

Cisco IOS Firewall Authentication Proxy

This feature module describes the Cisco IOS Firewall Authentication Proxy feature. It includes information on the benefits of the feature, supported platforms, configuration tasks, and so forth.

This document includes the following sections:

- Feature Overview on page 1
- Supported Platforms on page 10
- Supported Standards, MIBs, and RFCs on page 10
- Prerequisites on page 10
- Configuration Tasks on page 11
- Monitoring and Maintaining the Authentication Proxy on page 16
- Configuration Examples on page 17
- Command Reference on page 30
- Debug Commands on page 44

Feature Overview

The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access was associated with a user's IP address, or a single security policy had to be applied to an entire user group or subnet. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

With the authentication proxy feature, users can log into the network or access the Internet via HTTP, and their specific access profiles are automatically retrieved and applied from a CiscoSecure ACS, or other RADIUS, or TACACS+ authentication server. The user profiles are active only when there is active traffic from the authenticated users.

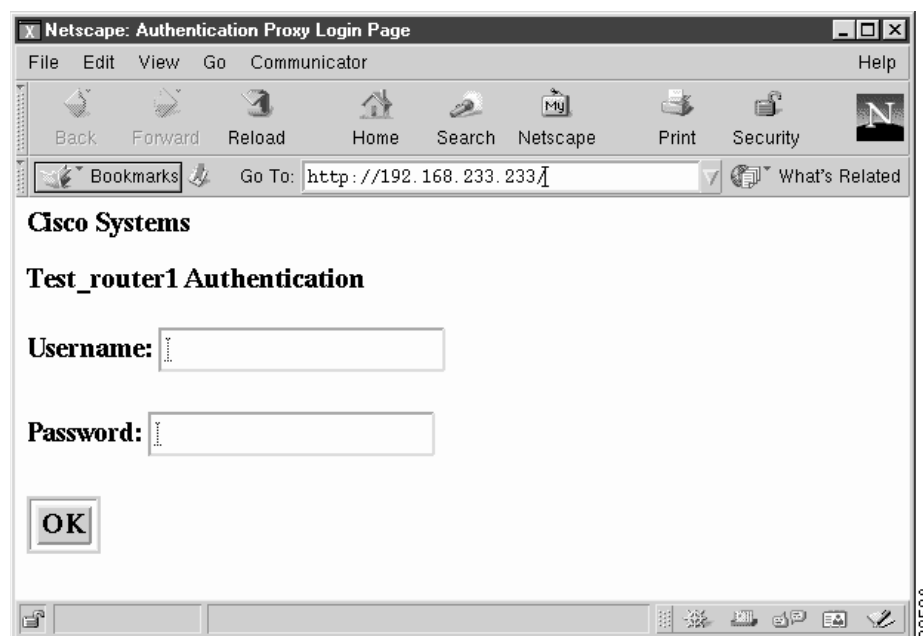
The authentication proxy is compatible with other Cisco IOS security features such as Network Address Translation (NAT), Context-based Access Control (CBAC), IP Security (IPSec) encryption, and VPN client software.

How the Authentication Proxy Works

When a user initiates an HTTP session through firewall, it triggers the authentication proxy. The authentication proxy first checks to see if the user has been authenticated. If a valid authentication entry exists for the user, the connection is completed with no further intervention by the authentication proxy. If no entry exists, the authentication proxy responds to the HTTP connection request by prompting the user for a username and password.

Figure 1 illustrates the authentication proxy HTML login page.

Figure 1 Authentication Proxy Login Page



Users must successfully authenticate with the authentication server by entering a valid username and password.

If the authentication succeeds, the user's authorization profile is retrieved from the AAA server. The authentication proxy uses the information in this profile to create dynamic access control entries (ACEs) and add them to the inbound (input) access control list (ACL) of an input interface, and to the outbound (output) ACL of an output interface, if an output ACL exists at the interface. By doing this, the firewall allows authenticated users access to the network as permitted by the authorization profile. For example, a user can initiate a Telnet connection through the firewall if Telnet is permitted in the user's profile.

