius-ppp
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login radius-login group radius local**command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database.
- The **aaa authentication ppp radius-ppp if-needed group radius**command configures the Cisco IOS software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, PPP authentication is not performed.

- The **aaa authorization exec default group radius if-authenticated** command queries the RADIUS database for information that is used during EXEC authorization, such as autocommands and privilege levels, but only provides authorization if the user has successfully authenticated.
- The **aaa authorization network default group radius** command queries RADIUS for network authorization, address assignment, and other access lists.
- The **login authentication radius-login** command enables the radius-login method list for line 3.
- The **ppp authentication radius-ppp** command enables the radius-ppp method list for serial interface 0.

The following example shows how to configure the router to prompt for and verify a username and password, authorize the user's EXEC level, and specify it as the method of authorization for privilege level 2. In this example, if a local username is entered at the username prompt, that username is used for authentication.

If the user is authenticated using the local database, EXEC authorization using RADIUS will fail because no data is saved from the RADIUS authentication. The method list also uses the local database to find an autocommand. If there is no autocommand, the user becomes the EXEC user. If the user then attempts to issue commands that are set at privilege level 2, TACACS+ is used to attempt to authorize the command.

```
aaa authentication login default group radius local
aaa authorization exec default group radius local
aaa authorization command 2 default group tacacs+ if-authenticated
radius-server host 172.16.71.146 auth-port 1645 acct-port 1646
radius-server attribute 44 include-in-access-req
radius-server attribute 8 include-in-access-req
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login default group radius local** command specifies that the username and password are verified by RADIUS or, if RADIUS is not responding, by the router's local user database.
- The **aaa authorization exec default group radius local**command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.
- The **aaa authorization command 2 default group tacacs+ if-authenticated**command specifies TACACS+ authorization for commands set at privilege level 2, if the user has already successfully authenticated.
- The **radius-server host 172.16.71.146 auth-port 1645 acct-port 1646**command specifies the IP address of the RADIUS server host, the UDP destination port for authentication requests, and the UDP destination port for accounting requests.
- The **radius-server attribute 44 include-in-access-req** command sends RADIUS attribute 44 (Acct-Session-ID) in access-request packets.
- The **radius-server attribute 8 include-in-access-req** command sends RADIUS attribute 8 (Framed-IP-Address) in access-request packets.

TACACS Authentication Examples

The following example shows how to configure TACACS+ as the security protocol to be used for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
interface serial 0
ppp authentication chap pap test
```

```
tacacs-server host 10.1.2.3
tacacs-server key goaway
```

The lines in this sample TACACS+ authentication configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **interface** command selects the line.
- The **ppp authentication** command applies the test method list to this line.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3.
- The **tacacs-server key** command defines the shared encryption key to be "goaway."

The following example shows how to configure AAA authentication for PPP:

```
aaa authentication ppp default if-needed group tacacs+ local
```

In this example, the keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP is not necessary and can be skipped. If authentication is needed, the keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa authentication ppp MIS-access if-needed group tacacs+ local
interface serial 0
ppp authentication pap MIS-access
```

In this example, because the list does not apply to any interfaces (unlike the default list, which applies automatically to all interfaces), the administrator must select interfaces to which this authentication scheme should apply by using the **interface** command. The administrator must then apply this method list to those interfaces by using the **ppp authentication** command.

Kerberos Authentication Examples

To specify Kerberos as the login authentication method, use the following command:

```
aaa authentication login default krb5
```

To specify Kerberos authentication for PPP, use the following command:

```
aaa authentication ppp default krb5
```

AAA Scalability Example

The following example shows a general security configuration using AAA with RADIUS as the security protocol. In this example, the network access server is configured to allocate 16 background processes to handle AAA requests for PPP.

```
aaa new-model
radius-server host alcatraz
```

```
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authentication login admins local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa processes 16
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication pap dialins
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authentication login admins local** command defines another method list, “admins,” for login authentication.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa processes** command allocates 16 background processes to handle AAA requests for PPP.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the “admins” method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication pap dialins**command applies the “dialins” method list to the specified interfaces.

Login and Failed Banner Examples

The following example shows how to configure a login banner (in this case, the phrase “Unauthorized Access Prohibited”) that will be displayed when a user logs in to the system. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication login default group radius
```

This configuration produces the following login banner:

```
Unauthorized Access Prohibited
Username:
```

The following example shows how to additionally configure a failed login banner (in this case, the phrase “Failed login. Try again.”) that will be displayed when a user tries to log in to the system and fails. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication fail-message *Failed login. Try again.*
aaa authentication login default group radius
```

This configuration produces the following login and failed login banner:

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

AAA Packet of Disconnect Server Key Example

The following example shows how to configure POD (packet of disconnect), which terminates connections on the network access server (NAS) when particular session attributes are identified.

```
aaa new-model
aaa authentication ppp default radius
aaa accounting network default start-stop radius
aaa accounting delay-start
aaa pod server server-key xyz123
radius-server host 172.16.0.0 non-standard
radius-server key rad123
```

Double Authentication Examples

The examples in this section illustrate possible configurations to be used with double authentication. Your configurations could differ significantly, depending on your network and security requirements.

Note These configuration examples include specific IP addresses and other specific information. This information is for illustration purposes only: your configuration will use different IP addresses, different usernames and passwords, and different authorization statements.

- Configuration of the Local Host for AAA with Double Authentication Examples, page 54

Configuration of the Local Host for AAA with Double Authentication Examples

These two examples show how to configure a local host to use AAA for PPP and login authentication, and for network and EXEC authorization. One example is shown for RADIUS and one example for TACACS+.

In both examples, the first three lines configure AAA, with a specific server as the AAA server. The next two lines configure AAA for PPP and login authentication, and the last two lines configure network and EXEC authorization. The last line is necessary only if the **access-profile** command will be executed as an autocommand.

The following example shows router configuration with a RADIUS AAA server:

```
aaa new-model
radius-server host secureserver
radius-server key myradiuskey
aaa authentication ppp default group radius
aaa authentication login default group radius
aaa authorization network default group radius
aaa authorization exec default group radius
```

The following example shows router configuration with a TACACS+ server:

```
aaa new-model
tacacs-server host security
tacacs-server key mytacacskey
aaa authentication ppp default group tacacs+
aaa authentication login default group tacacs+
aaa authorization network default group tacacs+
aaa authorization exec default group tacacs+
```

Configuration of the AAA Server for First-Stage PPP Authentication and Authorization Example

This example shows a configuration on the AAA server. A partial sample AAA configuration is shown for RADIUS.

TACACS+ servers can be configured similarly. See Complete Configuration with TACACS Example for more information.

This example defines authentication/authorization for a remote host named “hostx” that will be authenticated by CHAP in the first stage of double authentication. Note that the ACL AV pair limits the remote host to Telnet connections to the local host. The local host has the IP address 10.0.0.2.

The following example shows a partial AAA server configuration for RADIUS:

```
hostx   Password = “welcome”
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
        cisco-avpair = “lcp:interface-config=ip unnumbered ethernet 0”,
        cisco-avpair = “ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet”,
        cisco-avpair = “ip:inacl#4=deny icmp any any”,
        cisco-avpair = “ip:route#5=10.0.0.0 255.0.0.0”,
        cisco-avpair = “ip:route#6=10.10.0.0 255.0.0.0”,
        cisco-avpair = “ipx:inacl#3=deny any”
```

Configuration of the AAA Server for Second-Stage Per-User Authentication and Authorization Examples

This section contains partial sample AAA configurations on a RADIUS server. These configurations define authentication and authorization for a user (Pat) with the username “patuser,” who will be user-authenticated in the second stage of double authentication.

TACACS+ servers can be configured similarly. See Complete Configuration with TACACS Example for more information.

Three examples show sample RADIUS AAA configurations that could be used with each of the three forms of the **access-profile** command.

The first example shows a partial sample AAA configuration that works with the default form (no keywords) of the **access-profile** command. Note that only ACL AV pairs are defined. This example also sets up the **access-profile** command as an autocommand.

```
patuser   Password = “welcome”
          User-Service-Type = Shell-User,
          cisco-avpair = “shell:autocmd=access-profile”
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = “ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet”,
          cisco-avpair = “ip:inacl#4=deny icmp any any”
```

The second example shows a partial sample AAA configuration that works with the **access-profile merge** form of the **access-profile** command. This example also sets up the **access-profile merge**command as an autocommand.

```
patuser   Password = “welcome”
          User-Service-Type = Shell-User,
          cisco-avpair = “shell:autocmd=access-profile merge”
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = “ip:inacl#3=permit tcp any any”
          cisco-avpair = “ip:route=10.0.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.1.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.2.0.0 255.255.0.0"
```

The third example shows a partial sample AAA configuration that works with the **access-profile replace** form of the **access-profile** command. This example also sets up the **access-profile replace**command as an autocommand.

```
patuser   Password = “welcome”
          User-Service-Type = Shell-User,
          cisco-avpair = “shell:autocmd=access-profile replace”
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = “ip:inacl#3=permit tcp any any”,
          cisco-avpair = “ip:inacl#4=permit icmp any any”,
          cisco-avpair = “ip:route=10.10.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.11.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.12.0.0 255.255.0.0"
```

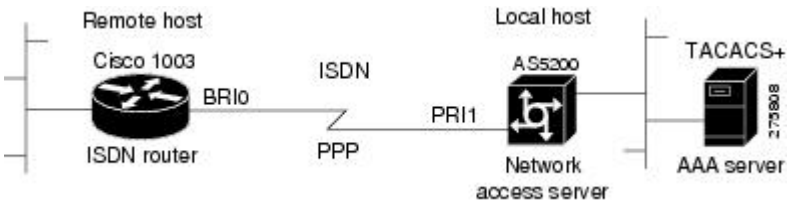
Complete Configuration with TACACS Example

This example shows TACACS+ authorization profile configurations both for the remote host (used in the first stage of double authentication) and for specific users (used in the second stage of double authentication). This TACACS+ example contains approximately the same configuration information as shown in the previous RADIUS examples.

This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.” The configurations for these three usernames illustrate different configurations that correspond to the three different forms of the **access-profile** command. The three user configurations also illustrate setting up the autocommand for each form of the **access-profile** command.

The figure below shows the topology. The example that follows the figure shows a TACACS+ configuration file.

Figure 3 Example Topology for Double Authentication



This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.”

```
key = “mytacacskey”
default authorization = permit
#-----------------------------Remote Host (BRI)------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#----------------------------------------------------------------------
user = hostx
{
    login = cleartext “welcome”
    chap = cleartext “welcome”
    service = ppp protocol = lcp {
                interface-config=”ip unnumbered ethernet 0"
    }
    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.
            inacl#3=”permit tcp any 172.21.114.0 0.0.0.255 eq telnet”
            inacl#4=”deny icmp any any”
            route#5=”10.0.0.0 255.0.0.0"
            route#6=”10.10.0.0 255.0.0.0"
    }
    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3=”deny any”
    }
}
#------------------- “access-profile” default user “only acls” ------------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#-------------------------------------------------------------------------------
user = pat_default
{
        login = cleartext “welcome”
        chap = cleartext “welcome”
        service = exec
        {
                # This is the autocommand that executes when pat_default logs in.
```

```
                autocmd = “access-profile”
        }
        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3=”permit tcp any host 10.0.0.2 eq telnet”
                inacl#4=”deny icmp any any”
        }
        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#--------------------- “access-profile merge” user --------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
#
#-------------------------------------------------------------------------------
user = pat_merge
{
        login = cleartext “welcome”
        chap = cleartext “welcome”
        service = exec
        {
                # This is the autocommand that executes when pat_merge logs in.
                autocmd = “access-profile merge”
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3=”permit tcp any any”
                route#2=”10.0.0.0 255.255.0.0"
                route#3=”10.1.0.0 255.255.0.0"
                route#4=”10.2.0.0 255.255.0.0"
        }
        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#--------------------- “access-profile replace” user ---------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be “invalid” for this context.
#-------------------------------------------------------------------------------
user = pat_replace
{
        login = cleartex
t
“
```

```
welcome
”
        chap = cleartext “welcome”
        service = exec
        {
                # This is the autocommand that executes when pat_replace logs in.
                autocmd = “access-profile replace”
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3=”permit tcp any any”
                inacl#4=”permit icmp any any”
                route#2=”10.10.0.0 255.255.0.0"
                route#3=”10.11.0.0 255.255.0.0"
                route#4=”10.12.0.0 255.255.0.0"
        }
        service = ppp protocol = ipx
        {
                # put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
```

Automated Double Authentication Example

This example shows a complete configuration file for a Cisco 2509 router with automated double authentication configured. The configuration commands that apply to automated double authentication are preceded by descriptions with a double asterisk (**).

```
Current configuration:
!
version 11.3
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the TACACS+ AAA server:
aaa authentication login default group tacacs+
aaa authentication login console none
! **The following command enables device authentication via the TACACS+ AAA server:
aaa authentication ppp default group tacacs+
! **The following command causes the remote user’s authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization exec default group tacacs+
! **The following command causes the remote device’s authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization network default group tacacs+
enable password mypassword
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 172.16.2.75
! **The following command globally enables automated double authentication:
ip trigger-authentication timeout 60 port 7500
isdn switch-type basic-5ess
```

```
!
!
interface Ethernet0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
 ip address 172.21.127.105 255.255.255.248
 encapsulation ppp
 no ip mroute-cache
 no keepalive
 shutdown
 clockrate 2000000
 no cdp enable
!
interface Serial1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no cdp enable
!
! **Automated double authentication occurs via the ISDN BRI interface BRI0:
interface BRI0
 ip unnumbered Ethernet0
! **The following command turns on automated double authentication at this interface:
 ip trigger-authentication
! **PPP encapsulation is required:
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer idle-timeout 500
 dialer map ip 172.21.127.113 name myrouter 60074
 dialer-group 1
 no cdp enable
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 171.69.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
```

```
!
end
```

MS-CHAP Example

The following example shows how to configure a Cisco AS5200 Universal Access Server (enabled for AAA and communication with a RADIUS security server) for PPP authentication using MS-CHAP:

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
username root password ALongPassword
radius-server host alcatraz
radius-server key myRaDiUSpassWoRd
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication ms-chap dialins
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines another method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication ms-chap dialins**command selects MS-CHAP as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the “admins” method list for login authentication.

- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

Additional References

Related Documents

Related Topic	Document Title
Authorization	Configuring Authorization module.
Accounting	Configuring Accounting module.
RADIUS server	Configuring RADIUS module.
TACACS+ server	Configuring TACACS+ module.
Kerberos	Configuring Kerberos module.

Standards and RFCs

Standard/RFC	Title
RFC 2903	*Generic AAA Architecture*
RFC 2904	*AAA Authorization Framework*
RFC 2906	*AAA Authorization Requirements*
RFC 2989	*Criteria for Evaluating AAA Protocols for Network Access*
RFC 5176	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10 Feature Information for Configuring Authentication

Feature Name	Releases	Feature Information
Authentication	12.0 XE 2.1	This feature was introduced in the Cisco IOS Release 12.0 software. This feature was introduced in the Cisco IOS Release XE 2.1 software.
AAA Per-User Scalability	12.2(27)SB 12.2(33)SR 15.0(1)M	This feature was introduced in Cisco IOS Release 12.2(27)SB. This feature was integrated into Cisco IOS Release 12.2(33)SR. This feature was integrated into Cisco IOS Release 15.0(1)M.

Feature Name	Releases	Feature Information
RADIUS - CLI to Prevent Sending of Access Request with a Blank Username	12.2(33)SRD Cisco IOS XE Release 2.4	This Authentication feature prevents an Access Request with a blank username from being sent to the RADIUS server. This functionality ensures that unnecessary RADIUS server interaction is avoided and RADIUS logs are kept short. The following command was introduced: **aaa authentication suppress null-username**.
LDAP integration with Active Directory	15.1(1)T	This feature provides the authentication and authorization support for AAA. LDAP is a standard-based protocol used to access directories. It is based on a client server model similar to RADIUS. LDAP is deployed on Cisco devices to send authentication requests to a central LDAP server that contains all user authentication and network service access information. The following command was introduced: **aaa authentication login default group ldap**

Feature Name	Releases	Feature Information
Change of Authorization (COA)	12.2(33)SXI4, 15.2(2)T	Beginning with Cisco IOS Release 12.2(33)SXI4, and integrated into Cisco IOS 15.2(2)T, the Cisco IOS supports the RADIUS Change of Authorization (CoA) extensions defined in RFC 5176. COA extensions are typically used in a pushed model and allow for the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers. The following commands were introduced: **aaa server radius dynamic author**, **authentication command bounce-port ignore**, **authentication command disable-port ignore**.

AAA Double Authentication Secured by Absolute Timeout

The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. This feature optimizes the connection to the network by service providers to only connections that are authorized, and it increases the security of the overall access to the network by ensuring that no unwanted sessions are connected.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for AAA Double Authentication Secured by Absolute Timeout

- You need access to a Cisco RADIUS or TACACS+ server and should be familiar with configuring RADIUS or TACACS+.
- You should be familiar with configuring authentication, authorization, and accounting (AAA).
- You should be familiar with enabling AAA automated double authentication.

Restrictions for AAA Double Authentication Secured by Absolute Timeout

- The AAA Double Authentication Secured by Absolute Timeout feature, like the existing double authentication feature, is for PPP connections only. Automated double authentication cannot be used with other protocols, such as X.25 or Serial Line Internet Protocol (SLIP).
- There may be a minimal impact on performance if a TACACS+ server is used. However, there is no performance impact if a RADIUS server is used.

Information About AAA Double Authentication Secured by Absolute Timeout

AAA Double Authentication

With the current AAA double authentication mechanism, a user must pass the first authentication using a host username and password. The second authentication, after Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP), uses a login username and password. In the first authentication, a PPP session timeout will be applied to the virtual access interface if it is configured locally or remotely. The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. The per-user timeout, which can be customized, supersedes the generic absolute timeout value. This method works on the same principle as per-user access control lists (ACLs) in double authentication.

How to Apply AAA Double Authentication Secured by Absolute Timeout

Applying AAA Double Authentication Secured by Absolute Timeout

To apply the absolute timeout, you need to configure “Session-Timeout” in the login user profile as a link control protocol (LCP) per-user attribute. There is no new or modified command-line interface (CLI) for this feature, but before you use the **access-profile** command when enabling AAA double authentication, you must first reauthorize LCP per-user attributes (for example, Session-Timeout) and then reauthorize Network Control Protocols (NCPs) to apply other necessary criteria, such as ACLs and routes. See the Example for AAA Double Authentication Secured by Absolute Timeout.

Note Timeout configuration in a TACACS+ user profile is a little different from the configuration in a RADIUS user profile. In a RADIUS profile, only one “Session-Timeout” is configured, along with the autocommand “access-profile.” The timeout will be applied to the EXEC session and to the PPP session. In TACACS+, however, the timeout must be configured under the service types “exec” and “ppp” (LCP) to apply a timeout to the EXEC session and to the PPP session. If the timeout is configured only under the service type “ppp,” the timeout value is not available while doing an EXEC authorization--and the timeout will not be applied to the EXEC session.

Verifying AAA Double Authentication Secured by Absolute Timeout

To verify that AAA double authentication has been secured by absolute timeout and to see information about various attributes associated with the authentication, perform the following steps. These **show** and **debug** commands can be used in any order.

Note When idle timeout is configured on a full virtual access interface and a subvirtual access interface, the **show users** command displays the idle time for both the interfaces. However, if the idle timeout is not configured on both interfaces, then the **show users** command will display the idle time for the full virtual access interface only.

or

debug tacacs

SUMMARY STEPS

1. **enable**
2. **show users**
3. **show interfaces virtual-access** *number* [**configuration**]
4. **debug aaa authentication**
5. **debug aaa authorization**
6. **debug aaa per-user**
7. **debug ppp authentication**
8. Do one of the following:
   - **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Command or Action	Purpose
Step 2 **show users** **Example:** **enable** **Example:** `Router# show users`	Displays information about the active lines on the router.
Step 3 **show interfaces virtual-access** *number* [**configuration**] **Example:** `Router# show interfaces virtual-access 2 configuration`	Displays status, traffic data, and configuration information about a specified virtual access interface.
Step 4 **debug aaa authentication** **Example:** `Router# debug aaa authentication`	Displays information about AAA TACACS+ authentication.
Step 5 **debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information about AAA TACACS+ authorization.
Step 6 **debug aaa per-user** **Example:** `Router# debug aaa per-user`	Displays the attributes that are applied to each user as the user authenticates.
Step 7 **debug ppp authentication** **Example:** `Router# debug ppp authentication`	Displays whether a user is passing authentication.

Command or Action	Purpose

Step 8 Do one of the following:
• **debug radius**
Example:
`Router# debug radius`
Example:
Example:
Example:
debug tacacs
Example:
`Router# debug tacacs`
	Displays information associated with the RADIUS server.
	or
	Displays information associated with the TACACS+ server.