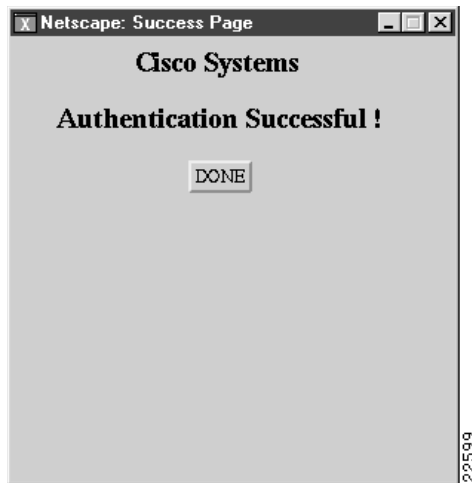
If the authentication fails, the authentication proxy reports the failure to the user, and prompts the user with multiple retries. If the user fails to authenticate after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

The login page is refreshed each time the user makes requests to access information from a web server.

The authentication proxy customizes each of the access list entries in the user profile by replacing the source IP addresses in the downloaded access list with the source IP address of the authenticated host.

At the same time that dynamic ACEs are added to the interface configuration, the authentication proxy sends a message to the user confirming that the login was successful. Figure 2 illustrates the login status in the HTML page.

Figure 2 Authentication Proxy Login Status Message



The authentication proxy sets up an inactivity (idle) timer for each user profile. As long as there is activity through the firewall, new traffic initiated from the user's host does not trigger the authentication proxy, and authorized user traffic is permitted access through the firewall.

If the idle timer expires, the authentication proxy removes the user's profile information and dynamic access lists entries. When this happens, traffic from the client host is blocked. The user must initiate another HTTP connection to trigger the authentication proxy.

Secure Authentication

The authentication proxy uses JavaScript to help achieve secure authentication using the client browser. Secure authentication prevents a client from mistakenly submitting a username and password to a network web server other than the authentication proxy router.

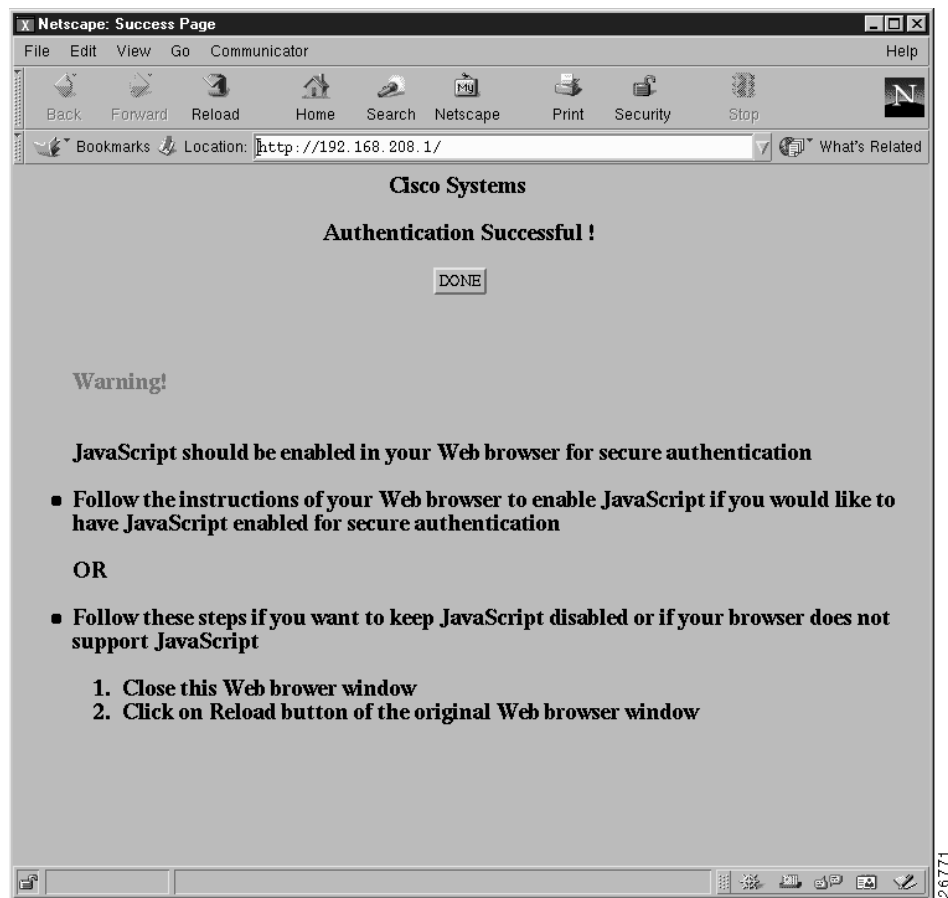
Operation with JavaScript

Users should enable JavaScript on the browser prior to initiating an HTTP connection. With JavaScript enabled on the browser, secure authentication is done automatically, and the user sees the authentication message shown in Figure 2. The HTTP connection is completed automatically for the user.

Operation without JavaScript

If the client browser does not support JavaScript, or if site security policy prevents users from enabling JavaScript, any login attempt generates a popup window with instructions for manually completing the connection. Figure 3 illustrates the authentication proxy login status message with JavaScript disabled on the browser.

Figure 3 Authentication Proxy Login Status Message with JavaScript Disabled



To close this window, click Close on the browser File menu.

After closing the popup window, the user should click Reload (Refresh for Internet Explorer) on the browser window in which the authentication login page is displayed. If the user's last authentication attempt succeeds, clicking Reload brings up the web page the user is trying to retrieve. If the user's last attempt fails, clicking Reload causes the authentication proxy to intercept the client HTTP traffic again, prompting the user with another login page soliciting the username and password.

If JavaScript is not enabled, it is strongly recommended that site administrators advise users of the correct procedure for closing the popup window as described in "Establishing User Connections with JavaScript Disabled" on page 15.

Using the Authentication Proxy

Unlike some Cisco IOS Firewall features that operate transparently to the user, the authentication proxy feature requires some user interaction on the client host. Table 1 describes the interaction of the authentication proxy with the client host.

Table 1 Authentication Proxy Interaction with the Client Host

Authentication Proxy Action with Client	Description
Triggering on HTTP connections	If a user is not currently authenticated at the firewall router, any HTTP connection initiated by the user triggers the authentication proxy. If the user is already authenticated, the authentication proxy is transparent to the user.
Logging in using the login page	Triggering the authentication proxy generates an HTML-based login page. The user must enter a username and password to authenticate with the AAA server. Figure 1 illustrates the authentication proxy login page.
Authenticating the user at the client	<p>Following the login attempt, the authentication proxy action can vary depending on whether JavaScript is enabled in the browser. If JavaScript is enabled, and authentication is successful, the authentication proxy displays a message indicating the status of the authentication as shown in Figure 2. After displaying the authentication status, the proxy automatically completes the HTTP connection.</p> <p>If JavaScript is disabled, and authentication is successful, the authentication proxy generates a popup window with additional instructions for completing the connection. See Figure 3.</p> <p>If authentication is unsuccessful in any case, the user must log in again from the login page.</p>

When to Use the Authentication Proxy

Here are a few examples of when you might use the authentication proxy:

- You want to manage access privileges on a per-user basis using the services provided by the authentication servers instead of configuring access control based on host IP address or global access policies.
- You want to authenticate and authorize local users before permitting access to intranet or Internet services or hosts through the firewall.
- You want to authenticate and authorize remote users before permitting access to local services or hosts through the firewall.
- You want to control access for specific extranet users. For example, you might want to authenticate and authorize the financial officer of a corporate partner with one set of access privileges, while authorizing the technology officer for that same partner to use another set of access privileges.
- You want to use the authentication proxy in conjunction with VPN client software to validate users and to assign specific access privileges.