Examples

The following sample output is from the **show users** command:

```
Router# show users
     Line       User        Host(s      Idle       Location
   *  0 con 0    aaapbx2     idle        00:00:00   aaacon2 10
      8 vty 0    broker_def  idle        00:00:08   192.168.1.8
      Interface  User            Mode        Idle       Peer Address
      Vi2        broker_default  VDP         00:00:01   192.168.1.8 <=========
      Se0:22     aaapbx2         Sync PPP    00:00:23
```

The following sample output is from the **show interfaces virtual-access** command:

```
Router# show interfaces virtual-access 2 configuration
Virtual-Access2 is a Virtual Profile (sub)interface
Derived configuration: 150 bytes
!
interface Virtual-Access2
  ip unnumbered Serial0:23
  no ip route-cache
  timeout absolute 3 0
! The above line shows that the per-user session timeout has been applied.
  ppp authentication chap
  ppp timeout idle 180000
! The above line shows that the absolute timeout has been applied.
```

Examples for AAA Double Authentication Secured by Absolute Timeout

RADIUS User Profile Example

The following sample output shows that a RADIUS user profile has been applied and that AAA double authentication has been secured by an absolute timeout:

```
aaapbx2 Password = "password1",
 Service-Type = Framed,
 Framed-Protocol = PPP,
 Session-Timeout = 180,
 Idle-Timeout = 180000,
 cisco-avpair = "ip:inacl#1=permit tcp any any eq telnet"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
broker_default Password = "password1",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
broker_merge Password = "password1",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile merge",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
 cisco-avpair = "ip:route#3=10.4.0.0 255.0.0.0"
 cisco-avpair = "ip:route#4=10.5.0.0 255.0.0.0"
 cisco-avpair = "ip:route#5=10.6.0.0 255.0.0.0"
broker_replace Password = "password1",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile replace",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
 cisco-avpair = "ip:route#3=10.4.0.0 255.0.0.0"
 cisco-avpair = "ip:route#4=10.5.0.0 255.0.0.0"
 cisco-avpair = "ip:route#5=10.6.0.0 255.0.0.0"
```

TACACS User Profile Example

The following sample output shows that a TACACS+ user profile has been applied and that AAA double authentication has been secured by an absolute timeout.

Remote Host

The following allows the remote host to be authenticated by the local host during first-stage authentication and provides the remote host authorization profile.

```
user = aaapbx2
 chap = cleartext Cisco
 pap = cleartext cisco
 login = cleartext cisco
 service = ppp protocol = lcp
  idletime = 3000
```

```
  timeout = 3
 service = ppp protocol = ip
  inacl#1="permit tcp any any eq telnet"
 service = ppp protocol = ipx
```

access-profile Command Without Any Arguments

Using the **access-profile** command without any arguments causes the removal of any access lists that are found in the old configuration (both per-user and per-interface) and ensures that the new profile contains only access-list definitions.

```
user = broker_default
 login = cleartext Cisco
 chap = cleartext "cisco"
 service = exec
  autocmd = "access-profile"
! This is the autocommand that executes when broker_default logs in.
  timeout = 6
 service = ppp protocol = lcp
  timeout = 6
 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
  inacl#1="permit tcp any any"
  inacl#2="permit icmp host 10.0.0.0 any"
 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

access-profile Command with merge Keyword

With the “merge” option, all old access lists are removed (as before), but then almost any AV pair is allowed to be uploaded and installed. This merge will allow for the uploading of any custom static routes, Service Advertisement Protocol (SAP) filters, and other requirements that the user may need in his or her profile. This merge must be used with care because it leaves everything open in terms of conflicting configurations.

```
user = broker_merge
 login = cleartext Cisco
 chap = cleartext "cisco"
 service = exec
  autocmd = "access-profile merge"
! This is the autocommand that executes when broker_merge logs in.
  timeout = 6
 service = ppp protocol = lcp
 timeout = 6
 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
  route#1="10.4.0.0 255.0.0.0"
  route#2="10.5.0.0 255.0.0.0"
  route#3="10.6.0.0 255.0.0.0"
  inacl#5="permit tcp any any"
  inacl#6="permit icmp host 10.60.0.0 any"
 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

access-profile Command with the replace Keyword

If you use the **access-profile** command with the **replace** keyword, the command works as it does currently; that is, any old configuration is removed and any new configuration is installed.

Note When the **access-profile** command is configured, the new configuration is checked for address pools and address attribute-value (AV) pairs. Because addresses cannot be renegotiated at this point, the command will fail to work when it encounters such an address AV pair.

```
user = broker_replace
 login = cleartext Cisco
 chap = cleartext "cisco"
 service = exec
  autocmd = "access-profile replace"
! This is the autocommand that executes when broker_replace logs in.
  timeout = 6
 service = ppp protocol = lcp
  timeout = 6
 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
  route#1="10.7.0.0 255.0.0.0"
  route#2="10.8.0.0 255.0.0.0"
  route#3="10.9.0.0 255.0.0.0"
  inacl#4="permit tcp any any"
 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

Note Timeout configuration in a TACACS+ user profile is a little different from the configuration in a RADIUS user profile. In a RADIUS profile, only one “Session-Timeout” is configured, along with the autocommand **access-profile**. The timeout will be applied to the EXEC session and to the PPP session. In TACACS+, however, the timeout must be configured under the service types “exec” and “ppp” (LCP) to apply a timeout to the EXEC session and to the PPP session. If the timeout is configured only under the service type “ppp,” the timeout value is not available while doing an EXEC authorization--and the timeout will not be applied to the EXEC session.

Additional References

The following sections provide references related to AAA Double Authentication Secured by Absolute Timeout.

Related Documents

Related Topic	Document Title
AAA	Configuring Authentication feature module.
	Configuring Authorization feature module.
	Configuring Accounting feature module.
RADIUS	Configuring RADIUS feature module.
TACACS+	Configuring TACACS+ feature module
Security Commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for AAA Double Authentication Secured by Absolute Timeout

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11 Feature Information for AAA Double Authentication Secured by Absolute Timeout

Feature Name	Releases	Feature Information
AAA Double Authentication Secured by Absolute Timeout	12.3(7)T 12.2(28)SB Cisco IOS XE Release 2.3	The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. This feature optimizes the connection to the network by service providers to only connections that are authorized, and it increases the security of the overall access to the network by ensuring that no unwanted sessions are connected.

Login Password Retry Lockout

The Login Password Retry Lockout feature allows system administrators to lock out a local authentication, authorization, and accounting (AAA) user account after a configured number of unsuccessful attempts by the user to log in.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Login Password Retry Lockout

- You must be running a Cisco IOS image that contains the AAA component.

Restrictions for Login Password Retry Lockout

- Authorized users can lock themselves out because there is no distinction between an attacker who is guessing passwords and an authorized user who is entering the password incorrectly multiple times.
- A denial of service (DoS) attack is possible; that is, an authorized user could be locked out by an attacker if the username of the authorized user is known to the attacker.

Information About Login Password Retry Lockout

Lock Out of a Local AAA User Account

The Login Password Retry Lockout feature allows system administrators to lock out a local AAA user account after a configured number of unsuccessful attempts by the user to log in using the username that corresponds to the AAA user account. A locked-out user cannot successfully log in again until the user account is unlocked by the administrator.

A system message is generated when a user is either locked by the system or unlocked by the system administrator. The following is an example of such a system message:

```
%AAA-5-USER_LOCKED: User user1 locked out on authentication failure.
```

The system administrator cannot be locked out.

Note The system administrator is a special user who has been configured using the maximum privilege level (root privilege--level 15). A user who has been configured using a lesser privilege level can change the privilege level using the **enable** command. A user that can change to the root privilege (level 15) is able to act as a system administrator.

This feature is applicable to any login authentication method, such as ASCII, Challenge Handshake Authentication Protocol (CHAP), and Password Authentication Protocol (PAP).

Note No messages are displayed to users after authentication failures that are due to the locked status (that is, there is no distinction between a normal authentication failure and an authentication failure due to the locked status of the user).

How to Configure Login Password Retry Lockout

Configuring Login Password Retry Lockout

To configure the Login Password Retry Lockout feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **username** *name* [**privilege** *level*] **password** *encryption-type password*
4. **aaa new-model**
5. **aaa local authentication attempts max-fail** *number-of-unsuccessful-attempts*
6. **aaa authentication login default method**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**username** *name* [**privilege** *level*] **password** *encryption-type password* **Example:** `Router(config)# username user1 privilege 15 password 0 cisco`	Establishes a username-based authentication system.
Step 4	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 5	**aaa local authentication attempts max-fail** *number-of-unsuccessful-attempts* **Example:** `Router(config)# aaa local authentication attempts max-fail 3`	Specifies the maximum number of unsuccessful attempts before a user is locked out.

	Command or Action	Purpose
Step 6	**aaa authentication login default method** **Example:** `Router(config)# aaa authentication login default local`	Sets the authentication, authorization, and accounting (AAA) authentication method at login. For example, **aaa authentication login default** *local* specifies the local AAA user database.

Unlocking a Login Locked-Out User

To unlock a login locked-out user, perform the following steps.

Note This task can be performed only by users having the root privilege (level 15).

SUMMARY STEPS

1. **enable**
2. **clear aaa local user lockout** {**username** *username* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear aaa local user lockout** {**username** *username*	**all**} **Example:** `Router# clear aaa local user lockout username user1`	Unlocks a locked-out user.

Clearing the Unsuccessful Login Attempts of a User

This task is useful for cases in which the user configuration was changed and the unsuccessful login attempts of a user that are already logged must be cleared.

To clear the unsuccessful login attempts of a user that have already been logged, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear aaa local user fail-attempts** {**username** *username* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear aaa local user fail-attempts** {**username** *username* | **all**} **Example:** `Router# clear aaa local user fail-attempts username user1`	Clears the unsuccessful attempts of the user. • This command is useful for cases in which the user configuration was changed and the unsuccessful attempts that are already logged must be cleared.

Monitoring and Maintaining Login Password Retry Lockout Status

To monitor and maintain the status of the Login Password Retry Lockout configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show aaa local user lockout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show aaa local user lockout** **Example:** `Router# show aaa local user lockout`	Displays a list of the locked-out users for the current login password retry lockout configuration.

Example

The following output shows that user1 is locked out:

```
Router# show aaa local user lockout
```

```
Local-user          Lock time
user1               04:28:49 UTC Sat Jun 19 2004
```

Configuration Examples for Login Password Retry Lockout

Displaying the Login Password Retry Lockout Configuration Example

The following **show running-config** command output illustrates that the maximum number of failed user attempts has been set for 2 as the login password retry lockout configuration:

```
Router # show running-config
Building configuration...
Current configuration : 1214 bytes
!
version 12.3
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname LAC-2
!
boot-start-marker
boot-end-marker
!
!
username sysadmin
username sysad privilege 15 password 0 cisco
username user1 password 0 cisco
aaa new-model
aaa local authentication attempts max-fail 2
!
!
aaa authentication login default local
aaa dnis map enable
aaa session-id common
```

Additional References

The following sections provide references related to Login Password Retry Lockout.

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Login Password Retry Lockout

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 12 Feature Information for Login Password Retry Lockout

Feature Name	Releases	Feature Information
Login Password Retry Lockout	12.3(14)T 12.2(33)SRE	The Login Password Retry Lockout feature allows system administrators to lock out a local AAA user account after a configured number of unsuccessful attempts by the user to log in. This feature was introduced in Cisco IOS Release 12.3(14)T. This feature was integrated into Cisco IOS Release 12.2(33)SRE. The following commands were introduced or modified: **aaa local authentication attempts max-fail**, **clear aaa local user fail-attempts**, **clear aaa local user lockout**.

Glossary

- **local AAA method** --Method by which it is possible to configure a local user database on a router and to have AAA provision authentication or authorization of users from this database.
- **local AAA user** --User who is authenticated using the local AAA method.

Throttling of AAA RADIUS Records

The Throttling of AAA (RADIUS) Records feature supports throttling of access (authentication and authorization) and accounting records that are sent to the RADIUS server. This feature allows a user to configure the appropriate throttling rate to avoid network congestion and instability; such as when there is insufficient bandwidth to accommodate a sudden burst of records generated from the Cisco IOS router to the RADIUS server.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Throttling of AAA RADIUS Records

Benefits of the Throttling of AAA RADIUS Records Feature

A Network Access Server (NAS), acting as RADIUS client, can generate a burst of accounting or access requests, causing severe network congestion or causing the RADIUS server to become overloaded with a burst of RADIUS traffic. This problem could be compounded when multiple NASs interact with the RADIUS servers.

The following conditions can trigger a sudden burst of RADIUS traffic:

- An interface flap, which in turn brings down all the subscriber sessions and generates accounting requests for each subscriber.
- The Cisco IOS High Availability (HA) program generating a START record for every session that survived a switchover, such as the scenario described the preceding bullet.

A large number of generated requests can make the network unstable if there is insufficient bandwidth or if the RADIUS server is slow to respond. Neither the User Datagram Protocol (UDP) transport layer nor the RADIUS protocol has a flow control mechanism. The throttling mechanism provided by this feature provides a solution for these issues.

Throttling Access Requests and Accounting Records

The Throttling of AAA (RADIUS) Records feature introduces a mechanism to control packets (flow control) at the NAS level, which improves the RADIUS server performance.

Because of their specific uses, access requests and accounting records must be treated separately. Access request packets are time sensitive, while accounting record packets are not.

- If a response to an access request is not returned to the client in a timely manner, the protocol or the user will time out, impacting the device transmission rates.
- Accounting records packets are not real-time critical.

When configuring threshold values on the same server, it is important to prioritize threshold values for the handling of the time-sensitive access request packets and to place a lesser threshold value on the accounting records packets.

In some cases, when an Internet Service Provider (ISP) is using separate RADIUS servers for access requests and accounting records, only accounting records throttling may be required.

- The Throttling of AAA (RADIUS) Records is disabled, by default.
- Throttling functionality can be configured globally or at server group level.

How to Configure Throttling of AAA RADIUS Records

This section describes how to configure throttling of access (authentication and authorization) and accounting records that are sent to the RADIUS server for both, global and server groups.

Server-group configurations are used to enable or disable throttling for a particular server group and to specify the threshold value for that server group.

Note Server-group configurations override any configured global configurations.

- Throttling Accounting and Access Request Packets Globally, page 86
- Throttling Accounting and Access Request Packets Per Server Group, page 87

Throttling Accounting and Access Request Packets Globally

To globally configure the throttling of accounting and access request packets, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server throttle {** [**accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]}
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server throttle {** [**accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]} **Example:** `Router(config)# radius-server throttle accounting 100 access 200 access-timeout 2`	Configures global throttling for accounting and access request packets. For this example: • The accounting threshold value (the range is 0-65536) is set to 100, and the access threshold value is set to 200. **Note** The default threshold value is 0 (throttling disabled). • The number of timeouts per transaction value (the range is 1-10) is set to 2.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Throttling Accounting and Access Request Packets Per Server Group

The following server-group configuration can be used to enable or disable throttling for a specified server group and to specify the threshold value for that server group.

To configure throttling of server-group accounting and access request packets, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *server-group-name*
4. **throttle** {[**accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *server-group-name* **Example:** `Router(config)# aaa group server radius myservergroup`	Enters server-group configuration mode.
Step 4	**throttle** {[**accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]} **Example:** `Router(config-sg-radius)# throttle accounting 100 access 200 access-timeout 2`	Configures the specified server-group throttling values for accounting and access request packets. For this example: • The accounting threshold value (the range is 0-65536) is set to 100, and the access threshold value is set to 200. **Note** The default threshold value is 0 (throttling disabled). • The number of time-outs per transaction value (the range is 1-10) is set to 2.
Step 5	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits server-group configuration mode.

Configuration Examples for Throttling of AAA RADIUS Records

Throttling Accounting and Access Request Packets Globally Example

The following example shows how to limit the number of accounting requests sent to a server to 100:

```
enable
configure terminal
radius-server throttle accounting 100
```

The following example shows how to limit the number of access requests packets sent to a server to 200 and sets the number of time-outs allowed per transactions to 2:

```
enable
configure terminal
radius-server throttle access 200
radius-server throttle access 200 access-timeout 2
```

The following example shows how to throttle both accounting and access request packets:

```
enable
configure terminal
radius-server throttle accounting 100 access 200
```

Throttling Accounting and Access Request Packets Per Server Group Example

The following example shows how to limit the number of accounting requests sent to server-group-A to 100:

```
enable
configure terminal
aaa group server radius server-group-A
throttle accounting 100
```

The following example shows how to limit the number of access requests packets sent to server-group-A to 200 and sets the number of time-outs allowed per transactions to 2:

```
enable
configure terminal
aaa group server radius server-group-A
throttle access 200 access-timeout 2
```

The following example shows how to throttle both accounting and access request packets for server-group-A:

```
enable
configure terminal
aaa group server radius server-group-A
throttle accounting 100 access 200
```

Additional References

The following sections provide references related to the Throttling of AAA (RADIUS) Records feature.

Related Documents

Related Topic	Document Title
AAA and RADIUS	*Cisco IOS Security Configuration Guide: Securing User Services*, Release 15.0.

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Throttling of AAA RADIUS Records

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13 Feature Information for Throttling of AAA (RADIUS) Records

Feature Name	Releases	Feature Information
Throttling of AAA (RADIUS) Records	12.2(33)SRC 12.4(20)T	The Throttling of AAA (RADIUS) Records feature supports throttling of access (authentication and authorization) and accounting records that are sent to the RADIUS server. This feature allows a user to configure the appropriate throttling rate to avoid network congestion and instability; such as when there is insufficient bandwidth to accommodate a sudden burst of records generated from the Cisco IOS router to the RADIUS server. In Release 12.2(33)SRC, this feature was introduced on the Cisco 7200 and Cisco 7200 routers. The following commands were introduced or modified by this feature: **radius-server throttle, throttle**

MSCHAP Version 2

The MSCHAP Version 2 feature (introduced in Cisco IOS Release 12.2(2)XB5) allows Cisco routers to utilize Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAP V2) authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server (NAS).

For Cisco IOS Release 12.4(6)T, MSCHAP V2 now supports a new feature: AAA Support for MSCHAPv2 Password Aging. Prior to Cisco IOS Release 12.4(6)T, when Password Authentication Protocol (PAP)-based clients sent username and password values to the authentication, authorization, and accounting (AAA) subsystem, AAA generated an authentication request to the RADIUS server. If the password expired, the RADIUS server replied with an authentication failure message. The reason for the authentication failure was not passed back to AAA subsystem; thus, users were denied access because of authentication failure but were not informed why they were denied access.

The Password Aging feature, available in Cisco IOS Release 12.4(6)T, notifies crypto-based clients that the password has expired and provides a generic way for the user to change the password. The Password Aging feature supports only crypto-based clients.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MSCHAP Version 2

- Configure an interface type and enter interface configuration mode by using the **interface** command.
- Configure the interface for PPP encapsulation by using the **encapsulation** command.
- Be sure that the client operating system supports all MSCHAP V2 capabilities.
- For Cisco IOS Release 12.4(6)T, the Password Aging feature only supports RADIUS authentication for crypto-based clients.
- To ensure that the MSCHAP Version 2 features correctly interpret the authentication failure attributes sent by the RADIUS server, you must configure the **ppp max-bad-auth** command and set the number of authentication retries at two or more.
- In order for the MSCHAP Version 2 feature to support the ability to change a password, the authentication failure attribute, which is sent by the RADIUS server, must be correctly interpreted as described in Configuring MSCHAP V2 Authentication.

In addition, the **radius server vsa send authentication** command must be configured, allowing the RADIUS client to send a vendor-specific attribute to the RADIUS server. The Change Password feature is supported only for RADIUS authentication.

- The Microsoft Windows 2000, Microsoft Windows XP, and Microsoft Windows NT operating systems have a known caveat that prevents the Change Password feature from working. You must download a patch from Microsoft at the following URL:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q326770

For more information on completing these tasks, see the section “PPP Configuration ” in the *Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T. The RADIUS server must be configured for authentication. Refer to vendor-specific documentation for information on configuring RADIUS authentication on the RADIUS server.

Restrictions for MSCHAP Version 2

- MSCHAP V2 authentication is not compatible with MSCHAP V1 authentication.
- The change password option is supported only for RADIUS authentication and is not available for local authentication.

Information About MSCHAP Version 2

MSCHAP V2 authentication is the default authentication method used by the Microsoft Windows 2000 operating system. Cisco routers that support this authentication method enable Microsoft Windows 2000 operating system users to establish remote PPP sessions without configuring an authentication method on the client.

MSCHAP V2 authentication introduced an additional feature not available with MSCHAP V1 or standard CHAP authentication: the Change Password feature. This features allows the client to change the account password if the RADIUS server reports that the password has expired.

Note MSCHAP V2 authentication is an updated version of MSCHAP that is similar to but incompatible with MSCHAP Version 1 (V1). MSCHAP V2 introduces mutual authentication between peers and a Change Password feature.

How to Configure MSCHAP Version 2

Configuring MSCHAP V2 Authentication

To configure the NAS to accept MSCHAP V2 authentication for local or RADIUS authentication and to allow proper interpretation of authentication failure attributes and vendor-specific RADIUS attributes for RADIUS authentication, use the following commands beginning in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server vsa send authentication**
4. **interface** *type number*
5. **ppp max-bad-auth** *number*
6. **ppp authentication ms-chap-v2**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server vsa send authentication** **Example:** `Router(config)# radius-server vsa send authentication`	Configures the NAS to recognize and use vendor-specific attributes.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 5	**ppp max-bad-auth** *number* **Example:** `Router(config-if)# ppp max-bad-auth 2`	Configures a point-to-point interface to reset immediately after an authentication failure or within a specified number of authentication retries. • The default value for the *number*argument is 0 seconds (immediately). • The range is between 0 and 255. **Note** The *number* argument must be set to a value of at least 2 for authentication failure attributes to be interpreted by the NAS.
Step 6	**ppp authentication ms-chap-v2** **Example:** `Router(config-if)# ppp authentication ms-chap-v2`	Enables MSCHAP V2 authentication on a NAS.
Step 7	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Verifying MSCHAP V2 Configuration

To verify that the MSCHAP Version 2 feature is configured properly, perform the following steps.

SUMMARY STEPS

1. **show running-config interface** *type number*
2. **debug ppp negotiation**
3. **debug ppp authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show running-config interface** *type number* **Example:** `Router# show running-config interface Asynch65`	Verifies the configuration of MSCHAP V2 as the authentication method for the specified interface.
Step 2	**debug ppp negotiation** **Example:** `Router# debug ppp negotiation`	Verifies successful MSCHAP V2 negotiation.
Step 3	**debug ppp authentication** **Example:** `Router# debug ppp authentication`	Verifies successful MSCHAP V2 authentication.

Configuring Password Aging for Crypto-Based Clients

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login**command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login**command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

After the RADIUS server requests a new password, AAA queries the crypto client, which in turn prompts the user to enter a new password.

To configure login authentication and password aging for crypto-based clients, use the following commands beginning in global configuration mode.

Note The AAA Password Expiry infrastructure notifies the Easy VPN client that the password has expired and provides a generic way for the user to change the password. Please use RADIUS-server domain-stripping feature wisely in combination with AAA password expiry support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} **passwd-expiry** *method1* [*method2...*]
5. **crypto map** *map-name* **client authentication list** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa authentication login** {**default** | *list-name*} **passwd-expiry** *method1* [*method2...*] **Example:** `Router(config)# aaa authentication login userauthen passwd-expiry group radius`	Enables password aging for crypto-based clients on a local authentication list.
Step 5	**crypto map** *map-name* **client authentication list** *list-name* **Example:** **Example:** `Router(config)# crypto map clientmap client authentication list userauthen`	Configures user authentication (a list of authentication methods) on an existing crypto map.

Configuration Examples

- Configuring Local Authentication Example, page 99
- Configuring RADIUS Authentication Example, page 99
- Configuring Password Aging with Crypto Authentication Example, page 99

Configuring Local Authentication Example

The following example configures PPP on an asynchronous interface and enables MSCHAP V2 authentication locally:

```
interface Async65
  ip address 10.0.0.2 255.0.0.0
  encapsulation ppp
  async mode dedicated
  no peer default ip address
  ppp max-bad-auth 3
  ppp authentication ms-chap-v2
  username client password secret
```

Configuring RADIUS Authentication Example

The following example configures PPP on an asynchronous interface and enables MSCHAP V2 authentication via RADIUS:

```
interface Async65
  ip address 10.0.0.2 255.0.0.0
  encapsulation ppp
  async mode dedicated
  no peer default ip address
  ppp max-bad-auth 3
  ppp authentication ms-chap-v2
  exit
aaa authentication ppp default group radius
 radius-server host 10.0.0.2 255.0.0.0
 radius-server key secret
 radius-server vsa send authentication
```

Configuring Password Aging with Crypto Authentication Example

The following example configures password aging by using AAA with a crypto-based client:

```
aaa authentication login userauthen passwd-expiry group radius
!
aaa session-id common
!
crypto isakmp policy 3
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group 3000client
 key cisco123
 dns 10.1.1.10
 wins 10.1.1.20
 domain cisco.com
 pool ippool
 acl 153
!
crypto ipsec transform-set myset esp-3des esp-sha-hmac
!
crypto dynamic-map dynmap 10
 set transform-set myset
!
crypto map clientmap client authentication list userauthen
!
radius-server host 10.140.15.203 auth-port 1645 acct-port 1646
radius-server domain-stripping prefix-delimiter $
radius-server key cisco123
radius-server vsa send authentication
```

```
radius-server vsa send authentication 3gpp2
!
end
```

Additional References

The following sections provide references related to the MSCHAP Version 2 feature.

Related Documents

Related Topic	Document Title
Configuring PPP interfaces	PPP Configuration in the *Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T.
Descriptions of the tasks and commands necessary to configure and maintain Cisco networking devices	*Cisco IOS Dial Technologies Command Reference*
Lists of IOS Security Commands	*Cisco IOS Security Command Reference*
Configuring PPP authentication using AAA	Configuring PPP Authentication Using AAA in the Configuring Authentication module in the *Cisco IOS Security Configuration Guide: Securing User Services* , Release 12.4T.
Configuring RADIUS Authentication	Configuring RADIUS module in the *Cisco IOS Security Configuration Guide: Securing User Services*, Release 12.4T.

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1661	*Point-to-Point Protocol (PPP)*

RFC	Title
RFC 2548	*Microsoft Vendor-specific RADIUS Attributes*
RFC 2759	*Microsoft PPP CHAP Extensions, Version 2*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MSCHAP Version 2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14 Feature Information for MSCHAP Version 2

Feature Name	Releases	Feature Information
MSCHAP Version 2	12.2(2)XB5 12.2(13)T 12.4(6)T	The MSCHAP Version 2 feature (introduced in Cisco IOS Release 12.2(2)XB5) allows Cisco routers to utilize Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAP V2) authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server (NAS). In 12.2(2)XB5, this feature was introduced. In 12.2(13)T, this feature was integrated into Cisco IOS Release 12.2(13)T. In 12.4(6)T, this feature was updated to include the crypto-based Password Aging feature. The following commands were introduced or modified: **aaa authentication login**, and **ppp authentication ms-chap-v2.**

RADIUS Packet of Disconnect

The RADIUS Packet of Disconnect feature is used to terminate a connected voice call.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for RADIUS Packet of Disconnect

- Configure AAA as described in *Cisco IOS Security Configuration Guide: Securing User Services* , Release 15.0(1)M.
- Use Cisco IOS Release 12.2(11)T or later.

Restrictions for RADIUS Packet of Disconnect

Proper matching identification information must be communicated by the following:

- Billing server and gateway configuration
- Gateway’s original accounting start request
- Server’s POD request

Information About RADIUS Packet of Disconnect

The Packet of Disconnect (POD) is a RADIUS access_request packet and is intended to be used in situations where the authenticating agent server wants to disconnect the user after the session has been accepted by the RADIUS access_accept packet.

When the POD is Needed

The POD may be needed in at least two situations:

- Detection of fraudulent use, which cannot be performed before accepting the call. A price structure so complex that the maximum session duration cannot be estimated before accepting the call. This may be the case when certain types of discounts are applied or when multiple users use the same subscription simultaneously.
- To prevent unauthorized servers from disconnecting users, the authorizing agent that issues the POD packet must include three parameters in its packet of disconnect request. For a call to be disconnected, all parameters must match their expected values at the gateway. If the parameters do not match, the gateway discards the packet of disconnect packet and sends a NACK (negative acknowledgement message) to the agent.

POD Parameters

The POD has the following parameters:

- An h323-conf-id vendor-specific attribute (VSA) with the same content as received from the gateway for this call.
- An h323-call-origin VSA with the same content as received from the gateway for the leg of interest.
- A 16-byte MD5 hash value that is carried in the *authentication* field of the POD request.
- Cisco allocated POD code 50 as the new code value for the Voice POD Request in Cisco IOS Release 12.2(27)SB and 12.4(15)T. This change was made because RFC 3576 *Dynamic Authorization Extensions to RADIUS* recently extended RADIUS standards to officially support both a Disconnect Message (DM) and Change-of-Authorization (CoA), which is supported through the POD.

RFC 3576 specifies the following POD codes:

-
  - 40 - Disconnect-Request
  - 41 - Disconnect-ACK
  - 42 - Disconnect-NAK
  - 43 - CoA-Request
  - 44 - CoA-ACK
  - 45 - CoA-NAK

How to Configure the RADIUS Packet of Disconnect

Configuring the RADIUS POD

Use the following tasks to configure the RADIUS POD:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router (config)# **aaa pod server** [**port** *port-number*] [**auth-type** {**any**| **all**| **session-key**}] **server-key** [*encryption-type*] *string*
4. Router# exit
5. Router# **show running-configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	Router (config)# **aaa pod server** [**port** *port-number*] [**auth-type** {**any**| **all**| **session-key**}] **server-key** [*encryption-type*] *string* **Example:** `Router(config)# aaa pod server server-key xyz123`	Enables inbound user sessions to be disconnected when specific session attributes are presented. • **port** *port-number* --(Optional) The network access server User Datagram Protocol (UDP) port to use for POD requests. Default value is 1700. • **auth-type** --(Optional) The type of authorization required for disconnecting sessions. • any--Session that matches all of the attributes sent in the POD packet is disconnected. The POD packet may contain one or more of four key attributes (user-name, framed-IP-address, session-ID, and session-key). • **all** --Only a session that matches all four key attributes is disconnected. **all** is the default. • **session-key** --Session with a matching session-key attribute is disconnected. All other attributes are ignored. • **server-key** --Configures the shared-secret text string. • *encryption-type* --(Optional) Single-digit number that defines whether the text immediately following is encrypted, and, if so, what type of encryption is used. Defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using an encryption algorithm defined by Cisco. • *string* --The shared-secret text string that is shared between the network access server and the client workstation. This shared-secret string must be the same on both systems.
Step 4	Router# exit	Exits global configuration mode.

Command or Action	Purpose

Step 5 Router# **show running-configuration**
 	
Example:
`Router# show running-configuration`
 	
Example:
`!`
 	
Example:
`aaa authentication login h323 group radius`
 	
Example:
`aaa authorization exec h323 group radius`
 	
Example:
`aaa accounting update newinfo`
 	
Example:
`aaa accounting connection h323 start-stop group radius`
 	
Example:
`aaa pod server server-key cisco`
 	
Example:
`aaa session-id common`
 	
Example:
`!`
	Verifies that the gateway is configured correctly in priveleged EXEC mode.

- Troubleshooting Tips, page 107

Troubleshooting Tips

Use the following tips to troubleshoot POD issues:

- Ensure that the POD port is configured correctly in both the gateway (using **aaa pod server**command) and the radius server. Both should be the same.
- Ensure that the shared-secret key configured in the gateway (using **aaa pod server**command) and in the AAA server are the same.
- Turn on **debug aaa pod** command to see what's going on. This will let you know if the gateway receives the POD packet from the server and if so, it will display any errors encountered.

The following example shows output from a successful POD request, when using the **show debug** command.

```
Router# debug aaa podAAA POD packet processing debugging is on
Router# show debugGeneral OS:
  AAA POD packet processing debugging is on
Router#
Apr 25 17:15:59.318:POD:172.19.139.206 request queued
Apr 25 17:15:59.318:voice_pod_request:
Apr 25 17:15:59.318:voip_populate_pod_attr_list:
Apr 25 17:15:59.318:voip_pod_get_guid:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=50
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr=h323-conf-id
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=50 value_len=35
Apr 25 17:15:59.318:voip_pod_get_guid:conf-id=FFA7785F F7F607BB
00000000 993FB1F4 n_bytes=35
Apr 25 17:15:59.318:voip_pod_get_guid:GUID = FFA7785F F7F607BB 00000000
993FB1F4
Apr 25 17:15:59.318:voip_populate_pod_attr_list:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=23
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr=h323-originate
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=23 value_len=6
Apr 25 17:15:59.318:voip_get_call_direction:
Apr 25 17:15:59.318:voip_get_call_direction:returning answer
Apr 25 17:15:59.318:voip_eval_pod_attr:
Apr 25 17:15:59.318:cc_api_trigger_disconnect:
Apr 25 17:15:59.322:POD:Sending ACK to 172.19.139.206/1700
Apr 25 17:15:59.322:voip_pod_clean:
```

Additional References

The following sections provide references related to the RADIUS Packet of Disconnect feature.

Related Documents

Related Topic	Document Title
AAA	*Cisco IOS Security Configuration Guide: Securing User Services*, Release 15.0(1)M
Security commands	*Cisco IOS Security Command Reference*
CLI Configuration	*Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4T
Configuring AAA for voice gateways	*Configuring AAA for Cisco Voice Gateways*, Release 12.4T

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	*Remote Authentication Dial-in User Service*
RFC 3576	*Dynamic Authorization Extensions to RADIUS*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS Packet of Disconnect

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15 Feature Information for RADIUS Packet of Disconnect

Feature Name	Releases	Feature Information
RADIUS Packet of Disconnect	12.2(2)XB 12.1(2)XH 12.3(11)T 12.2(27)SB 12.4(15)T	The RADIUS Packet of Disconnect feature is used to terminate a connected voice call. In Cisco IOS Release12.2(2)XB, this feature was introduced on the Cisco 3600, Cisco 5350, and Cisco 5400. In Cisco IOS Release 12.1(2)XH and 12.1(3)T, this feature was introduced on the Cisco 5300 and Cisco 5800. In Cisco IOS Release 12.2(11)T, this feature was introduced on the Cisco 5400, Cisco 5850 In Cisco IOS Release 12.2(27)SB and 12.4(15)T, Cisco allocated POD code 50 as the new code value for the voice POD request The following commands were introduced or modified: **aaa pod server** and **debug aaa pod**

Glossary

AAA --authentication, authorization, and accounting.

NACK --negative acknowledgement message.

POD --packet of disconnect. An access_reject packet sent from a RADIUS server to the gateway in order to disconnect a call which has been connected already. After validation of the packet, the gateway disconnects the user. The packet contains the information to disconnect the call.

POD server--a Cisco gateway configured to accept and process POD requests from a RADIUS authentication/authorization agent.

RADIUS --Remote Authentication Dial-In User Service. An authentication and accounting system used by many Internet service providers.

UDP --User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

VoIP-- voice over IP. The ability to carry normal telephony-style voice over an IP-based Internet with POTS-like functionality, reliability, and voice quality. VoIP is a blanket term that generally refers to the Cisco standards-based (for example, H.323) approach to IP voice traffic.

VSA --vendor-specific attribute.

Configuring MAC Authentication Bypass

The MAC Authentication Bypass feature is a MAC-address-based authentication mechanism that allows clients in a network to integrate with the Cisco Identity Based Networking Services (IBNS) and Network Admission Control (NAC) strategy using the client MAC address. The MAC Authentication Bypass feature is applicable to the following network environments:

- Network environments in which a supplicant code is not available for a given client platform.
- Network environments in which the end client configuration is not under administrative control, that is, the IEEE 802.1X requests are not supported on these networks.

Standalone MAC Authentication Bypass (MAB) is an authentication method that grants network access to specific MAC addresses regardless of 802.1X capability or credentials. As a result, devices such as cash registers, fax machines, and printers can be readily authenticated, and network features that are based on authorization policies can be made available.

Before standalone MAB support was available, MAB could be configured only as a failover method for 802.1x authentication. Standalone MAB is independent of 802.1x authentication.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring MAC Authentication Bypass

IEEE 802.1x--Port-Based Network Access Control

You should understand the concepts of port-based network access control and have an understanding of how to configure port-based network access control on your Cisco platform. See the *Cisco IOS Security Configuration Guide: Securing User Services* , Release 15.0, for more information.

RADIUS and ACLs

You should understand the concepts of the RADIUS protocol and have an understanding of how to create and apply access control lists (ACLs). For more information, see the documentation for your Cisco platform and the *Cisco IOS Security Configuration Guide: Securing User Services* , Release 15.0.

The switch must have a RADIUS configuration and be connected to the Cisco secure access control server (ACS). For more information, see the *User Guide for Secure ACS Appliance 3.2* .

Information About Configuring MAC Authentication Bypass

Overview of the Cisco IOS Auth Manager

The capabilities of devices connecting to a given network can be different, thus requiring that the network support different authentication methods and authorization policies. The Cisco IOS Auth Manager handles network authentication requests and enforces authorization policies regardless of authentication method. The Auth Manager maintains operational data for all port-based network connection attempts, authentications, authorizations, and disconnections and, as such, serves as a session manager.

The possible states for Auth Manager sessions are as follows:

- Idle--In the idle state, the authentication session has been initialized, but no methods have yet been run. This is an intermediate state.
- Running--A method is currently running. This is an intermediate state.
- Authc Success--The authentication method has run successfully. This is an intermediate state.
- Authc Failed--The authentication method has failed. This is an intermediate state.
- Authz Success--All features have been successfully applied for this session. This is a terminal state.
- Authz Failed--At least one feature has failed to be applied for this session. This is a terminal state.
- No methods--No method provided a result for this session. This is a terminal state.

Standalone MAB

MAB uses the MAC address of the connecting device to grant or deny network access. To support MAB, the RADIUS authentication server maintains a database of MAC addresses for devices that require access to the network. MAB generates a RADIUS request with a MAC address in the Calling-Station-Id (attribute 31) and Service-Type (attribute 6) with a value of 10. After a successful authentication, the Auth Manager enables various authorization features specified by the authorization policy, such as ACL assignment and VLAN assignment.

How to Configure Configuring MAC Authentication Bypass

Enabling MAC Authentication Bypass

Perform this task to enable the MAC Authentication Bypass feature on an 802.1X port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **dot1x mac-auth-bypass** [**eap**]
5. **end**
6. **show dot1x interface** *type slot / port* **details**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Router(config)# interface FastEthernet 2/1`	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	**dot1x mac-auth-bypass** [**eap**] **Example:** `Router(config-if)# dot1x mac-auth-bypass`	Enables the MAC Authentication Bypass (MAB) feature on an 802.1X Port.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privilege EXEC mode.
Step 6	**show dot1x interface** *type slot* / *port* **details** **Example:** `Router# show dot1x interface FastEthernet 2/1 details`	Displays the interface configuration and the authenticator instances on the interface.

Enabling Standalone MAB

Ports enabled with the Standalone MAB feature can use the MAC address of connecting devices to grant or deny network access. Perform the steps described in this section to enable standalone MAB on individual ports.

Before you can configure standalone MAB, the switch must be connected to a Cisco Secure ACS server and RADIUS authentication, authorization, and accounting (AAA) must be configured.

Note Standalone MAB can be configured on switched ports only--it cannot be configured on routed ports.

Note If you are unsure whether MAB or MAB EAP is enabled or disabled on the switched port, use t
mabor **default mab eap** commands in interface configuration mode to configure MAB or MAI
default.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 5	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.

	Command or Action	Purpose
Step 6	**authentication port-control auto** **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	Enables MAB.
Step 8	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

- Troubleshooting Tips, page 118

Troubleshooting Tips

The following commands can help troubleshoot standalone MAB:

- **debug authentication**
- **debug mab all**
- **show authentication registrations**
- **show authentication sessions**
- **show mab**

Enabling Reauthentication on a Port

By default, ports are not automatically reauthenticated. You can enable automatic reauthentication and specify how often reauthentication attempts are made.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **authentication periodic**
9. **authentication timer reauthenticate** {*seconds* | **server**}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 5	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.

	Command or Action	Purpose
Step 6	**authentication port-control auto** **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	Enables MAB.
Step 8	**authentication periodic** **Example:** `Switch(config-if)# authentication periodic`	Enables reauthentication.
Step 9	**authentication timer reauthenticate** {*seconds* | **server**} **Example:** `Switch(config-if)# authentication timer reauthenticate 900`	Configures the time, in seconds, between reauthentication attempts.
Step 10	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Specifying the Security Violation Mode

When there is a security violation on a port, the port can be shut down or traffic can be restricted. By default, the port is shut down. You can configure the period of time for which the port is shut down.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **authentication violation** {**restrict** | **shutdown**}
9. **authentication timer restart** *seconds*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 5	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.

	Command or Action	Purpose
Step 6	**authentication port-control auto** **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	Enables MAB.
Step 8	**authentication violation** {**restrict** | **shutdown**} **Example:** `Switch(config-if)# authentication violation shutdown`	Configures the action to be taken when a security violation occurs on the port.
Step 9	**authentication timer restart** *seconds* **Example:** `Switch(config-if)# authentication timer restart 30`	Configures the period of time, in seconds, after which an attempt is made to authenticate an unauthorized port.
Step 10	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Configuring MAC Authentication Bypass

Example Standalone MAB Configuration

The following example shows how to configure standalone MAB on a port. In this example, the client is reauthenticated every 1200 seconds and the connection is dropped after 600 seconds of inactivity.