Applying the Authentication Proxy

Apply the authentication proxy in the inbound direction at any interface on the router where you want per-user authentication and authorization. Applying the authentication proxy inbound at an interface causes it to intercept a user's initial connection request before that request is subjected to any other processing by the firewall. If the user fails to authenticate with the AAA server, the connection request is dropped.

How you apply the authentication proxy depends on your security policy. For example, you can block all traffic through an interface, and enable the authentication proxy feature to require authentication and authorization for all user initiated HTTP connections. Users are authorized for services only after successful authentication with the AAA server.

The authentication proxy feature also allows you to use standard access lists to specify a host or group of hosts whose initial HTTP traffic triggers the proxy.

Figure 4 shows the authentication proxy applied at the LAN interface with all network users required to authenticate upon the initial connection (all traffic is blocked at each interface).

Figure 4 Applying the Authentication Proxy at the Local Interface

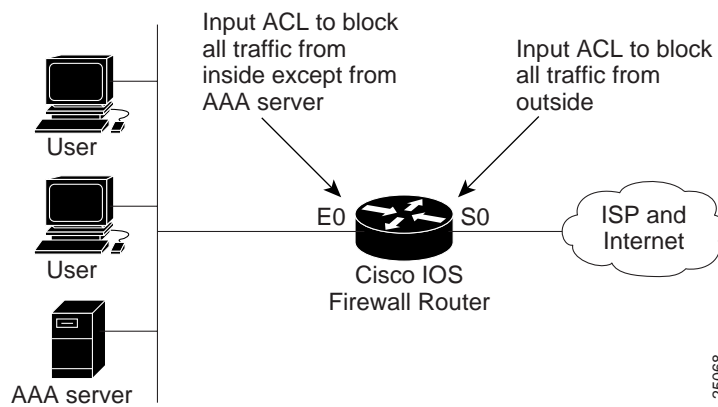
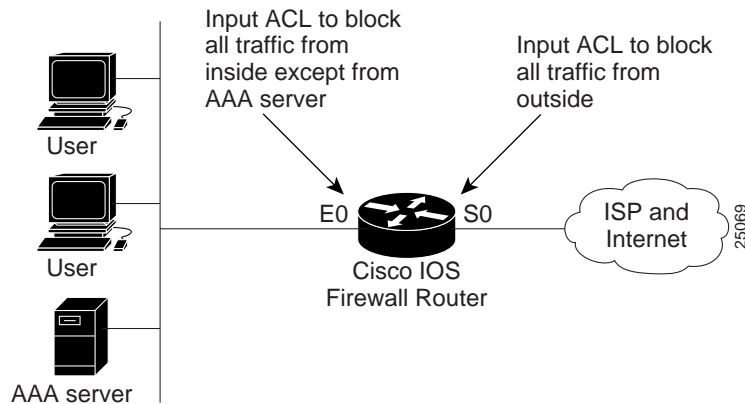


Figure 5 shows the authentication proxy applied at the dial-in interface with all network traffic blocked at each interface.

Figure 5 Applying the Authentication Proxy at an Outside Interface

Operation with One-time Passwords

Using a one-time password, users enter the username and one-time password in the HTML log-in page as usual.

Users must enter the correct token password within the first three attempts. After three incorrect entries, the user must enter two valid token passwords in succession before authentication is granted with the AAA server.

Compatibility with Other Security Features

The authentication proxy is compatible with Cisco IOS software and with Cisco IOS security features:

- Intrusion Detection System (IDS)
- NAT
- CBAC
- IPSec encryption
- VPN client software

The authentication proxy works transparently with the IDS and IPSec encryption features. The following sections describe the relationship of the NAT, CBAC, and VPN client software features with the authentication proxy.

NAT Compatibility

The authentication proxy feature is compatible with NAT; however, to run successfully with NAT, you must configure CBAC.

For example, when dynamic NAT is configured, the client's IP address might be translated to different addresses during the time the user is authenticated with the authentication proxy. Assume that the client is running a HTTP session. The user's original IP address is 10.1.1.1, which is

translated by NAT to 192.168.2.2. NAT guarantees that during this session, 10.1.1.1 is always translated to 192.168.2.2. When the user is first authenticated, a set of dynamic ACEs is created to support the user. Subsequent sessions can use different NAT addresses, which are not covered by the original dynamic ACEs created by the authentication proxy. In this case, it is strongly recommended that you configure CBAC to take care of the translated addresses and to create the matching ACEs.

CBAC ensures that the translated address for the session is associated with the original host address.

CBAC Compatibility

To run successfully in all configurations, and to ensure return traffic for authorized user connections is permitted through the firewall, configure CBAC with the authentication proxy.

Because the authentication proxy does not create ACEs to support return traffic or data channel traffic, you must either create static ACLs to allow the return traffic or configure CBAC inspection rules in the firewall configuration.

VPN Client Compatibility

Using the authentication proxy, network administrators can apply an extra layer of security and access control for VPN client traffic. If a VPN client initiates an HTTP connection, the authentication proxy first checks for prior client authentication. If the client is authenticated, authorized traffic is permitted. If the client is not authenticated, the HTTP request triggers the authentication proxy, and the user is prompted for a username and password.

If the user authentication is successful, the authentication proxy retrieves the user profile from the AAA server. The source address in the user profiles entries is replaced with the IP address of the authenticated VPN client from the decrypted packet.

Protection Against Denial-of-Service Attacks

The authentication proxy monitors the level of incoming HTTP requests. For each request, the authentication proxy prompts for the user's login credentials. A high number of open requests could indicate that the router is the subject of a denial-of-service (DoS) attack. The authentication proxy limits the level of open requests and drops additional requests until the number of open requests has dropped below 40.

If the firewall is experiencing a high level of connection requests requiring authentication, legitimate network users might experience delays when making connections, or the connection might be rejected and the user must try the connection again.

Risk of Spoofing with Authentication Proxy

When the authentication proxy is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface with user access privileges. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. The authentication proxy does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and the authentication proxy does not specifically address this problem.

Comparison with the Lock-and-Key Feature

Lock-and-key is another Cisco IOS Firewall feature that uses authentication and dynamic access list to provide user access through the firewall. Table 2 compares the authentication proxy and Lock-and-key features.

Table 2 Comparison of the Authentication Proxy and Lock-and-Key Features

Lock-and-Key	Authentication Proxy
Triggers on Telnet connection requests.	Triggers on HTTP connection requests.
TACACS+, RADIUS, or local authentication.	TACACS+ or RADIUS authentication and authorization.
Access lists are configured on the router only.	Access lists are retrieved from the AAA server only.
Access privileges are granted based on the user's host IP address.	Access privileges are granted on a per-user and host IP address basis.
Access lists are limited to one entry for each host IP address.	Access lists can have multiple entries as defined by the user profiles on the AAA server.
Associates a fixed IP addresses with a specific user. Users must log in from the host with that IP address.	Allows DHCP-based host IP addresses, meaning that users can log in from any host location and obtain authentication and authorization.

Use the authentication proxy in any network environment that provides a per-user security policy. Use Lock-and-key in network environments that might benefit from local authentication and a limited number of router-based access control policies based on host addresses. Use Lock-and-key in environments not using the Cisco IOS Firewall.

Benefits

- Provides dynamic, per-user authentication and authorization, authenticating users against industry standard TACACS+ and RADIUS authentication protocols.
- Authenticates and authorizes users, thus providing more robust protection against network attacks.
- Authenticates and authorizes users from any host IP address, allowing network administrators to configure host IP addresses using DHCP.
- Allows network managers to set individual, per-user security policy.
- Applies authentication and authorization to intranet, extranet, Internet, and VPN client users.
- Requires no special client features or software, providing transparent client operation using commonly available desktop browsers.