```
enable
 configure terminal
  interface GigabitEthernet2/1
   switchport
```

```
switchport mode access
switchport access vlan 2
authentication port-control auto
mab
authentication violation shutdown
authentication timer restart 30
authentication periodic
authentication timer reauthenticate 1200
authentication timer inactivity 600
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Authentication commands	*Cisco IOS Security Command Reference*
IEEE 802.1x--Flexible Authentication	*Cisco IOS Security Configuration Guide: Securing User Services*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
• CISCO-AUTH-FRAMEWORK-MIB • CISCO-MAC-AUTH-BYPASS-MIB • CISCO-PAE-MIB • IEEE8021-PAE-MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3580	*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring MAC Authentication Bypass

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16 Feature Information for Configuring MAC Authentication Bypass

Feature Name	Releases	Feature Information
MAC Authentication Bypass (MAB)	12.1(22)T 12.2(31)SG 12.2(33)SXH15.1(4)M	The MAC Authentication Bypass feature is a MAC-address-based authentication mechanism that allows clients in a network to integrate with the Cisco IBNS and NAC strategy using the client MAC address. In Cisco IOS Release 15.1(4)M support was extended for Integrated Services Router Generation 2 (ISR G2) platforms. The following commands were introduced or modified: **dot1x mac-auth-bypass**, **show dot1x interface**.

Feature Name	Releases	Feature Information
Standalone MAB Support	12.2(33)SXI , 15.2(2)T	This feature grants network access to devices based on MAC address regardless of 802.1x capability or credentials. The following commands were introduced or modified: **authentication periodic**, **authentication port-control**, **authentication timer inactivity**, **authentication timer reauthenticate**, **authentication timer restart**, **authentication violation**, **debug authentication**, **mab**, **show authentication interface**, **show mab**, **show authentication registrations**, **show authentication sessions**.

Configuring Authorization

The AAA authorization feature is used to determine what a user can and cannot do. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user is granted access to a requested service only if the information in the user profile allows it.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites

Before configuring authorization using named method lists, the following tasks must be performed:

- Enable AAA on your network access server.
- Configure AAA authentication. Authorization generally takes place after authentication and relies on authentication to work properly.
- Define the characteristics of your Lightweight Directory Access Protocol (LDAP), RADIUS, or TACACS+ security server if RADIUS or TACACS+ authorization is issued so that the Cisco network access server can communicate with the RADIUS or TACACS+ security server.
- Define the rights associated with specific users by using the **username** command if local authorization is issued.
- See the Related Documents section for more information on documents related to these prerequisites.

Information About Configuring Authorization

Named Method Lists for Authorization

Method lists for authorization define the ways that authorization is performed and the sequence in which these methods are performed. A method list is simply a named list describing the authorization methods to be queried (such as LDAP, RADIUS, or TACACS+), in sequence. Method lists enable one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method, or all methods defined are exhausted.

Note The Cisco IOS software attempts authorization with the next listed method only when there is no response from the previous method. If authorization fails at any point in this cycle--meaning that the security server or local username database responds by denying the user services--the authorization process stops and no other authorization methods are attempted.

Method lists are specific to the authorization type requested:

- **Auth-proxy** --Applies specific security policies on a per-user basis.
- **Commands** --Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **EXEC** --Applies to the attributes associated with a user EXEC terminal session.
- **Network** --Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- **Reverse Access** --Applies to reverse Telnet sessions.

When a named method list is created, a particular list of authorization methods for the indicated authorization type is defined.

Once defined, method lists must be applied to specific lines or interfaces before any of the defined methods are performed. The only exception is the default method list (which is named “default”). If the **aaa authorization** command for a particular authorization type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, local authorization takes place by default.

AAA Authorization Methods

AAA supports five different methods of authorization:

- TACACS+--The network access server exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating attribute-value pairs, which are stored in a database on the TACACS+ security server, with the appropriate user.
- If-Authenticated--The user is allowed to access the requested function provided the user has been authenticated successfully.
- **None** --The network access server does not request authorization information; authorization is not performed over this line/interface.
- Local--The router or access server consults its local database, as defined by the **username** command, for example, to authorize specific rights for users. Only a limited set of functions can be controlled through the local database.
- LDAP--The network access server requests authorization information from the RADIUS security server. LDAP authorization defines specific rights for users by associating attributes, which are stored in a database on the LDAP server, with the appropriate user.
- RADIUS--The network access server requests authorization information from the RADIUS security server. RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user.

- Authorization Methods, page 129

Authorization Methods

To have the network access server request authorization information through a TACACS+ security server, use the **aaa authorization** command with the **group tacacs+** *method* keyword. For more specific information about configuring authorization using a TACACS+ security server, see the Configuring TACACS+ feature module. For an example of how to enable a TACACS+ server to authorize the use of network services, including PPP and ARA, see the TACACS Authorization Examples for more information.

To allow users to have access to the functions they request as long as they have been authenticated, use the **aaa authorization** command with the **if-authenticated** *method* keyword. If this method is selected, all requested functions are automatically granted to authenticated users.

There may be times when it is not desirable to run authorization from a particular interface or line. To stop authorization activities on designated lines or interfaces, use the **none** *method* keyword. If this method is selected, authorization is disabled for all actions.

To select local authorization, which means that the router or access server consults its local user database to determine the functions a user is permitted to use, use the **aaa authorization** command with the **local** *method* keyword. The functions associated with local authorization are defined by using the **username** global configuration command. For a list of permitted functions, see the Configuring Authentication feature module.

To have the network access server request authorization through a LDAP security server, use the **ldap** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, see the Configuring RADIUS feature module

To have the network access server request authorization through a RADIUS security server, use the **radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, see the Configuring RADIUS feature module.

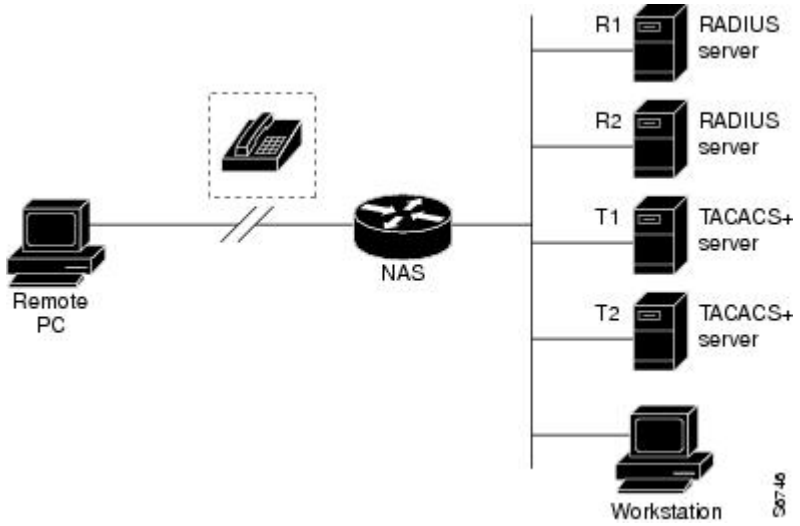
To have the network access server request authorization through a RADIUS security server, use the **aaa authorization** command with the **group radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, see the Configuring RADIUS feature module. For an example of how to enable a RADIUS server to authorize services, see the RADIUS Authorization Example for more information.

Note Authorization method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for authorization applies.

Method Lists and Server Groups

A server group is a way to group existing LDAP, RADIUS, or TACACS+ server hosts for use in method lists. The figure below shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Figure 4 Typical AAA Network Configuration



Using server groups, a subset of the configured server hosts can be specified and use them for a particular service. For example, server groups allows R1 and R2 to be defined as separate server groups, and T1 and T2 as separate server groups. This allows either R1 and T1 to be specified in the method list or R2 and T2 in the method list, which provides more flexibility in the way that RADIUS and TACACS+ resources are assigned.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service--for example, authorization--the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers. See the Configuring LDAP, Configuring RADIUS or Configuring TACACS+ feature modules.

AAA Authorization Types

Cisco IOS software supports five different types of authorization:

- **Auth-proxy** --Applies specific security policies on a per-user basis. See the Configuring Authentication Proxy sectionfor more information about where to find authentication proxy configuration documentation.
- **Commands** --Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **EXEC** --Applies to the attributes associated with a user EXEC terminal session.
- **Network** --Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- **Reverse Access** --Applies to reverse Telnet sessions.
- Configuration--Applies to downloading configurations from the AAA server.
- IP Mobile--Applies to authorization for IP mobile services.

- Authorization Types, page 131

Authorization Types

Named authorization method lists are specific to the indicated type of authorization.

To create a method list to enable authorization that applies specific security policies on a per-user basis, use the **auth-proxy** keyword.

To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARAP), use the **network** keyword.

To create a method list to enable authorization to determine if a user is allowed to run an EXEC shell, use the **exec** keyword.

To create a method list to enable authorization for specific, individual EXEC commands associated with a specific privilege level, use the **commands**keyword. (This allows all commands associated with a specified command level from 0 to 15 to be authorized.)

To create a method list to enable authorization for reverse Telnet functions, use the **reverse-access** keyword.

Authorization Attribute-Value Pairs

RADIUS and TACACS+ authorization both define specific rights for users by processing attributes, which are stored in a database on the security server. For both RADIUS and TACACS+, attributes are defined on the security server, associated with the user, and sent to the network access server where they are applied to the user's connection.

See the RADIUS Attributes and TACACS+ Attribute-Value Pairs sections for more information about supported RADIUS attributes and TACACS+ attribute-value pair documentation.

How to Configure Authorization

See Authorization Configuration Examples for more information.

- Configuring AAA Authorization Using Named Method Lists, page 132

Configuring AAA Authorization Using Named Method Lists

Perform this task to configure AAA authorization using named method lists:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access** | **configuration** | **ipmobile**} {**default** | *list-name*} [*method1* [*method2*...]]
4. Do one of the following:
   - **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
   -
   -
   - **interface** *interface-type interface-number*
5. Do one of the following:
   - **authorization** {**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*}
   -
   -
   - **ppp authorization** {**default** | *list-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access** | **configuration** | **ipmobile**} {**default** | *list-name*} [*method1* [*method2*...]] **Example:** `Router(config)# aaa authorization auth-proxy default`	Creates an authorization method list for a particular authorization type and enable authorization.

Command or Action	Purpose

Step 4 Do one of the following:
• **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
•
•
• **interface** *interface-type interface-number*
 	
Example:
`Router(config)# line aux 0`
 	
Example:
 	
Example:
 	
Example:
`Router(config)# interface interface-type interface-number`
	Enters the line configuration mode for the lines to which the authorization method list is applied.
	Alternately, enters the interface configuration mode for the interfaces to which the authorization method list is applied.

Command or Action	Purpose

Step 5 Do one of the following:
• **authorization** {**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*}
•
•
• **ppp authorization** {**default** | *list-name*}
 	
Example:
`Router(config-line)# authorization arap default`
 	
Example:
 	
Example:
 	
Example:
`Router(config-line)# ppp authorization default`
	Applies the authorization list to a line or set of lines.
	Or
	Applies the authorization list to an interface or set of interfaces.

Disabling Authorization for Global Configuration Commands

The **aaa authorization** command with the keyword **commands** attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level. Because there are configuration commands that are identical to some EXEC-level commands, there can be some confusion in the authorization process. Using **no aaa authorization config-commands** stops the network access server from attempting configuration command authorization.

To disable AAA authorization for all global configuration commands, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **no aaa authorization config-commands**	Disables authorization for all global configuration commands.

Configuring Authorization for Reverse Telnet

Telnet is a standard terminal emulation protocol used for remote terminal connection. Normally, a network access server is logged into and then Telnet is used to access other network devices from that network access server. There are times, however, when it is necessary to establish a reverse Telnet session. In reverse Telnet sessions, the Telnet connection is established in the opposite direction--from inside a

network to a network access server on the network periphery to gain access to modems or other devices connected to that network access server. Reverse Telnet is used to provide users with dialout capability by allowing them to Telnet to modem ports attached to a network access server.

It is important to control access to ports accessible through reverse Telnet. Failure to do so could, for example, allow unauthorized users free access to modems where they can trap and divert incoming calls or make outgoing calls to unauthorized destinations.

Authentication during reverse Telnet is performed through the standard AAA login procedure for Telnet. Typically the user has to provide a username and password to establish either a Telnet or reverse Telnet session. Reverse Telnet authorization provides an additional (optional) level of security by requiring authorization in addition to authentication. When enabled, reverse Telnet authorization can use RADIUS or TACACS+ to authorize whether or not this user is allowed reverse Telnet access to specific asynchronous ports, after the user successfully authenticates through the standard Telnet login procedure.

Reverse Telnet authorization offers the following benefits:

- An additional level of protection by ensuring that users engaged in reverse Telnet activities are indeed authorized to access a specific asynchronous port using reverse Telnet.
- An alternative method (other than access lists) to manage reverse Telnet authorization.

To configure a network access server to request authorization information from a TACACS+ or RADIUS server before allowing a user to establish a reverse Telnet session, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa authorization reverse-access** *method1* `[`*method2* ...`]`	Configures the network access server to request authorization information before allowing a user to establish a reverse Telnet session.

This feature enables the network access server to request reverse Telnet authorization information from the security server, whether RADIUS or TACACS+. The specific reverse Telnet privileges for the user on the security server itself must be configured.

Authorization Configuration Examples

Named Method List Configuration Example

The following example shows how to configure a Cisco AS5300 (enabled for AAA and communication with a RADIUS security server) for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database is queried for authentication and authorization information, and accounting services are handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
```

```
aaa authentication ppp dialins group radius local
aaa authorization network scoobee group radius local
aaa accounting network charley start-stop group radius
username root password ALongPassword
radius-server host alcatraz
radius-server key myRaDiUSpassWoRd
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization scoobee
 ppp accounting charley
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, admins, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.
- The **aaa authorization network scoobee group radius local** command defines the network authorization method list named scoobee, which specifies that RADIUS authorization is used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization is performed.
- The **aaa accounting network charley start-stop group radius**command defines the network accounting method list named charley, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) are used on serial lines using PPP.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins**command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **ppp authorization scoobee**command applies the scoobee network authorization method list to the specified interfaces.
- The **ppp accounting charley**command applies the charley network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.

- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

TACACS Authorization Examples

The following examples show how to use a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or an error occurs during the authorization process, the fallback method (none) is to grant all authorization requests:

```
aaa authorization network default group tacacs+ none
```

The following example shows how to allow network authorization using TACACS+:

```
aaa authorization network default group tacacs+
```

The following example shows how to provide the same authorization, but it also creates address pools called mci and att:

```
aaa authorization network default group tacacs+
ip address-pool local
ip local-pool mci 172.16.0.1 172.16.0.255
ip local-pool att 172.17.0.1 172.17.0.255
```

These address pools can then be selected by the TACACS daemon. A sample configuration of the daemon follows:

```
user = mci_customer1 {
    login = cleartext “some password”
    service = ppp protocol = ip {
        addr-pool=mci
    }
}
user = att_customer1 {
    login = cleartext “some other password”
    service = ppp protocol = ip {
        addr-pool=att
     }
```

RADIUS Authorization Example

The following example shows how to configure the router to authorize using RADIUS:

```
aaa new-model
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
radius-server host ip
radius-server key
```

The lines in this sample RADIUS authorization configuration are defined as follows:

- The **aaa authorization exec default group radius if-authenticated** command configures the network access server to contact the RADIUS server to determine if users are permitted to start an EXEC shell when they log in. If an error occurs when the network access server contacts the RADIUS server, the fallback method is to permit the CLI to start, provided the user has been properly authenticated.

The RADIUS information returned may be used to specify an autocommand or a connection access list be applied to this connection.

- The **aaa authorization network default group radius** command configures network authorization through RADIUS. This can be used to govern address assignment, the application of access lists, and various other per-user quantities.

Note Since no fallback method is specified in this example, authorization fails if, for any reason, there is no response from the RADIUS server.

LDAP Authorization Example

The following example shows how to configure the router to authorize using LDAP:

```
aaa new-model
aaa authorization exec default group ldap if-authenticated
aaa authorization network default group ldap
```

The lines in this sample RADIUS authorization configuration are defined as follows:

- The **aaa authorization exec default group ldap if-authenticated** command configures the network access server to contact the LDAP server to determine if users are permitted to start an EXEC shell when they log in. If an error occurs when the network access server contacts the LDAP server, the fallback method is to permit the CLI to start, provided the user has been properly authenticated.

The LDAP information returned may be used to specify an autocommand or a connection access list be applied to this connection.

The **aaa authorization network default group ldap** command configures network authorization through LDAP. This command can be used to govern address assignment, the application of access lists, and various other per-user quantities.

Reverse Telnet Authorization Examples

The following examples show how to cause the network access server to request authorization information from a TACACS+ security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization reverse-access default group tacacs+
!
tacacs-server host 172.31.255.0
tacacs-server timeout 90
tacacs-server key goaway
```

The lines in this sample TACACS+ reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group tacacs+** command specifies TACACS+ as the default method for user authentication during login.
- The **aaa authorization reverse-access default group tacacs+** command specifies TACACS+ as the method for user authorization when trying to establish a reverse Telnet session.
- The **tacacs-server host** command identifies the TACACS+ server.
- The **tacacs-server timeout**command sets the interval of time that the network access server waits for the TACACS+ server to reply.
- The **tacacs-server key** command defines the encryption key used for all TACACS+ communications between the network access server and the TACACS+ daemon.

The following example shows how to configure a generic TACACS+ server to grant a user, pat, reverse Telnet access to port tty2 on the network access server named “maple” and to port tty5 on the network access server named “oak”:

```
user = pat
  login = cleartext lab
  service = raccess {
    port#1 = maple/tty2
    port#2 = oak/tty5
```

Note In this example, “maple” and “oak” are the configured host names of network access servers, not DNS names or alias.

The following example shows how to configure the TACACS+ server (CiscoSecure) to grant a user named pat reverse Telnet access:

```
user = pat
profile_id = 90
profile_cycle = 1
member = Tacacs_Users
service=shell {
default cmd=permit
}
service=raccess {
allow “c2511e0” “tty1” “.*”
refuse “.*” “.*” “.*”
password = clear “goaway”
```

Note CiscoSecure only supports reverse Telnet using the command line interface in versions 2.1(*x*) through version 2.2(1).

An empty “service=raccess {}” clause permits a user to have unconditional access to network access server ports for reverse Telnet. If no “service=raccess” clause exists, the user is denied access to any port for reverse Telnet.

The following example shows how to cause the network access server to request authorization from a RADIUS security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group radius
aaa authorization reverse-access default group radius
!
radius-server host 172.31.255.0
radius-server key go away
auth-port 1645 acct-port 1646
```

The lines in this sample RADIUS reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group radius** command specifies RADIUS as the default method for user authentication during login.
- The **aaa authorization reverse-access default group radius** command specifies RADIUS as the method for user authorization when trying to establish a reverse Telnet session.
- The **radius-server host** command identifies the RADIUS server.
- The **radius-server key** command defines the encryption key used for all RADIUS communications between the network access server and the RADIUS daemon.

The following example shows how to send a request to the RADIUS server to grant a user named “pat” reverse Telnet access at port tty2 on the network access server named “maple”:

```
Username = “pat”
Password = “goaway”
User-Service-Type = Shell-User
cisco-avpair = “raccess:port#1=maple/tty2”
```

The syntax "raccess:port=any/any" permits a user to have unconditional access to network access server ports for reverse Telnet. If no "raccess:port={*nasname* }/{*tty number* }" clause exists in the user profile, the user is denied access to reverse Telnet on all ports.

Additional References

The following sections provide references related to the Authorization feature.

Related Documents

Related Topic	Document Title
Authorization Commands	*Cisco IOS Security Command Reference*
RADIUS	Configuring RADIUS feature module.
LDAP	Configuring RADIUS feature Module.
RADIUS attributes	RADIUS Attributes Overview and RADIUS IETF Attributes feature module.
TACACS+	Configuring TACACS+ feature module.
TACACS+ Attribute-Value Pairs	TACACS+ Attribute-Value Pairs feature module.
Authentication	Configuring Authentication feature module.
Authentication Proxy	Configuring Authentication Proxy feature module.