Restrictions

- The authentication proxy triggers only on HTTP connections.
- HTTP services must be running on the standard (well-known) port, which is port 80 for HTTP.
- The authentication proxy does not support AAA accounting in this release.
- Client browsers must enable JavaScript for secure authentication.

- The authentication proxy access lists apply to traffic passing through the router. Traffic destined to the router is authenticated by the existing authentication methods provided by Cisco IOS software.
- The authentication proxy does not support concurrent usage; that is, if two users try to log in from the same host at the same time, authentication and authorization applies only to the user who first submits a valid username and password.
- Load balancing using multiple or different AAA servers is not supported.

Supported Platforms

- Cisco 2600 series
- Cisco 3600 series
- Cisco 7100
- Cisco 7200 series

Additional platform support is planned for future Cisco IOS software releases.

Supported Standards, MIBs, and RFCs

MIBs

No new or modified MIBs are supported by this feature.

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

RFCs

No new or modified RFCs are supported by this feature.

Standards

No new or modified standards are supported by this feature.

Prerequisites

Client Browser

For the authentication proxy to work properly, the client browser must be running the following browser software:

- Microsoft Internet Explorer 3.0 or later
- Netscape Navigator 3.0 or later

Standard Access Lists

The authentication proxy has an option to use standard access lists. You must have a solid understanding of how access lists are used to filter traffic before you attempt to configure the authentication proxy. For an overview of how to use access lists with the Cisco IOS Firewall, refer to the chapter, “Access Control Lists: Overview and Guidelines,” in the Cisco IOS Release 12.0 *Security Configuration Guide*.

AAA Services

The authentication proxy employs user authentication and authorization as implemented in Cisco's authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication and authorization before you configure the authentication proxy. User authentication and authorization is explained in the chapter, “Authentication, Authorization, and Accounting (AAA),” in the Cisco IOS Release 12.0 *Security Configuration Guide*.

CBAC

To run the authentication proxy successfully with the Cisco IOS Firewall, configure CBAC on the firewall. For complete information on the CBAC feature, refer to “Cisco IOS Firewall Feature Set” in the Cisco IOS Release 12.0 New Features section on Cisco Connection Online (CCO).

Configuration Tasks

To configure the authentication proxy feature, perform the following tasks:

- Configuring AAA (Required)
- Configuring the HTTP Server (Required)
- Configuring the Authentication Proxy (Required)
- Verifying the Authentication Proxy (Optional)

Configuring AAA

You must configure the authentication proxy for AAA services. Use the following commands in global configuration mode to enable authorization and to define the authorization methods:

Step	Command	Purpose
1	<code>router(config)# aaa new-model</code>	Use this command to enable the AAA functionality on the router.
2	<code>router(config)# aaa authentication login default TACACS+ RADIUS</code>	Define the list of authentication methods at login.
3	<code>router(config)# aaa authorization auth-proxy default [method1 [method2...]]</code>	Use the auth-proxy keyword to enable authentication proxy for AAA methods.
4	<code>router(config)# tacacs-server host hostname</code>	Specify an AAA server. For RADIUS servers, use the radius server host command.
5	<code>router(config)# tacacs-server key sting</code>	Set the authentication and encryption key for communications between the router and the AAA server. For RADIUS servers use the radius server key command.
6	<code>router(config)# access-list access-list-number permit tcp host source eq tacacs host destination</code>	Create an ACL entry to allow the AAA server return traffic to the firewall. The source address is the IP address of the AAA server, and the destination is the IP address of the router interface where the AAA server resides.

In addition to configuring AAA on the firewall router, the authentication proxy requires a per-user access profile configuration on the AAA server. To support the authentication proxy, configure the AAA authorization service “auth-proxy” on the AAA server as outlined here:

- Define a separate section of authorization for **auth-proxy** to specify the downloadable user profiles. This does not interfere with other type of services, such as EXEC. The following example shows a user profile on a TACACS server:

```
default authorization = permit
key = cisco
user = newuser1 {
login = cleartext cisco
service = auth-proxy
{
priv-lvl=15
proxyacl#1="permit tcp any any eq 26"
proxyacl#2="permit icmp any host 60.0.0.2"
proxyacl#3="permit tcp any any eq ftp"
proxyacl#4="permit tcp any any eq ftp-data"
proxyacl#5="permit tcp any any eq smtp"
proxyacl#6="permit tcp any any eq telnet"
}
}
```

- The only supported attribute in the AAA server user configuration is **proxyacl#n**. Use the **proxyacl#n** attribute when configuring the access lists in the profile. The attribute **proxyacl#n** is for both RADIUS and TACACS+ attribute-value (AV) pairs.
- The privilege level must be set to 15 for all users.
- The access lists in the user profile on the AAA server must have **permit** only access commands.
- Set the source address to **any** in the each of the user profile access list entries. The source address in the access lists is replaced with the source address of the host making the authentication proxy request when the user profile is downloaded to the firewall.
- The supported AAA servers are CiscoSecure ACS 2.3 for Windows NT, CiscoSecure ACS 2.3 for UNIX, TACACS+ server (vF4.02.alpha), Ascend RADIUS server - radius-980618 (required apair patch), and Livingston RADIUS server (v1.16).

Refer to the “AAA Server User Profile” section on page 27 for sample AAA server configurations.

Configuring the HTTP Server

To use the authentication proxy, you must also enable the HTTP server on the firewall and set the HTTP server authentication method to use AAA. Enter the following commands in global configuration mode:

Step	Command	Purpose
1	<code>router(config)# ip http server</code>	Enable the HTTP server on the router. The authentication proxy uses the HTTP server to communicate with the client for user authentication.
2	<code>router(config)# ip http authentication aaa</code>	Set the HTTP server authentication method to AAA.
3	<code>router(config)# ip http access-class access-list-number</code>	Specify the access list for the HTTP server. Use the standard access list number configured in the “Interface Configuration” section on page 18.

Configuring the Authentication Proxy

Note Set the **auth-cache-time** option for any authentication proxy rule to a higher value than the idle timeout value for any CBAC inspection rule. When the authentication proxy removes an authentication cache along with its associated dynamic user ACL, there might be some idle connections monitored by CBAC, and removal of user-specific ACLs could cause those idle connections to hang. If CBAC has a shorter idle time out, CBAC resets these connections when the idle timeout expires; that is, before the authentication proxy removes the user profile.

To configure the authentication proxy, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<code>router(config)# ip auth-proxy auth-cache-time min</code>	Set the global authentication proxy idle timeout value in minutes. If the timeout expires, user authentication entries are removed, along with any associated dynamic access lists. The default value is 60 minutes.
2	<code>router(config)# ip auth-proxy auth-proxy-banner</code>	(Optional) Display the name of the firewall router in the authentication proxy login page. The banner is disabled by default.
3	<code>router(config)# ip auth-proxy name auth-proxy-name http [auth-cache-time min] [list std-access-list</code>	<p>Create authentication proxy rules. The rules define how you apply authentication proxy. This command associates connection initiating HTTP protocol traffic with an authentication proxy name. You can associate the named rule with an access control list, providing control over which hosts use the authentication proxy feature. If no standard access list is defined, the named authentication proxy rule intercepts HTTP traffic from all hosts whose connection initiating packets are received at the configured interface.</p> <p>(Optional) The auth-cache-time option overrides the global authentication proxy cache timer. This option provides more control over timeout values for a specific authentication proxy rule. If no value is specified, the proxy rule assumes the value set with the ip auth-proxy auth-cache-time command.</p> <p>(Optional) The list option allows you to apply a standard access list to a named authentication proxy rule. HTTP connections initiated from hosts in the access list are intercepted by the authentication proxy.</p>
4	<code>router(config)# interface type</code>	Enter interface configuration mode by specifying the interface type on which to apply the authentication proxy.
5	<code>router(config-if)# ip auth-proxy auth-proxy-name</code>	In interface configuration mode, apply the named authentication proxy rule at the interface. This command enables the authentication proxy rule with that name.