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Authorization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 17 Feature Information for Configuring Authorization

Feature Name	Releases	Feature Information
Configuring Authorization	10.0 Cisco IOS XE Release 2.1	The AAA authorization feature is used to determine what a user can and cannot do. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user is granted access to a requested service only if the information in the user profile allows it. This feature was introduced in Cisco IOS Release 10.0. This feature was introduced on Cisco ASR 1000 Series Routers.
LDAP integration with Active Directory	15.1(1)T	LDAP is a standard-based protocol used to access directories. It is based on client server model similar to RADIUS. LDAP is deployed on Cisco devices to send authentication requests to a central LDAP server that contains all user authentication and network service access information. This feature provides authentication and authorization support for AAA. The following command was modified: **aaa authorization**

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configuring Accounting

The AAA Accounting feature allows the services that users are accessing and the amount of network resources that users are consuming to be tracked. When AAA Accounting is enabled, the network access server reports user activity to the TACACS+ or RADIUS security server (depending on which security method is implemented) in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, and auditing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Accounting

The following tasks must be performed before configuring accounting using named method lists:

- Enable AAA on the network access server by using the **aaa new-model**command in global configuration mode.
- Define the characteristics of the RADIUS or TACACS+ security server if RADIUS or TACACS+ authorization is issued. For more information about configuring the Cisco network access server to communicate with the RADIUS security server, see the Configuring RADIUS module. For more information about configuring the Cisco network access server to communicate with the TACACS+ security server, see the Configuring TACACS+ module.

Restrictions for Configuring Accounting

- Accounting information can be sent simultaneously to a maximum of only four AAA servers.
- For Service Selection Gateway (SSG) systems, the **aaa accounting network broadcast** command broadcasts only **start-stop** accounting records. If interim accounting records are configured using the **ssg accounting interval** command, the interim accounting records are sent only to the configured default RADIUS server.

Information About Configuring Accounting

Named Method Lists for Accounting

Similar to authentication and authorization method lists, method lists for accounting define the way accounting is performed and the sequence in which these methods are performed.

Named accounting method lists allow particular security protocol to be designated and used on specific lines or interfaces for accounting services. The only exception is the default method list (which is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is simply a named list describing the accounting methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists allow one or more security protocols to be designated and used for accounting, thus ensuring a backup system for accounting in case the initial method fails. Cisco IOS software uses the first method listed to support accounting; if that method fails to respond, the Cisco IOS software selects the next accounting method listed in the method list. This process continues until there is successful communication with a listed accounting method, or all methods defined are exhausted.

Note The Cisco IOS software attempts accounting with the next listed accounting method only when there is no response from the previous method. If accounting fails at any point in this cycle--meaning that the security server responds by denying the user access--the accounting process stops and no other accounting methods are attempted.

Accounting method lists are specific to the type of accounting being requested. AAA supports seven different types of accounting:

- **Network** --Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.
- **EXEC** --Provides information about user EXEC terminal sessions of the network access server.
- **Commands** --Provides information about the EXEC mode commands that a user issues. Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.

- **Connection** --Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.
- **System** --Provides information about system-level events.
- **Resource** --Provides "start" and "stop" records for calls that have passed user authentication, and provides "stop" records for calls that fail to authenticate.
- **VRRS** --Provides information about Virtual Router Redundancy Service (VRRS).

Note System accounting does not use named accounting lists; only the default list for system accounting can be defined.

Once again, when a named method list is created, a particular list of accounting methods for the indicated accounting type are defined.

Accounting method lists must be applied to specific lines or interfaces before any of the defined methods are performed. The only exception is the default method list (which is named "default"). If the **aaa accounting** command for a particular accounting type is issued without specifying a named method list, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined (A defined method list overrides the default method list). If no default method list is defined, then no accounting takes place.

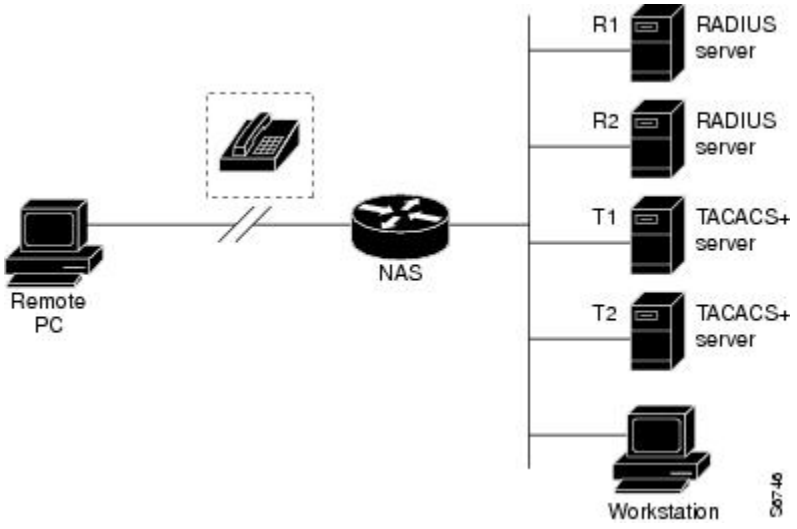
This section includes the following subsections:

- Method Lists and Server Groups, page 147
- AAA Accounting Methods, page 148
- Accounting Record Types, page 148
- Accounting Methods, page 148

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. The figure below shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers.

Figure 5 Typical AAA Network Configuration



Cisco IOS software, RADIUS and TACACS+ server configurations are global. A subset of the configured server hosts can be specified using server groups. These server groups can be used for a particular service. For example, server groups allow R1 and R2 to be defined as separate server groups (SG1 and SG2), and T1 and T2 as separate server groups (SG3 and SG4). This means either R1 and T1 (SG1 and SG3) or R2 and T2 (SG2 and SG4) can be specified in the method list, which provides more flexibility in the way that RADIUS and TACACS+ resources are assigned.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server from the same IP address. If two different host entries on the same RADIUS server are configured for the same service--for example, accounting--the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services (The RADIUS host entries are tried in the order in which they are configured).

For more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers, see the “Configuring RADIUS” or “Configuring TACACS+” module in the Cisco IOS Security Configuration Guide: Securing User Services .

AAA Accounting Methods

The Cisco IOS software supports the following two methods for accounting:

- TACACS+--The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting AV pairs and is stored on the security server.
- RADIUS--The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting AV pairs and is stored on the security server.

Accounting Record Types

For minimal accounting, use the **stop-only** keyword, which instructs the specified method (**RADIUS** or **TACACS+**) to send a stop record accounting notice at the end of the requested user process. For more accounting information, use the **start-stop** keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event. To stop all accounting activities on this line or interface, use the **none** keyword.

Accounting Methods

The table below lists the supported accounting methods.

Table 18 AAA Accounting Methods

Keyword	Description
group radius	Uses the list of all RADIUS servers for accounting.
group tacacs+	Uses the list of all TACACS+ servers for accounting.

Keyword	Description
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server group *group-name*.

The method argument refers to the actual method the authentication algorithm tries. Additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all other methods return an error, specify additional methods in the command. For example, to create a method list named acct_tac1 that specifies RADIUS as the backup method of authentication in the event that TACACS+ authentication returns an error, enter the following command:

```
aaa accounting network acct_tac1 stop-only group tacacs+ group radius
```

To create a default list that is used when a named list is not specified in the **aaa accounting**command, use the **default** keyword followed by the methods that are wanted to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa accounting network default stop-only group radius
```

AAA Accounting supports the following methods:

- **group tacacs** --To have the network access server send accounting information to a TACACS+ security server, use the **group tacacs+** *method* keyword.
- **group radius** --To have the network access server send accounting information to a RADIUS security server, use the **group radius** *method* keyword.

Note Accounting method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for accounting applies.

- **group** *group-name* --To specify a subset of RADIUS or TACACS+ servers to use as the accounting method, use the **aaa accounting**command with the **group** *group-name* method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the **group loginrad**.

To specify **group loginrad** as the method of network accounting when no other method list has been defined, enter the following command:

```
aaa accounting network default start-stop group loginrad
```

Before a group name can be used as the accounting method, communication with the RADIUS or TACACS+ security server must be enabled.

AAA Accounting Types

Network Accounting

Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through an EXEC session:

```
Wed Jun 27 04:44:45 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “0000000D”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:45:00 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000E”
        Framed-IP-Address = “10.1.1.2”
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
Wed Jun 27 04:47:46 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000E”
        Framed-IP-Address = “10.1.1.2”
        Framed-Protocol = PPP
        Acct-Input-Octets = 3075
        Acct-Output-Octets = 167
        Acct-Input-Packets = 39
        Acct-Output-Packets = 9
```

```
        Acct-Session-Time = 171
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
Wed Jun 27 04:48:45 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “408”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “0000000D”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who first started an EXEC session:

```
Wed Jun 27 04:00:35 2001 172.16.25.15    username1   tty4    562/4327528
starttask_id=28      service=shell
Wed Jun 27 04:00:46 2001 172.16.25.15    username1   tty4 562/4327528
starttask_id=30      addr=10.1.1.1   service=ppp
Wed Jun 27 04:00:49 2001 172.16.25.15    username1   tty4    408/4327528     update
task_id=30      addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
Wed Jun 27 04:01:31 2001 172.16.25.15    username1   tty4    562/4327528
stoptask_id=30       addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
bytes_in=2844        bytes_out=1682  paks_in=36      paks_out=24     elapsed_time=51
Wed Jun 27 04:01:32 2001 172.16.25.15    username1   tty4    562/4327528
stoptask_id=28       service=shell   elapsed_time=57
```

Note The precise format of accounting packets records may vary depending on the security server daemon.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:30:52 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 3
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000B”
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:36:49 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 3
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000B”
        Framed-Protocol = PPP
        Framed-IP-Address = “10.1.1.1”
        Acct-Input-Octets = 8630
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
```

```
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:02:19 2001 172.16.25.15    username1   Async5  562/4327528
starttask_id=35      service=ppp
Wed Jun 27 04:02:25 2001 172.16.25.15    username1   Async5  562/4327528     update
task_id=35      service=ppp     protocol=ip     addr=10.1.1.2
Wed Jun 27 04:05:03 2001 172.16.25.15    username1   Async5  562/4327528
stoptask_id=35       service=ppp     protocol=ip     addr=10.1.1.2   bytes_in=3366
bytes_out=2149       paks_in=42      paks_out=28     elapsed_time=164
```

EXEC Accounting

EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

The following example shows the information contained in a RADIUS EXEC accounting record for a dial-in user:

```
Wed Jun 27 04:26:23 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 1
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329483”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000006”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
Wed Jun 27 04:27:25 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 1
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329483”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000006”
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a dial-in user:

```
Wed Jun 27 03:46:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=2       service=shell
Wed Jun 27 04:08:55 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=2       service=shell   elapsed_time=1354
```

The following example shows the information contained in a RADIUS EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:48:32 2001
        NAS-IP-Address = “172.16.25.15”
```

```
        NAS-Port = 26
        User-Name = “username1”
        Caller-ID = “10.68.202.158”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000010”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:48:46 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 26
        User-Name = “username1”
        Caller-ID = “10.68.202.158”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000010”
        Acct-Session-Time = 14
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:06:53 2001        172.16.25.15    username1   tty26   10.68.202.158
starttask_id=41      service=shell
Wed Jun 27 04:07:02 2001        172.16.25.15    username1   tty26   10.68.202.158
stoptask_id=41       service=shell   elapsed_time=9
```

Command Accounting

Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 27 03:46:47 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=3       service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 27 03:46:58 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=4       service=shell   priv-lvl=1      cmd=show interfaces Ethernet 0
<cr>
Wed Jun 27 03:47:03 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=5       service=shell   priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 27 03:47:17 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=6       service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 27 03:47:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=7       service=shell   priv-lvl=15     cmd=interface Serial 0 <cr>
Wed Jun 27 03:47:29 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=8       service=shell   priv-lvl=15     cmd=ip address 10.1.1.1
255.255.255.0 <cr>
```

Note The Cisco implementation of RADIUS does not support command accounting.

Connection Accounting

Connection accounting provides information about all outbound connections made from the network access server such as Telnet, LAT, TN3270, PAD, and rlogin.

The following example shows the information contained in a RADIUS connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 04:28:00 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “00000008”
        Login-Service = Telnet
        Login-IP-Host = “10.68.202.158”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:28:39 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “00000008”
        Login-Service = Telnet
        Login-IP-Host = “10.68.202.158”
        Acct-Input-Octets = 10774
        Acct-Output-Octets = 112
        Acct-Input-Packets = 91
        Acct-Output-Packets = 99
        Acct-Session-Time = 39
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 03:47:43 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=10      service=connection      protocol=telnet addr=10.68.202.158
cmd=telnet username1-sun
Wed Jun 27 03:48:38 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=10      service=connection      protocol=telnet addr=10.68.202.158
cmd=telnet username1-sun     bytes_in=4467   bytes_out=96    paks_in=61      paks_out=72
elapsed_time=55
```

The following example shows the information contained in a RADIUS connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 04:29:48 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
```

```
        Acct-Session-Id = “0000000A”
        Login-Service = Rlogin
        Login-IP-Host = “10.68.202.158”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:30:09 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “0000000A”
        Login-Service = Rlogin
        Login-IP-Host = “10.68.202.158”
        Acct-Input-Octets = 18686
        Acct-Output-Octets = 86
        Acct-Input-Packets = 90
        Acct-Output-Packets = 68
        Acct-Session-Time = 22
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 03:48:46 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=12      service=connection      protocol=rlogin addr=10.68.202.158
cmd=rlogin username1-sun /user username1
Wed Jun 27 03:51:37 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=12      service=connection      protocol=rlogin addr=10.68.202.158
cmd=rlogin username1-sun /user username1 bytes_in=659926 bytes_out=138   paks_in=2378
paks_
out=1251        elapsed_time=171
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound LAT connection:

```
Wed Jun 27 03:53:06 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX
Wed Jun 27 03:54:15 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX  bytes_in=0      bytes_out=0     paks_in=0      paks_out=0      elapsed_time=6
```

System Accounting

System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The following accounting record shows a typical TACACS+ system accounting record server indicating that AAA Accounting has been turned off:

```
Wed Jun 27 03:55:32 2001        172.16.25.15    unknown unknown unknown start
task_id=25   service=system  event=sys_acct  reason=reconfigure
```

Note The precise format of accounting packets records may vary depending on the TACACS+ daemon.

The following accounting record shows a TACACS+ system accounting record indicating that AAA Accounting has been turned on:

```
Wed Jun 27 03:55:22 2001        172.16.25.15    unknown unknown unknown stop
task_id=23   service=system  event=sys_acct  reason=reconfigure
```

Additional tasks for measuring system resources are covered in the Cisco IOS software configuration guides. For example, IP accounting tasks are described in the Configuring IP Services chapter in the *CiscoIOS Application Services Configuration Guide* .

Resource Accounting

The Cisco implementation of AAA accounting provides “start” and “stop” record support for calls that have passed user authentication. The additional feature of generating “stop” records for calls that fail to authenticate as part of user authentication is also supported. Such records are necessary for users employing accounting records to manage and monitor their networks.

This section includes the following subsections:

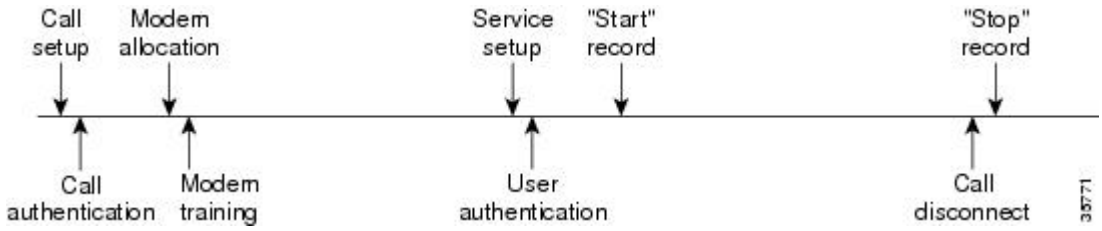
AAA Resource Failure Stop Accounting

Before AAA resource failure stop accounting, there was no method of providing accounting records for calls that failed to reach the user authentication stage of a call setup sequence. Such records are necessary for users employing accounting records to manage and monitor their networks and their wholesale customers.

This functionality generates a “stop” accounting record for any calls that do not reach user authentication; “stop” records are generated from the moment of call setup. All calls that pass user authentication behave as they did before; that is, no additional accounting records are seen.

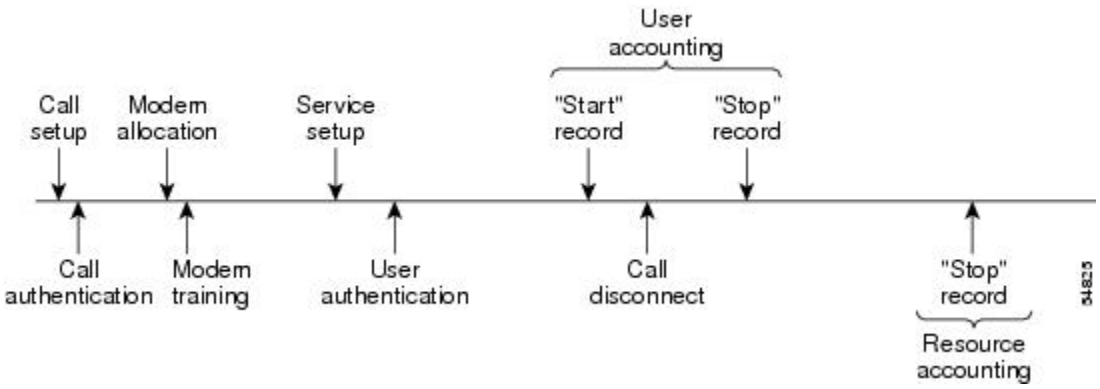
The figure below illustrates a call setup sequence with normal call flow (no disconnect) and without AAA resource failure stop accounting enabled.

Figure 6 Modem Dial-In Call Setup Sequence With Normal Flow and Without Resource Failure Stop Accounting Enabled



The figure below illustrates a call setup sequence with normal call flow (no disconnect) and with AAA resource failure stop accounting enabled.

Figure 7 Modem Dial-In Call Setup Sequence With Normal Flow and WIth Resource Failure Stop Accounting Enabled

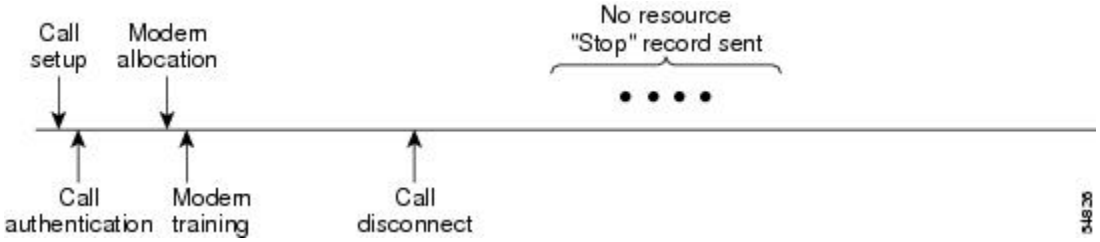


The figure below illustrates a call setup sequence with call disconnect occurring before user authentication and with AAA resource failure stop accounting enabled.

Figure 8 Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and With Resource Failure Stop Accounting Enabled

The figure below illustrates a call setup sequence with call disconnect occurring before user authentication and without AAA resource failure stop accounting enabled.

Figure 9 Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and Without Resource Failure Stop Accounting Enabled



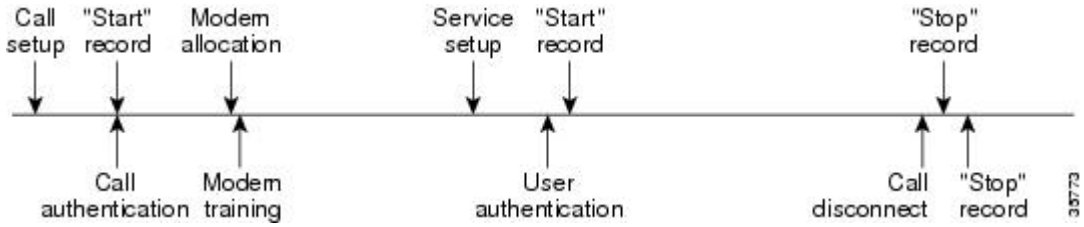
AAA Resource Accounting for Start-Stop Records

AAA resource accounting for start-stop records supports the ability to send a “start” record at each call setup, followed by a corresponding “stop” record at the call disconnect. This functionality can be used to manage and monitor wholesale customers from one source of data reporting, such as accounting records.

With this feature, a call setup and call disconnect “start-stop” accounting record tracks the progress of the resource connection to the device. A separate user authentication “start-stop” accounting record tracks the user management progress. These two sets of accounting records are interlinked by using a unique session ID for the call.

The figure below illustrates a call setup sequence with AAA resource start-stop accounting enabled.

Figure 10 Modem Dial-In Call Setup Sequence With Resource Start-Stop Accounting Enabled



VRRS Accounting

Virtual Router Redundancy Service (VRRS) provides a multiclient information abstraction and management service between a First Hop Redundancy Protocol (FHRP) and a registered client. The VRRS multiclient service provides a consistent interface with FHRP protocols by abstracting over several FHRPs and providing an idealized view of their state. VRRS manages data updates, allowing interested clients to register in one place and receive updates for named FHRP groups or all registered FHRP groups.

Virtual Router Redundancy Protocol (VRRP) is an FHRP that acts as a server that pushes FHRP status information out to all registered VRRS clients. Clients obtain status on essential information provided by the FHRP, including current and previous redundancy states, active and inactive L3 and L2 addresses, and, in some cases, information about other redundant gateways in the network. Clients can use this information to provide stateless and stateful redundancy information to clients and protocols.

- VRRS Accounting Plug-in, page 158

VRRS Accounting Plug-in

The VRRS Accounting plug-in provides a configurable AAA method list mechanism that provides updates to a RADIUS server when a VRRS group transitions its state. The VRRS accounting plug-in is an extension of existing AAA system accounting messages. The VRRS Accounting plug-in provides accounting-on and accounting-off messages and an additional Vendor-Specific Attribute (VSA) that sends the configured VRRS name in RADIUS accounting messages. The VRRS name is configured using the **vrrp name** command in interface configuration mode.

The VRRS Accounting plug-in provides a configurable AAA method list mechanism that provides updates to a RADIUS server when a VRRS group transitions its state.

The VRRS accounting plug-in is an extension of existing AAA system accounting messages. The VRRS Accounting plug-in provides accounting-on and accounting-off messages and an additional Vendor-Specific Attribute (VSA) that sends the configured VRRS name in RADIUS accounting messages. The VRRS name is configured using the **vrrp name** command in interface configuration mode. The VRRS Accounting plug-in sends an accounting-on message to RADIUS when a VRRS group transitions to the master state, and it sends an accounting-off message when a VRRS group transitions from the master state.

The following RADIUS attributes are included in VRRS accounting messages by default:

- Attribute 4, NAS-IP-Address
- Attribute 26, Cisco VSA Type 1, VRRS Name
- Attribute 40, Acct-Status-Type
- Attribute 41, Acct-Delay-Time
- Attribute 44, Acct-Session-Id

Accounting messages for a VRRS transitioning out of master state are sent after all PPPoE accounting stop messages for sessions that are part of that VRRS.

AAA Accounting Enhancements

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Broadcasting is allowed among groups of RADIUS or TACACS+ servers, and each server group can define its backup servers for failover independently of other groups.

Thus, service providers and their end customers can use different protocols (RADIUS or TACACS+) for the accounting server. Service providers and their end customers can also specify their backup servers independently. As for voice applications, redundant accounting information can be managed independently through a separate group with its own failover sequence.

AAA Session MIB

The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). The data of the client is presented so that it correlates directly to the AAA Accounting information reported by either the RADIUS or the TACACS+ server. AAA session MIB provides the following information:

- Statistics for each AAA function (when used in conjunction with the **show radius statistics** command)
- Status of servers providing AAA functions
- Identities of external AAA servers
- Real-time information (such as idle times), providing additional criteria for use by SNMP networks for assessing whether or not to terminate an active call

Note This command is supported only on Cisco AS5300 and Cisco AS5800 universal access server platforms.

The table below shows the SNMP user-end data objects that can be used to monitor and terminate authenticated client connections with the AAA session MIB feature.

Table 19 SNMP End-User Data Objects

SessionId	The session identification used by the AAA Accounting protocol (same value as reported by RADIUS attribute 44 (Acct-Session-ID)).
UserId	The user login ID or zero-length string if a login is unavailable.
IpAddr	The IP address of the session or 0.0.0.0 if an IP address is not applicable or unavailable.
IdleTime	The elapsed time in seconds that the session has been idle.
Disconnect	The session termination object used to disconnect the given client.
CallId	The entry index corresponding to this accounting session that the Call Tracker record stored.

The table below describes the AAA summary information provided by the AAA session MIB feature using SNMP on a per-system basis.

Table 20 SNMP AAA Session Summary

ActiveTableEntries	Number of sessions currently active.
ActiveTableHighWaterMark	Maximum number of sessions present at once since last system reinstallation.
TotalSessions	Total number of sessions since last system reinstallation.
DisconnectedSessions	Total number of sessions that have been disconnected using since last system reinstallation.

Accounting Attribute-Value Pairs

The network access server monitors the accounting functions defined in either TACACS+ AV pairs or RADIUS attributes, depending on which security method is implemented.

How to Configure AAA Accounting

Configuring AAA Accounting Using Named Method Lists

To configure AAA Accounting using named method lists, perform the following steps:

Note System accounting does not use named method lists. For system accounting, define only the default method list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2...*]]
4. Do one of the following:
   - Router(config)# **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
   -
   -
   -
   - Router(config)# **interface** *interface-type interface-number*
5. Do one of the following:
   - Router(config-line)# **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*}
   -
   -
   -
   - Router(config-if)# **ppp accounting**{**default** | *list-name*}
6. Router(config-line)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2*...]] **Example:** `Router(config)# aaa accounting system default start-stop`	Creates an accounting method list and enables accounting. The argument *list-name* is a character string used to name the created list.
Step 4	Do one of the following: • Router(config)# **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] • • • • Router(config)# **interface** *interface-type interface-number* **Example:** `Router(config)# line aux line1`	Enters the line configuration mode for the lines to which the accounting method list is applied. or Enters the interface configuration mode for the interfaces to which the accounting method list is applied.

	Command or Action	Purpose
Step 5	Do one of the following: • Router(config-line)# **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*} • • • • Router(config-if)# **ppp accounting**{**default** | *list-name*} **Example:** `Router(config-line)# accounting arap default`	Applies the accounting method list to a line or set of lines. or Applies the accounting method list to an interface or set of interfaces.
Step 6	Router(config-line)# **end** **Example:** `Router(config-line)# end`	(Optional) Exits line configuration mode and returns to global configuration mode.

This section includes the following subsection:

- Configuring RADIUS System Accounting, page 163

Configuring RADIUS System Accounting

Perform this task to configure RADIUS system accounting on the global RADIUS server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server accounting system host-config**
5. **aaa group server radius** *server-name*
6. **server-private** {*host-name* | *ip-address*} **key** {[**0** *server-key* | **7** *server-key*] *server-key*
7. **accounting system host-config**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA network security services.
Step 4	**radius-server accounting system host-config** **Example:** `Router(config)# radius-server accounting system host-config`	Enables the router to send a system accounting record for the addition and deletion of a RADIUS server.
Step 5	**aaa group server radius** *server-name* **Example:** `Router(config)# aaa group server radius radgroup1`	Adds the RADIUS server and enters server-group configuration mode. • The *server-name* argument specifies the RADIUS server group name.
Step 6	**server-private** {*host-name* | *ip-address*} **key** {[**0** *server-key* | **7** *server-key*] *server-key* **Example:** `Router(config-sg-radius)# server-private 172.16.1.11 key cisco`	Enters the hostname or IP address of the RADIUS server and hidden server key. • (Optional) **0** with the *server-key*argument specifies that an unencrypted (cleartext) hidden server key follows. • (Optional) **7** with the *server-key* argument specifies that an encrypted hidden server key follows. • The *server-key* argument specifies the hidden server key. If the *server-key*argument is configured without the **0** or **7** preceding it, it is unencrypted. **Note** Once the **server-private** command is configured, RADIUS system accounting is enabled.

	Command or Action	Purpose
Step 7	**accounting system host-config** **Example:** `Router(config-sg-radius)# accounting system host-config`	Enables the generation of system accounting records for private server hosts when they are added or deleted.
Step 8	**end** **Example:** `Router(config-sg-radius)# end`	Exits server-group (config-sg-radius) configuration mode and returns to global configuration mode.

Suppressing Generation of Accounting Records for Null Username Sessions

When AAA Accounting is activated, the Cisco IOS software issues accounting records for all users on the system, including users whose username string, because of protocol translation, is NULL. An example of this is users who come in on lines where the **aaa authentication login** *method-list* **none** command is applied. To prevent accounting records from being generated for sessions that do not have usernames associated with them, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting suppress null-username**	Prevents accounting records from being generated for users whose username string is NULL.

Generating Interim Accounting Records

To enable periodic interim accounting records to be sent to the accounting server, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting update** [**newinfo**] [**periodic**] *number*	Enables periodic interim accounting records to be sent to the accounting server.

When the **aaa accounting update**commandis activated, the Cisco IOS software issues interim accounting records for all users on the system. If the keyword **newinfo** is used, interim accounting records are sent to the accounting server every time there is new accounting information to report. An example of this would be when IPCP completes IP address negotiation with the remote peer. The interim accounting record includes the negotiated IP address used by the remote peer.

When used with the keyword **periodic**, interim accounting records are sent periodically as defined by the *number* argument. The interim accounting record contains all of the accounting information recorded for that user up to the time the interim accounting record is sent.

Caution Using the **aaa accounting update periodic** command can cause heavy congestion when many users are logged in to the network.

Generating Accounting Records for Failed Login or Session

When AAA Accounting is activated, the Cisco IOS software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail to authenticate at login or during session negotiation, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting send stop-record authentication failure**	Generates "stop" records for users who fail to authenticate at login or during session negotiation using PPP.
Router(config)# aaa accounting send stop-record always	Sends authentication, authorization, and accounting (AAA) stop records regardless of whether a start record was sent earlier.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, you can specify the NETWORK records to be generated before EXEC-stop records. In cases such as billing customers for specific services, it can be desirable to keep network start and stop records together, essentially "nesting" them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

To nest accounting records for user sessions, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting nested**	Nests network accounting records.

Configuring AAA Resource Failure Stop Accounting

To enable resource failure stop accounting, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting resource** *method-list* **stop-failure group** *server-group*	Generates a “stop” record for any calls that do not reach user authentication. **Note** Before configuring this feature, the tasks described in the Prerequisites for Configuring Accounting, page 145 section must be performed, and SNMP must be enabled on the network access server. For more information about enabling SNMP on a Cisco router or access server, see the Configuring SNMP Support chapter in the Cisco IOS Network Management Configuration Guide.

Configuring AAA Resource Accounting for Start-Stop Records

To enable full resource accounting for start-stop records, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting resource** *method-list* **start-stop group** *server-group*	Supports the ability to send a “start” record at each call setup. followed with a corresponding “stop” record at the call disconnect. **Note** Before configuring this feature, the tasks described in the Prerequisites for Configuring Accounting, page 145 section must be performed, and SNMP must be enabled on the network access server. For more information about enabling SNMP on a Cisco router or access server, see the Configuring SNMP Support chapter in the Cisco IOS Network Management Configuration Guide. **Note**

Configuring AAA Broadcast Accounting

To configure AAA broadcast accounting, use the aaa accounting command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] *method1* [*method2...*]	Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring Per-DNIS AAA Broadcast Accounting

To configure AAA broadcast accounting per DNIS, use the **aaa dnis map accounting network**command in global configuration mode:

Command	Purpose
`Router(config)#` **aaa dnis map** *dnis-number* **accounting network** [**start-stop** | **stop-only** | **none**] [**broadcast**] *method1* [*method2...*]	Allows per-DNIS accounting configuration. This command has precedence over the global **aaa accounting** command. Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring AAA Session MIB

The following tasks must be performed before configuring the AAA session MIB feature:

- Configure SNMP. For information on SNMP, see the chapter Configuring SNMP Support in the Cisco IOS Network Management Configuration Guide.
- Configure AAA.
- Define the RADIUS or TACACS+ server characteristics.

Note Overusing SNMP can affect the overall system performance; therefore, normal network management performance must be considered when this feature is used.

To configure AAA session MIB, use the following command in global configuration mode

SUMMARY STEPS

1. Router(config)# **aaa session-mib disconnect**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa session-mib disconnect**	Monitors and terminates authenticated client connections using SNMP. To terminate the call, the **disconnect** keyword must be used.

Configuring VRRS Accounting

Perform the following task to configure Virtual Router Redundancy Service (VRRS) to send AAA Accounting messages to the AAA server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting vrrs** {**default** | *list-name*} **start-stop** *method1* [*method2...*]
4. **aaa attribute list** *list-name*
5. **attribute type** *name value* [**service** *service*] [**protocol** *protocol*][**mandatory**][**tag** *tag-value*]
6. **exit**
7. **vrrs** *vrrs-group-name*
8. **accounting delay** *seconds*
9. **accounting method** {**default** | *accounting-method-list*}
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting vrrs** {**default** | *list-name*} **start-stop** *method1* [*method2...*] **Example:** `Router(config)# aaa accounting vrrs default start-stop`	Enables AAA accounting for VRRS.
Step 4	**aaa attribute list** *list-name* **Example:** `Router(config)# aaa attribute list list1`	Defines a AAA attribute list locally on a router, and enters attribute list configuration mode.

	Command or Action	Purpose
Step 5	**attribute type** *name value* [**service** *service*] [**protocol** *protocol*] [**mandatory**][**tag** *tag-value*] **Example:** `Router(config-attr-list)# attribute type example 1`	Defines an attribute type that is to be added to an attribute list locally on a router.
Step 6	**exit** **Example:** `Router(config-attr-list)# exit`	Exits attribute list configuration mode and returns to global configuration mode.
Step 7	**vrrs** *vrrs-group-name* **Example:** `Router(config)# vrrs vrrs1`	(Optional) Defines a VRRP group and configures parameters for the VRRS group, and enters VRRS configuration mode.
Step 8	**accounting delay** *seconds* **Example:** `Router(config-vrrs)# accounting delay 10`	(Optional) Specifies the delay time for sending accounting-off messages to the VRRS.
Step 9	**accounting method** {**default** | *accounting-method-list*} **Example:** `Router(config-vrrs)# accounting method default`	(Optional) Enables VRRS accounting for a VRRP group.
Step 10	**exit** **Example:** `Router(config-vrrs)# exit`	Exits VRRS configuration mode.

Establishing a Session with a Router if the AAA Server is Unreachable

To establish a console or telnet session with a router if the AAA server is unreachable, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **no aaa accounting system guarantee-first**	The **aaa accounting system guarantee-first** command guarantees system accounting as the first record, which is the default condition. In some situations, users may be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than three minutes. To resolve this problem, the **no aaa accounting system guarantee-first** command can be used.

Note Entering the **no aaa accounting system guarantee-first** command is not the only condition by which the console or telnet session can be started. For example, if the privileged EXEC session is being authenticated by TACACS and the TACACS server is not reachable, then the session cannot start.

Monitoring Accounting

No specific **show** command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users currently logged in, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **show accounting**	Allows display of the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Troubleshooting Accounting

To troubleshoot accounting information, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **debug aaa accounting**	Displays information on accountable events as they occur.

Configuration Examples for AAA Accounting

Example Configuring Named Method List

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) in order for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database is queried for authentication and authorization information, and accounting services are handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network blue1 group radius local
aaa accounting network red1 start-stop group radius group tacacs+
username root password ALongPassword
tacacs-server host 172.31.255.0
tacacs-server key goaway
radius-server host 172.16.2.7
radius-server key myRaDiUSpassWoRd
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization blue1
 ppp accounting red1
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins”, which specifies that first RADIUS authentication and then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.
- The **aaa authorization network blue1 group radius local** command defines the network authorization method list named “blue1”, which specifies that RADIUS authorization is used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization is performed.
- The **aaa accounting network red1 start-stop group radius group tacacs+**command defines the network accounting method list named red1, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) are used on serial lines using PPP. If the RADIUS server fails to respond, accounting services are handled by a TACACS+ server.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **tacacs-server host** command defines the name of the TACACS+ server host.
- The **tacacs-server key** command defines the shared secret text string between the network access server and the TACACS+ server host.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.

- The **ppp authentication chap dialins**command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **ppp authorization blue1**command applies the blue1 network authorization method list to the specified interfaces.
- The **ppp accounting red1**command applies the red1 network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

The **show accounting**command yields the following output for the preceding configuration:

```
Active Accounted actions on tty1, User username2 Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

The table below describes the fields contained in the preceding output.

Table 21 show accounting Field Descriptions

Field	Description
Active Accounted actions on	Terminal line or interface name user with which the user logged in.
User	User’s ID.
Priv	User’s privilege level.
Task ID	Unique identifier for each accounting session.
Accounting record	Type of accounting session.
Elapsed	Length of time (hh:mm:ss) for this session type.
attribute=value	AV pairs associated with this accounting session.

Example Configuring AAA Resource Accounting

The following example shows how to configure the resource failure stop accounting and resource accounting for start-stop records functions:

```
!Enable AAA on your network access server.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login
authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
```

```
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default
method to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all start-
stop accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method
to use for all start-stop accounting services.
aaa accounting network default start-stop group radius
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

Example Configuring AAA Broadcast Accounting

The following example shows how to turn on broadcast accounting using the global **aaa accounting** command:

```
aaa group server radius isp
 server 10.0.0.1
 server 10.0.0.2
aaa group server tacacs+ isp_customer
 server 172.0.0.1
aaa accounting network default start-stop broadcast group isp group isp_customer
radius-server host 10.0.0.1
radius-server host 10.0.0.2
radius-server key key1
tacacs-server host 172.0.0.1 key key2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connections to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Example Configuring Per-DNIS AAA Broadcast Accounting

The following example shows how to turn on per DNIS broadcast accounting using the global **aaa dnis map accounting network**command:

```
aaa group server radius isp
 server 10.0.0.1
 server 10.0.0.2
aaa group server tacacs+ isp_customer
 server 172.0.0.1
aaa dnis map enable
aaa dnis map 7777 accounting network start-stop broadcast group isp group isp_customer
radius-server host 10.0.0.1
radius-server host 10.0.0.2
radius-server key key_1
tacacs-server host 172.0.0.1 key key_2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connection calls having DNIS number 7777 to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Example AAA Session MIB

The following example shows how to set up the AAA session MIB feature to disconnect authenticated client connections for PPP users:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
aaa session-mib disconnect
```

Example Configuring VRRS Accounting

The following example shows how to configure VRRS to send AAA Accounting messages to the AAA server:

```
Router# configure terminal
Router(config)# aaa accounting vrrs vrrp-mlist-1 start-stop group radius
Router(config)# aaa attribute list vrrp-1-attr
Router(config-attr-list)# attribute type account-delay 10
Router(config-attr-list)# exit
Router(config)# vrrs vrrp-group-1
Router(config-vrrs)# accounting delay 10
Router(config-vrrs)# accounting method vrrp-mlist-1
Router(config-vrrs)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Authorization	Configuring Authorization module
Authentication	Configuring Authentication module
Accounting Commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2903	*Generic AAA Architecture*
RFC 2904	*AAA Authorization Framework*
RFC 2906	*AAA Authorization Requirements*
RFC 2989	*Criteria for Evaluating AAA Protocols for Network Access*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 22 Feature Information for Configuring Accounting

Feature Name	Releases	Feature Information
AAA Broadcast Accounting	12.2 12.2S 12.2SB 12.2SX 12.4T	AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously.
AAA Resource Accounting for Start-Stop Records	12.2 12.4T 12.2S 12.2SB 12.2SX	AAA resource accounting for start-stop records supports the ability to send a “start” record at each call setup, followed by a corresponding “stop” record at the call disconnect. This functionality can be used to manage and monitor wholesale customers from one source of data reporting, such as accounting records.
AAA Session MIB	12.2 12.4T 12.2S 12.2SB 12.2SX	The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using SNMP. The data of the client is presented so that it correlates directly to the AAA Accounting information reported by either the RADIUS or the TACACS+ server.
AAA: IPv6 Accounting Delay Enhancements	15.1(1)S	VRRS provides a multiclient information abstraction and management service between a First Hop Redundancy Protocol (FHRP) and a registered client.

AAA Dead-Server Detection

The AAA Dead-Server Detection feature allows you to configure the criteria to be used to mark a RADIUS server as dead. If no criteria are explicitly configured, the criteria are computed dynamically on the basis of the number of outstanding transactions. Using this feature will result in less deadtime and quicker packet processing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for AAA Dead-Server Detection

- You must have access to a RADIUS server.
- You should be familiar with configuring a RADIUS server.
- You should be familiar with configuring authentication, authorization, and accounting (AAA).
- Before a server can be marked as dead, you must first configure the **radius-server deadtime** command. If this command is not configured, even if the criteria are met for the server to be marked as dead, the server state will be the “up” state.

Restrictions for AAA Dead-Server Detection

- Original transmissions are not counted in the number of consecutive timeouts that must occur on the router before the server is marked as dead--only the number of retransmissions are counted.

Information About AAA Dead-Server Detection

Criteria for Marking a RADIUS Server As Dead

The AAA Dead-Server Detection feature allows you to determine the criteria that are used to mark a RADIUS server as dead. That is, you can configure the minimum amount of time, in seconds, that must elapse from the time that the router last received a valid packet from the RADIUS server to the time the server is marked as dead. If a packet has not been received since the router booted, and there is a timeout, the time criterion will be treated as though it has been met.

In addition, you can configure the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. If the server performs both authentication and accounting, both types of packets are included in the number. Improperly constructed packets are counted as though they are timeouts. Only retransmissions are counted, not the initial transmission. (Each timeout causes one retransmission to be sent.)

Note Both the time criterion and the tries criterion must be met for the server to be marked as dead.

The RADIUS dead-server detection configuration will result in the prompt detection of RADIUS servers that have stopped responding. This configuration will also result in the avoidance of servers being improperly marked as dead when they are “swamped” (responding slowly) and the avoidance of the state of servers being rapidly changed from dead to live to dead again. This prompt detection of nonresponding RADIUS servers and the avoidance of swamped and dead-to-live-to-dead-again servers will result in less deadtime and quicker packet processing.

How to Configure AAA Dead-Server Detection

Configuring AAA Dead-Server Detection

To configure AAA Dead-Server Detection, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server deadtime** *minutes*
5. **radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.
Step 4	**radius-server deadtime** *minutes* **Example:** `Router (config)# radius-server deadtime 5`	Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.
Step 5	**radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*] **Example:** `Router (config)# radius-server dead-criteria time 5 tries 4`	Forces one or both of the criteria--used to mark a RADIUS server as dead--to be the indicated constant.

• Troubleshooting Tips, page 181

Troubleshooting Tips

After you have configured AAA Dead-Server Detection, you should verify your configuration using the **show running-config** command. This verification is especially important if you have used the **no** form of

the **radius-server dead-criteria** command. The output of the **show running-config** command must show the same values in the "Dead Criteria Details" field that you configured using the **radius-server dead-criteria** command.

Verifying AAA Dead-Server Detection

To verify your AAA Dead-Server Detection configuration, perform the following steps. The **show** and **debug** commands may be used in any order.

SUMMARY STEPS

1. **enable**
2. **debug aaa dead-criteria transactions**
3. **show aaa dead-criteria**
4. **show aaa servers** [**private** | **public**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa dead-criteria transactions** **Example:** `Router# debug aaa dead-criteria transactions`	Displays AAA dead-criteria transaction values.
Step 3	**show aaa dead-criteria** **Example:** `Router# show aaa dead-criteria`	Displays dead-criteria information for a AAA server.
Step 4	**show aaa servers** [**private**	**public**] **Example:** `Router# show aaa server private`	Displays the status and number of packets that are sent to and received from all public and private authentication, authorization, and accounting (AAA) RADIUS servers. • The **private** keyword optionally displays the AAA servers only. • The **public** keyword optionally displays the AAA servers only.

Configuration Examples for AAA Dead-Server Detection

Configuring AAA Dead-Server Detection Example

The following example shows that the router will be considered dead after 5 seconds and four tries:

```
Router (config)# aaa new-model
Router (config)# radius-server deadtime 5
Router (config)# radius-server dead-criteria time 5 tries 4
```

debug aaa dead-criteria transactions Command Example

The following output example shows dead-criteria transaction information for a particular server group:

```
Router# debug aaa dead-criteria transactions
AAA Transaction debugs debugging is on
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Computed Retransmit Tries: 22, Current Max Tries: 22
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Computed Dead Detect Interval: 25s, Current Max
Interval: 25s
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Estimated Outstanding Transactions: 6, Current Max
Transactions: 6
```

show aaa dead-criteria Command Example

The following output example shows that dead-server-detection information has been requested for a RADIUS server at the IP address 172.19.192.80:

```
Router# show aaa dead-criteria radius 172.19.192.80 radius
RADIUS Server Dead Criteria:
============================
Server Details:
    Address : 172.19.192.80
    Auth Port : 1645
    Acct Port : 1646
Server Group : radius
Dead Criteria Details:
    Configured Retransmits : 62
    Configured Timeout : 27
    Estimated Outstanding Transactions: 5
    Dead Detect Time : 25s
    Computed Retransmit Tries: 22
    Statistics Gathered Since Last Successful Transaction
================================================
Max Computed Outstanding Transactions: 5
Max Computed Dead Detect Time: 25s
Max Computed Retransmits : 22
```

Additional References

The following sections provide references related to the AAA Dead-Server Detection feature.

Related Documents

Related Topic	Document Title
Configuring RADIUS	Configuring RADIUS feature module.
Configuring AAA	Configuring Authentication
	Configuring Authorization
	Configuring Accounting
Security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2865	*Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for AAA Dead-Server Detection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 23 Feature Information for AAA Dead-Server Detection

Feature Name	Releases	Feature Information
AAA Dead-Server Detection	12.3(6) 12.3(7)T Cisco IOS XE Release 2.1 Cisco IOS XE 3.1.0SG	Allows you to configure the criteria to be used to mark a RADIUS server as dead. The following commands were introduced or modified: **debug aaa dead-criteria transactions**, **radius-server dead-criteria**, **show aaa dead-criteria**, **show aaa servers**.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Per VRF AAA

The Per VRF AAA feature allows ISPs to partition authentication, authorization, and accounting (AAA) services on the basis of Virtual Private Network (VPN) routing and forwarding (VRF) instances, allowing their customers to control some of their own AAA services.

The list of servers in server groups is extended to include the definitions of private servers in addition to references to the hosts in the global configuration, allowing access to both customer servers and global service provider servers simultaneously.

For Cisco IOS Release 12.2(15)T or later releases, a customer template can be used, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template. This feature has also been referred to as the Dynamic Per VRF AAA feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Per VRF AAA

Before configuring the Per VRF AAA feature, AAA must be enabled. See “How to Configure Per VRF AAA” section on page 6 for more information.

Restrictions for Per VRF AAA

- This feature is supported only for RADIUS servers.
- Operational parameters should be defined once per VRF rather than set per server group, because all functionality must be consistent between the network access server (NAS) and the AAA servers.
- The ability to configure a customer template either locally or remotely is available only for Cisco IOS Release 12.2(15)T and later releases.

Information About Per VRF AAA

When you use the Per VRF AAA feature, AAA services can be based on 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. Thus, ISPs can scale their VPN offerings more efficiently because they no longer have to use RADIUS proxies and ISPs can also provide their customers with additional flexibility.

- How Per VRF AAA Works, page 188
- AAA Accounting Records, page 189
- New Vendor-Specific Attributes, page 189

How Per VRF AAA Works

To support AAA on a per customer basis, some AAA features must be made VRF aware. That is, ISPs must be able to define operational parameters--such as AAA server groups, method lists, system accounting, and protocol-specific parameters--and bind those parameters to a particular VRF instance. Defining and binding the operational parameters can be accomplished using one or more of the following methods:

- Virtual private dialup network (VPDN) virtual template or dialer interfaces that are configured for a specific customer
- Locally defined customer templates--Per VPN with customer definitions. The customer template is stored locally on the VHG. This method can be used to associate a remote user with a specific VPN based on the domain name or dialed number identification service (DNIS) and provide the VPN-specific configuration for virtual access interface and all operational parameters for the customer AAA server.
- Remotely defined customer templates--Per VPN with customer definitions that are stored on the service provider AAA server in a RADIUS profile. This method is used to associate a remote user with a specific VPN based on the domain name or DNIS and provide the VPN-specific configuration for the virtual access interface and all operational parameters for the AAA server of the customer.

Note The ability to configure locally or remotely defined customer templates is available only with Cisco IOS Release 12.2(15)T and later releases.

AAA Accounting Records

The Cisco implementation of AAA accounting provides “start” and “stop” record support for calls that have passed user authentication. Start and stop records are necessary for users employing accounting records to manage and monitor their networks.

New Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (VSA) attribute 26. Attribute 26 encapsulates VSAs, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco’s vendor-ID is 9, and the supported option has vendor-type 1, which is named “cisco-avpair.” The value is a string of the following format:

```
protocol : attribute sep value *
```

“Protocol” is a value of the Cisco “protocol” attribute for a particular type of authorization. “Attribute” and “value” are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and “sep” is “=” for mandatory attributes and “*” for optional attributes. This format allows the full set of features available for TACACS+ authorization to be used also for RADIUS.

The table below summarizes the VSAs that are now supported with Per VRF AAA.

Table 24 VSAs supported with Per VRF AAA

VSA Name	Value Type	Description
Note Each VSA must have the prefix “template:” before the VSA name, unless a different prefix is explicitly stated.		
account-delay	string	This VSA must be “on.” The functionality of this VSA is equal to the **aaa accounting delay-start**command for the customer template.
account-send-stop	string	This VSA must be “on.” The functionality of this VSA is equal to the **aaa accounting send stop-record authentication**command with the **failure** keyword.
account-send-success-remote	string	This VSA must be “on.” The functionality of this VSA is equal to the **aaa accounting send stop-record authentication**command with the **success** keyword.

VSA Name	Value Type	Description
attr-44	string	This VSA must be “access-req.” The functionality of this VSA is equal to the **radius-server attribute 44 include-in-access-req** command.
ip-addr	string	This VSA specifies the IP address, followed by the mask that the router uses to indicate its own IP address and mask in negotiation with the client; for example, ip-addr=192.168.202.169 255.255.255.255
ip-unnumbered	string	This VSA specifies the name of an interface on the router. The functionality of this VSA is equal to the **ip unnumbered** command, which specifies an interface name such as “Loopback 0.”
ip-vrf	string	This VSA specifies which VRF will be used for the packets of the end user. This VRF name should match the name that is used on the router via the **ip vrf forwarding** command.
peer-ip-pool	string	This VSA specifies the name of an IP address pool from which an address will be allocated for the peer. This pool should be configured using the **ip local pool**command or should be automatically downloadable via RADIUS.

VSA Name	Value Type	Description
ppp-acct-list	string	This VSA defines the accounting method list that is to be used for PPP sessions. The VSA syntax is as follows: “ppp-acct-list=[start-stop | stop-only | none] group X [group Y] [broadcast].” It is equal to the **aaa accounting network mylist** command functionality. The user must specify at least one of the following options: start-stop, stop-only, or none. If either start-stop or stop-only is specified, the user must specify at least one, but not more than four, group arguments. Each group name must consist of integers. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.” After each group has been specified, the user can specify the broadcast option.
ppp-authen-list	string	This VSA defines which authentication method list is to be used for PPP sessions and, if more than one method is specified, in what order the methods should be used. The VSA syntax is as follows: “ppp-authen-list=[groupX | local | local-case | none | if-needed],” which is equal to the **aaa authentication ppp mylist** command functionality. The user must specify at least one, but no more than four, authentication methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.”

VSA Name	Value Type	Description
ppp-authen-type	string	This VSA allows the end user to specify at least one of the following authentication types: pap, chap, eap, ms-chap, ms-chap-v2, any, or a combination of the available types that is separated by spaces. The end user will be permitted to log in using only the methods that are specified in this VSA. PPP will attempt these authentication methods in the order presented in the attribute.
ppp-author-list	string	This VSA defines the authorization method list that is to be used for PPP sessions. It indicates which methods will be used and in what order. The VSA syntax is as follows: “ppp-author-list=[groupX] [local] [if-authenticated] [none],” which is equal to the **aaa authorization network mylist**command functionality. The user must specify at least one, but no more than four, authorization methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.”

Note The RADIUS VSAs--rad-serv, rad-server-filter, rad-serv-source-if, and rad-serv-vrf--must have the prefix “aaa:” before the VSA name.

VSA Name	Value Type	Description
rad-serv	string	This VSA indicates the IP address, key, timeout, and retransmit number of a server, as well as the group of the server. The VSA syntax is as follows: “rad-serv=a.b.c.d [key SomeKey] [auth-port X] [acct-port Y] [retransmit V] [timeout W].” Other than the IP address, all parameters are optional and can be issued in any order. If the optional parameters are not specified, their default values will be used. The key cannot contain any spaces; for “retransmit V,” “V” can range from 1-100; for “timeout W,” the “W” can range from 1-1000.
rad-serv-filter	string	The VSA syntax is as follows: “rad-serv-filter=authorization | accounting-request | reply-accept | reject-filtername.” The filtername must be defined via the **radius-server attribute list filtername** command.
rad-serv-source-if	string	This VSA specifies the name of the interface that is used for transmitting RADIUS packets. The specified interface must match the interface configured on the router.
rad-serv-vrf	string	This VSA specifies the name of the VRF that is used for transmitting RADIUS packets. The VRF name should match the name that was specified via the **ip vrf forwarding** command.

How to Configure Per VRF AAA

Configuring Per VRF AAA

Configuring AAA

Perform this task to enable AAA:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **ip vrf default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**ip vrf default** **Example:** `Router(config)# ip vrf default`	This command must be configured before any VRF-related AAA commands are configured, such as the **radius-server domain-stripping** command, to ensure that the default VRF name is a NULL value until a default VRF name is configured.

Configuring Server Groups

Perform this task to configure server groups.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *groupname*
5. **server-private** *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa group server radius** *groupname* **Example:** `Router(config)# aaa group server radius v2.44.com`	Groups different RADIUS server hosts into distinct lists and distinct methods. Enters server-group configuration mode.
Step 5	**server-private** *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] **Example:** `Router(config-sg-radius)# server-private 10.10.130.2 auth-port 1600 key ww`	Configures the IP address of the private RADIUS server for the group server. **Note** If private server parameters are not specified, global configurations will be used. If global configurations are not specified, default values will be used.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits from server-group configuration mode; returns to global configuration mode.

Configuring Authentication Authorization and Accounting for Per VRF AAA

Perform this task to configure authentication, authorization, and accounting for Per VRF AAA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2*...]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**}{**default** | *list-name*} *method1* [*method2*...]
6. **aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname*
7. **aaa accounting delay-start** [**vrf** *vrf-name*]
8. **aaa accounting send stop-record authentication** {**failure** | **success remote-server**} [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.

Command or Action	Purpose
Step 4 **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication ppp method_list_v2.44.com group v2.44.com`	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP.
Step 5 **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**}{**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authorization network method_list_v2.44.com group v2.44.com`	Sets parameters that restrict user access to a network.
Step 6 **aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config)# aaa accounting system default vrf v2.44.com start-stop group v2.44.com`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. **Note** The **stop-only** keyword is not available in Cisco IOS Release 12.4(24)T and later releases.
Step 7 **aaa accounting delay-start** [**vrf** *vrf-name*] **Example:** `Router(config)# aaa acounting delay-start vrf v2.44.com`	Displays generation of the start accounting records until the user IP address is established.

Command or Action	Purpose
Step 8 **aaa accounting send stop-record authentication** {**failure** | **success remote-server**} [**vrf** *vrf-name*] **Example:** `Router(config)# aaa accounting send stop-record authentication failure vrf v2.44.com`	Generates accounting stop records. When using the **failure** keyword a "stop" record will be sent for calls that are rejected during authentication. When using the **success** keyword a "stop" record will be sent for calls that meet one of the following criteria: • Calls that are authenticated by a remote AAA server when the call is terminated. • Calls that are not authenticated by a remote AAA server and the start record has been sent. • Calls that are successfully established and then terminated with the "stop-only" **aaa accounting** configuration. **Note** The **success** and **remote-server** keywords are available in Cisco IOS Release 12.4(2)T and later releases. **Note** The **success** and **remote-server** keywords are not available in Cisco IOS Release 12.2SX.

Configuring RADIUS-Specific Commands for Per VRF AAA

To configure RADIUS-specific commands for Per VRF AAA you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*]
4. **radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*]

DETAILED STEPS

Command or Action	Purpose
Step 1 **enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*] **Example:** `Router(config)# ip radius source-interface loopback55`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets and enables the specification on a per-VRF basis.
Step 4	**radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*] **Example:** `Router(config)# radius-server attribute 44 include-in-access-req vrf v2.44.com`	Sends RADIUS attribute 44 in access request packets before user authentication and enables the specification on a per-VRF basis.

Configuring Interface-Specific Commands for Per VRF AAA

Perform this task to configure interface-specific commands for Per VRF AAA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip vrf forwarding** *vrf-name*
5. **ppp authentication** {*protocol1* [*protocol2*...]} *listname*
6. **ppp authorization** *list-name*
7. **ppp accounting default**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface loopback11`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding v2.44.com`	Associates a VRF with an interface.
Step 5	**ppp authentication** {*protocol1* [*protocol2*...]} *listname* **Example:** `Router(config-if)# ppp authentication chap callin V2_44_com`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 6	**ppp authorization** *list-name* **Example:** `Router(config-if)# ppp authorization V2_44_com`	Enables AAA authorization on the selected interface.
Step 7	**ppp accounting default** **Example:** `Router(config-if)# ppp accounting default`	Enables AAA accounting services on the selected interface.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits interface configuration mode.

Configuring Per VRF AAA Using Local Customer Templates

- Configuring AAA with Local Customer Templates, page 201

Configuring AAA with Local Customer Templates

Perform the tasks as outlined in the Configuring AAA section.

Configuring Server Groups with Local Customer Templates

Perform the tasks as outlined in the Configuring Server Groups.

Configuring Authentication Authorization and Accounting for Per VRF AAA with Local Customer Templates

Perform the tasks as outlined in the Configuring Authentication Authorization and Accounting for Per VRF AAA, page 196.

Configuring Authorization for Per VRF AAA with Local Customer Templates

Perform this task to configure authorization for Per VRF AAA with local templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization network default local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables the use of local or remote templates.
Step 4	**aaa authorization network default local** **Example:** `Router(config)# aaa authorization network default local`	Specifies local as the default method for authorization.

Configuring Local Customer Templates

Perform this task to configure local customer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn search-order domain**
4. **template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**]
5. **peer default ip address pool** *pool-name*
6. **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
7. **ppp authorization** [**default** | *list-name*]
8. **aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Command or Action	Purpose
Step 3 **vpdn search-order domain** **Example:** `Router (config)# vpdn search-order domain`	Looks up the profiles based on domain.
Step 4 **template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**] **Example:** `Router (config)# template v2.44.com`	Creates a customer profile template and assigns a unique name that relates to the customer that will be receiving it. Enters template configuration mode. **Note** Steps 5, 6, and 7 are optional. Enter **multilink**, **peer**, and **ppp** keywords appropriate to customer application requirements.
Step 5 **peer default ip address pool** *pool-name* **Example:** `Router(config-template)# peer default ip address pool v2_44_com_pool`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.
Step 6 **ppp authentication** {*protocol1* [*protocol2*...]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 7 **ppp authorization** [**default** | *list-name*] **Example:** `Router(config-template)# ppp authorization v2_44_com`	(Optional) Sets the PPP link authorization method.
Step 8 **aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config-template)# aaa accounting v2_44_com`	(Optional) Enables AAA operational parameters for the specified customer profile.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.

Configuring Per VRF AAA Using Remote Customer Templates

Configuring AAA with Remote Customer Templates

Perform the tasks as outlined in the Configuring AAA section.

Configuring Server Groups

Perform the tasks as outlined in the Configuring Server Groups.

Configuring Authentication for Per VRF AAA with Remote Customer Templates

Perform this task to configure authentication for Per VRF AAA with remote customer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2*...]
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2*...] **Example:** `Router(config)# ppp authentication ppp default group radius`	Specifies one or more authentication, authorization, and accounting (AAA) authentication methods for use on serial interfaces that are running PPP.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...] **Example:** `Router(config)# aaa authorization network default group sp`	Sets parameters that restrict user access to a network.

Configuring Authorization for Per VRF AAA with Remote Customer Templates

Perform this task to configure authorization for Per VRF AAA with remote customer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables use of local or remote templates.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...] **Example:** `Router(config)# aaa authorization network default sp`	Specifies the server group that is named as the default method for authorization.

Configuring the RADIUS Profile on the SP RADIUS Server

See the Per VRF AAA Using a Remote RADIUS Customer Template Example for an example of how to update the RADIUS profile.

Verifying VRF Routing Configurations

Perform this task to verify VRF routing configurations:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**show ip route vrf** *vrf-name* **Example:** `Router(config)# show ip route vrf northvrf`	Displays the IP routing table associated with a VRF.

Troubleshooting Per VRF AAA Configurations

To troubleshoot the Per VRF AAA feature, use at least one of the following commands in EXEC mode:

Command	Purpose
`Router#` **debug aaa accounting**	Displays information on accountable events as they occur.
`Router#` **debug aaa authentication**	Displays information on AAA authentication.
`Router#` **debug aaa authorization**	Displays information on AAA authorization.
`Router#` **debug ppp negotiation**	Displays information on traffic and exchanges in an internetwork implementing PPP.
`Router#` **debug radius**	Displays information associated with RADIUS.
`Router#` **debug vpdn event**	Displays Layer 2 Transport Protocol (L2TP) errors and events that are a part of normal tunnel establishment or shutdown for VPNs.
`Router#` **debug vpdn error**	Displays debug traces for VPN.

Configuration Examples for Per VRF AAA

Per VRF Configuration Examples

Per VRF AAA Example

The following example shows how to configure the Per VRF AAA feature using a AAA server group with associated private servers:

```
aaa new-model
aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com
aaa accounting delay-start vrf v1.55.com
aaa accounting send stop-record authentication failure vrf v1.55.com
aaa group server radius v1.55.com
    server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
    ip vrf forwarding v1.55.com
ip radius source-interface loopback55
radius-server attribute 44 include-in-access-req vrf v1.55.com
```

Per VRF AAA Using a Locally Defined Customer Template Example

The following example shows how to configure the Per VRF AAA feature using a locally defined customer template with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa authorization network default local
aaa authorization template
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com
aaa group server radius V1_55_com
  server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
  ip vrf forwarding V1.55.com
template V1.55.com
  peer default ip address pool V1_55_com_pool
  ppp authentication chap callin V1_55_com
  ppp authorization V1_55_com
  ppp accounting V1_55_com
  aaa accounting delay-start
  aaa accounting send stop-record authentication failure
  radius-server attribute 44 include-in-access-req
  ip vrf forwarding v1.55.com
  ip radius source-interface Loopback55
```

Per VRF AAA Using a Remote RADIUS Customer Template Example

The following examples shows how to configure the Per VRF AAA feature using a remotely defined customer template on the SP RADIUS server with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization template
aaa authorization network default group sp
aaa group server radius sp
```

```
   server 10.3.3.3
radius-server host 10.3.3.3 auth-port 1645 acct-port 1646 key sp_key
```

The following RADIUS server profile is configured on the SP RADIUS server:

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

Customer Template Examples

Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example

The following example shows how to create a locally configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authentication ppp V1_55_com group V1_55_com
aaa authorization template
aaa authorization network default local group radius
aaa authorization network V1_55_com group V1_55_com
aaa accounting network V1_55_com start-stop broadcast group V1_55_com group SP_AAA_server
aaa group server radius SP_AAA_server
 server 10.10.100.7 auth-port 1645 acct-port 1646
aaa group server radius V1_55_com
 server-private 10.10.132.4 auth-port 1645 acct-port 1646
 authorization accept min-author
 accounting accept usage-only
 ip vrf forwarding V1.55.com
ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55
template V1.55.com
 peer default ip address pool V1.55-pool
 ppp authentication chap callin V1_55_com
 ppp authorization V1_55_com
 ppp accounting V1_55_com
 aaa accounting delay-start
 aaa accounting send stop-record authentication failure
 radius-server attribute 44 include-in-access-req
vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
```

```
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41
interface Virtual-Template13
 ip vrf forwarding V1.55.com
 ip unnumbered Loopback55
 ppp authentication chap callin
 ppp multilink
ip local pool V1.55-pool 10.1.55.10 10.1.55.19 group V1.55-group
ip radius source-interface Loopback0
ip radius source-interface Loopback55 vrf V1.55.com
radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46
radius-server host 10.10.100.7 auth-port 1645 acct-port 1646 key ww
radius-server host 10.10.132.4 auth-port 1645 acct-port 1646 key ww
```

Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example

The following example shows how to create a remotely configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authorization template
aaa authorization network default local group radius
ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55
vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41
interface Virtual-Template13
 no ip address
 ppp authentication chap callin
 ppp multilink
ip local pool V1.55-pool 10.1.55.10 10.1.55.19 group V1.55-group
radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46
```

The customer template is stored as a RADIUS server profile for v1.55.com.

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "aaa:rad-serv#2=10.10.100.7 key ww"
cisco-avpair = "aaa:rad-serv-source-if#2=Loopback 0"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1 group 2 broadcast"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "aaa:rad-serv-filter#1=authorization accept min-author"
cisco-avpair = "aaa:rad-serv-filter#1=accounting accept usage-only"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

AAA Accounting Stop Records Examples

The following AAA accounting stop record examples show how to configure the **aaa accounting send stop-record authentication** command to control the generation of “stop” records when the **aaa accounting** command is issued with the **start-stop** or **stop-only** keyword.

Note The **success** and **remote-server** keywords are available in Cisco IOS Release 12.4(2)T and later releases.

AAA Accounting Stop Record and Successful Call Example

The following example shows “start” and “stop” records being sent for a successful call when the **aaa accounting send stop-record authentication** command is issued with the **failure** keyword.

```
Router# show running-config | include aaa

.
.
.
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting send stop-record authentication failure
aaa accounting network default start-stop group radius
.
.
.
*Jul  7 03:28:31.543: AAA/BIND(00000018): Bind i/f Virtual-Template2
*Jul  7 03:28:31.547: ppp14 AAA/AUTHOR/LCP: Authorization succeeds trivially
*Jul  7 03:28:33.555: AAA/AUTHOR (0x18): Pick method list 'default'
*Jul  7 03:28:33.555: AAA/BIND(00000019): Bind i/f
*Jul  7 03:28:33.555:  Tnl 5192 L2TP: O SCCRQ
*Jul  7 03:28:33.555:  Tnl 5192 L2TP: O SCCRQ, flg TLS, ver 2, len 141, tnl 0,
ns 0, nr 0
         C8 02 00 8D 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
         00 06 11 30 80 10 00 00 00 07 4C 41 43 2D 74 75
         6E 6E 65 6C 00 19 00 00 00 08 43 69 73 63 6F 20
         53 79 73 74 65 6D 73 ...
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse SCCRP
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 2, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Protocol Ver 256
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 3, len 10, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Framing Cap 0x0
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 4, len 10, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Bearer Cap 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 6, len 8, flag 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Firmware Ver 0x1120
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 7, len 16, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Hostname LNS-tunnel
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 8, len 25, flag 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Vendor Name Cisco Systems, Inc.
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 9, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Assigned Tunnel ID 6897
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 10, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Rx Window Size 20050
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 11, len 22, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Chlng
         81 13 03 F6 A8 E4 1D DD 25 18 25 6E 67 8C 7C 39
```

```
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 13, len 22, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Chlng Resp
        4D 52 91 DC 1A 43 B3 31 B4 F5 B8 E1 88 22 4F 41
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: No missing AVPs in SCCRP
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: I SCCRP, flg TLS, ver 2, len 157, tnl
5192, ns 0, nr 1
contiguous pak, size 157
        C8 02 00 9D 14 48 00 00 00 00 00 01 80 08 00 00
        00 00 00 02 80 08 00 00 00 02 01 00 80 0A 00 00
        00 03 00 00 00 00 80 0A 00 00 00 04 00 00 00 00
        00 08 00 00 00 06 11 20 80 10 00 00 00 07 4C 4E
        53 2D 74 75 6E 6E 65 6C ...
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: I SCCRP from LNS-tunnel
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: O SCCCN  to LNS-tunnel tnlid 6897
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: O SCCCN, flg TLS, ver 2, len 42, tnl
6897, ns 1, nr 1
        C8 02 00 2A 1A F1 00 00 00 01 00 01 80 08 00 00
        00 00 00 03 80 16 00 00 00 0D 32 24 17 BC 6A 19
        B1 79 F3 F9 A9 D4 67 7D 9A DB
*Jul  7 03:28:33.571: uid:14 Tnl/Sn 5192/11 L2TP: O ICRQ to LNS-tunnel 6897/0
*Jul  7 03:28:33.571: uid:14 Tnl/Sn 5192/11 L2TP: O ICRQ, flg TLS, ver 2, len
63, tnl 6897, lsid 11, rsid 0, ns 2, nr 1
        C8 02 00 3F 1A F1 00 00 00 02 00 01 80 08 00 00
        00 00 00 0A 80 0A 00 00 00 0F C8 14 B4 03 80 08
        00 00 00 0E 00 0B 80 0A 00 00 00 12 00 00 00 00
        00 0F 00 09 00 64 0F 10 09 02 02 00 1B 00 00
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse  AVP 0, len 8, flag
0x8000 (M)
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse ICRP
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse  AVP 14, len 8, flag
0x8000 (M)
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Assigned Call ID 5
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: No missing AVPs in ICRP
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: I ICRP, flg TLS, ver 2, len
28, tnl 5192, lsid 11, rsid 0, ns 1, nr 3
contiguous pak, size 28
        C8 02 00 1C 14 48 00 0B 00 01 00 03 80 08 00 00
        00 00 00 0B 80 08 00 00 00 0E 00 05
*Jul  7 03:28:33.579: uid:14 Tnl/Sn 5192/11 L2TP: O ICCN to LNS-tunnel 6897/5
*Jul  7 03:28:33.579: uid:14 Tnl/Sn 5192/11 L2TP: O ICCN, flg TLS, ver 2, len
167, tnl 6897, lsid 11, rsid 5, ns 3, nr 2
        C8 02 00 A7 1A F1 00 05 00 03 00 02 80 08 00 00
        00 00 00 0C 80 0A 00 00 00 18 06 1A 80 00 00 0A
        00 00 00 26 06 1A 80 00 80 0A 00 00 00 13 00 00
        00 01 00 15 00 00 00 1B 01 04 05 D4 03 05 C2 23
        05 05 06 0A 0B E2 7A ...
*Jul  7 03:28:33.579: RADIUS/ENCODE(00000018):Orig. component type = PPoE
*Jul  7 03:28:33.579: RADIUS(00000018): Config NAS IP: 10.0.0.0
*Jul  7 03:28:33.579: RADIUS(00000018): sending
*Jul  7 03:28:33.579: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:28:33.579: RADIUS(00000018): Send Accounting-Request to
172.19.192.238:2196 id 1646/23, len 176
*Jul  7 03:28:33.579: RADIUS:  authenticator 3C 81 D6 C5 2B 6D 21 8E - 19 FF
43 B5 41 86 A8 A5
*Jul  7 03:28:33.579: RADIUS:  Acct-Session-Id     [44]  10  "00000023"
*Jul  7 03:28:33.579: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:28:33.579: RADIUS:  Tunnel-Medium-Type  [65]  6
00:IPv4                   [1]
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Client-Endpoi[66]  10  "10.0.0.1"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Server-Endpoi[67]  10  "10.0.0.2"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Assignment-Id[82]  5   "lac"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Type         [64]  6
00:L2TP                   [3]
*Jul  7 03:28:33.583: RADIUS:  Acct-Tunnel-Connecti[68]  12  "3356800003"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Client-Auth-I[90]  12  "LAC-tunnel"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Server-Auth-I[91]  12  "LNS-tunnel"
*Jul  7 03:28:33.583: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:28:33.583: RADIUS:  Acct-Authentic      [45]  6
Local                     [2]
*Jul  7 03:28:33.583: RADIUS:  Acct-Status-Type    [40]  6
Start                     [1]
```

```
*Jul  7 03:28:33.583: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:28:33.583: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:28:33.583: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:28:33.583: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:28:33.583: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:28:33.583: RADIUS:  Acct-Delay-Time     [41]  6
0
*Jul  7 03:28:33.683: RADIUS: Received from id 1646/23 172.19.192.238:2196,
Accounting-response, len 20
*Jul  7 03:28:33.683: RADIUS:  authenticator 1C E9 53 42 A2 8A 58 9A - C3 CC
1D 79 9F A4 6F 3A
```

AAA Accounting Stop Record and Rejected Call Example

The following example shows the “stop” record being sent for a rejected call during authentication when the **aaa accounting send stop-record authentication**command is issued with the **success** keyword.

```
Router# show running-config | include aaa
.
.
.
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting send stop-record authentication success remote-server
aaa accounting network default start-stop group radius
Router#
*Jul  7 03:39:40.199: AAA/BIND(00000026): Bind i/f Virtual-Template2
*Jul  7 03:39:40.199: ppp21 AAA/AUTHOR/LCP: Authorization succeeds trivially
*Jul  7 03:39:42.199: RADIUS/ENCODE(00000026):Orig. component type = PPoE
*Jul  7 03:39:42.199: RADIUS:  AAA Unsupported     [156] 7
*Jul  7 03:39:42.199: RADIUS:   30 2F 30 2F
30                                [0/0/0]
*Jul  7 03:39:42.199: RADIUS(00000026): Config NAS IP: 10.0.0.0
*Jul  7 03:39:42.199: RADIUS/ENCODE(00000026): acct_session_id: 55
*Jul  7 03:39:42.199: RADIUS(00000026): sending
*Jul  7 03:39:42.199: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:39:42.199: RADIUS(00000026): Send Access-Request to
172.19.192.238:2195 id 1645/14, len 94
*Jul  7 03:39:42.199: RADIUS:  authenticator A6 D1 6B A4 76 9D 52 CF - 33 5D
16 BE AC 7E 5F A6
*Jul  7 03:39:42.199: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.199: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:39:42.199: RADIUS:  CHAP-Password       [3]   19  *
*Jul  7 03:39:42.199: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:39:42.199: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:39:42.199: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:39:42.199: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.199: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:39:42.271: RADIUS: Received from id 1645/14 172.19.192.238:2195,
Access-Accept, len 194
*Jul  7 03:39:42.271: RADIUS:  authenticator 30 AD FF 8E 59 0C E4 6C - BA 11
23 63 81 DE 6F D7
*Jul  7 03:39:42.271: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.275: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  26
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   20  "vpdn:tunnel-
id=lac"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  29
```

```
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   23  "vpdn:tunnel-
type=l2tp"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  30
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   24  "vpdn:gw-
password=cisco"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  31
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   25  "vpdn:nas-
password=cisco"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  34
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   28  "vpdn:ip-
addresses=10.0.0.2"
*Jul  7 03:39:42.275: RADIUS:  Service-Type        [6]   6
Framed                   [2]
*Jul  7 03:39:42.275: RADIUS:  Framed-Protocol     [7]   6
PPP                      [1]
*Jul  7 03:39:42.275: RADIUS(00000026): Received from id 1645/14
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: Framed-Protocol
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: service-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: tunnel-id
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: tunnel-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: gw-password
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: nas-password
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: ip-addresses
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: service-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: Framed-Protocol
*Jul  7 03:39:42.279: AAA/BIND(00000027): Bind i/f
*Jul  7 03:39:42.279:  Tnl 21407 L2TP: O SCCRQ
*Jul  7 03:39:42.279:  Tnl 21407 L2TP: O SCCRQ, flg TLS, ver 2, len 134, tnl
0, ns 0, nr 0
         C8 02 00 86 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
         00 06 11 30 80 09 00 00 00 07 6C 61 63 00 19 00
         00 00 08 43 69 73 63 6F 20 53 79 73 74 65 6D 73
         2C 20 49 6E 63 2E 80 ...
*Jul  7 03:39:49.279:  Tnl 21407 L2TP: O StopCCN
*Jul  7 03:39:49.279:  Tnl 21407 L2TP: O StopCCN, flg TLS, ver 2, len 66, tnl
0, ns 1, nr 0
         C8 02 00 42 00 00 00 00 00 01 00 00 80 08 00 00
         00 00 00 04 80 1E 00 00 00 01 00 02 00 06 54 6F
         6F 20 6D 61 6E 79 20 72 65 74 72 61 6E 73 6D 69
         74 73 00 08 00 09 00 69 00 01 80 08 00 00 00 09
         53 9F
*Jul  7 03:39:49.279: RADIUS/ENCODE(00000026):Orig. component type = PPoE
*Jul  7 03:39:49.279: RADIUS(00000026): Config NAS IP: 10.0.0.0
*Jul  7 03:39:49.279: RADIUS(00000026): sending
*Jul  7 03:39:49.279: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:39:49.279: RADIUS(00000026): Send Accounting-Request to
172.19.192.238:2196 id 1646/32, len 179
*Jul  7 03:39:49.279: RADIUS:  authenticator 0A 85 2F F0 65 6F 25 E1 - 97 54
CC BF EA F7 62 89
*Jul  7 03:39:49.279: RADIUS:  Acct-Session-Id     [44]  10  "00000037"
*Jul  7 03:39:49.279: RADIUS:  Framed-Protocol     [7]   6
PPP                      [1]
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Medium-Type  [65]  6
00:IPv4                  [1]
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Client-Endpoi[66]  10  "10.0.0.1"
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Server-Endpoi[67]  10  "10.0.0.2"
*Jul  7 03:39:49.283: RADIUS:  Tunnel-Type         [64]  6
00:L2TP                  [3]
*Jul  7 03:39:49.283: RADIUS:  Acct-Tunnel-Connecti[68]  3   "0"
*Jul  7 03:39:49.283: RADIUS:  Tunnel-Client-Auth-I[90]  5   "lac"
*Jul  7 03:39:49.283: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:39:49.283: RADIUS:  Acct-Authentic      [45]  6
RADIUS                   [1]
*Jul  7 03:39:49.283: RADIUS:  Acct-Session-Time   [46]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Input-Octets   [42]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Output-Octets  [43]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Input-Packets  [47]  6
0
```

```
*Jul  7 03:39:49.283: RADIUS:  Acct-Output-Packets [48]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Terminate-Cause[49]  6   nas-
error                 [9]
*Jul  7 03:39:49.283: RADIUS:  Acct-Status-Type    [40]  6
Stop                  [2]
*Jul  7 03:39:49.283: RADIUS:  NAS-Port-Type       [61]  6
Virtual               [5]
*Jul  7 03:39:49.283: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:39:49.283: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:39:49.283: RADIUS:  Service-Type        [6]   6
Framed                [2]
*Jul  7 03:39:49.283: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:39:49.283: RADIUS:  Acct-Delay-Time     [41]  6
0
*Jul  7 03:39:49.335: RADIUS: Received from id 1646/32 172.19.192.238:2196,
Accounting-response, len 20
*Jul  7 03:39:49.335: RADIUS:  authenticator C8 C4 61 AF 4D 9F 78 07 - 94 2B
44 44 17 56 EC 03
```

Additional References

The following sections provide references related to Per VRF AAA.

Related Documents

Related Topic	Document Title
AAA: Configuring Server Groups	*Cisco IOS Security Configuration Guide: Securing User Services* , Release 12.4T
Cisco IOS Security Commands	*Cisco IOS Security Command Reference*
Cisco IOS Switching Services Commands	*Cisco IOS IP Switching Command Reference*
Configuring Multiprotocol Label Switching	*Cisco IOS Multiprotocol Label Switching Configuration Guide* , Release 12.4T
Configuring Virtual Templates section	Virtual Templates, Profiles, and Networks chapter in the *Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Per VRF AAA

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 25 Feature Information for Per VRF AAA

Feature Name	Releases	Feature Information
Per VRF AAA Dynamic Per VRF AAA Attribute Filtering Per-Domain and VRF Aware Framed-Routes RADIUS Per-VRF Server Group	12.2(1)DX 12.2(2)DD 12.2(4)B 12.2(13)T 12.2(15)T 12.4(2)T 12.2(28)SB 12.2(33)SR 12.2(33)SXI 12.2(33)SXH4	The Per VRF AAA feature allows authentication, authorization, and accounting (AAA) on the basis of Virtual Private Network (VPN) routing and forwarding (VRF) instances. For Cisco IOS Release 12.2(15)T or later releases, you can use a customer template, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template. In 12.2(1)DX, the Per VRF AAA feature was introduced on the Cisco 7200 series and the Cisco 7401ASR. In 12.2(2)DD, the **ip vrf forwarding** (server-group) and **radius-server domain-stripping** commands were added. The Per VRF AAA, Dynamic Per VRF AAA, and Attribute Filtering Per-Domain and VRF Aware Framed-Routes features were introduced in Cisco IOS Release 12.2(15)T. Also, the **aaa authorization template** command was added to this release. In 12.4(2)T, the **aaa accounting send stop-record authentication** command was updated with additional support for AAA accounting stop records. In 12.2(33)SRC, RADIUS Per-VRF Server Group feature was introduced. In Cisco IOS Release 12.2(33)SXI, these features were introduced. In Cisco IOS Release 12.2(33)SXH4, these features were introduced. The following commands were introduced or modified: **aaa**

Feature Name	Releases	Feature Information
		accounting, **aaa accounting delay-start**, **ip radius source-interface**, **radius-server attribute 44 include-in-access-req**, **server-private (RADIUS)**.

Glossary

AAA --authentication, authorization, and accounting. A framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

L2TP --Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

PE --Provider Edge. Networking devices that are located on the edge of a service provider network.

RADIUS --Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VPN --Virtual Private Network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the LNS instead of the LAC.

VRF --Virtual Route Forwarding. Initially, a router has only one global default routing/forwarding table. VRFs can be viewed as multiple disjoined routing/forwarding tables, where the routes of a user have no correlation with the routes of another user.