Verifying the Authentication Proxy

Verifying the authentication proxy configuration can have several components:

- Checking the Authentication Proxy Configuration (Optional)
- Establishing User Connections with JavaScript Enabled (Optional)
- Establishing User Connections with JavaScript Disabled (Optional)

Checking the Authentication Proxy Configuration

To check the current authentication proxy configuration, use the **show ip auth-proxy configuration** command in privileged EXEC mode:

Command	Purpose
<code>router# show ip auth-proxy configuration</code>	Display the authentication proxy configuration.

In the following example, the global authentication proxy idle timeout value is set to 60 minutes, the named authentication proxy rule is “pxy,” and the idle timeout value for this named rule is 1 minute. The display shows that no host list is specified, meaning that all connections initiating HTTP traffic at the interface are subject to the authentication proxy rule:

```
router# sh ip auth-proxy configuration
Authentication cache time is 60 minutes
Authentication Proxy Rule Configuration
Auth-proxy name pxy
http list not specified auth-cache-time 1 minutes
```

To verify that the authentication proxy is successfully configured on the router, ask a user to initiate an HTTP connection through the router. The user must have authentication and authorization configured at the AAA server. If the user authentication is successful, the firewall completes the HTTP connection for the user. If the authentication is unsuccessful, check the access list and the AAA server configurations.

Display the user authentication entries using the **show ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
<code>router# show ip auth-proxy cache</code>	Display the list of user authentication entries.

The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is HTTP_ESTAB, the user authentication was successful.

```
router# show ip auth-proxy cache
Authentication Proxy Cache
Client IP 192.168.25.215 Port 57882, timeout 1, state HTTP_ESTAB
```

Wait for one minute, which is the timeout value for this named rule, and ask the user to try the connection again. After one minute, the user connection is denied because the authentication proxy has removed the user’s authentication entry and any associated dynamic ACLs. The user is presented with a new authentication login page and must log in again to gain access through the firewall.

Establishing User Connections with JavaScript Enabled

To verify client connections using the authentication proxy with JavaScript enabled on the client browser, follow this procedure:

- Step 1** From a client host, initiate an HTTP connection through the firewall. This generates the authentication proxy login page.
- Step 2** At the authentication proxy login page, enter a username and password.
- Step 3** Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the authentication is successful, the connection is completed automatically. If the authentication fails, the authentication proxy reports the failure to the user, and prompts the user with multiple retries.

Note If the authentication attempt is unsuccessful after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

Establishing User Connections with JavaScript Disabled

The Authentication proxy design requires JavaScript to ensure secure authentication. You can use the authentication proxy without enabling JavaScript on the browser, but this poses a potential security risk if users do not properly establish network connections. The following procedure provides the steps to properly establish a connection with JavaScript disabled. Network administrators are strongly advised to instruct users on how to properly establish connections using the procedure in this section.

Note Failure to follow this procedure can cause user credentials to be passed to a network web server other than the authentication proxy or can cause the authentication proxy to reject the login attempt.

To verify client connections using the authentication proxy with JavaScript disabled on the client browser, follow this procedure:

- Step 1** Initiate an HTTP connection through the firewall.
This generates the authentication proxy login page.
- Step 2** From the authentication proxy login page at the client, enter the username and password.
- Step 3** Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the popup window indicates successful authentication, go to Step 7.

- Step 4** If the popup window displays a failed authentication message, click **Close** on the browser **File** menu.

Note Do not click **Reload (Refresh** for Internet Explorer) to close the popup window.

Step 5 From the original authentication login page, click **Reload (Refresh** for Internet Explorer) on the browser toolbar. The user login credentials are cleared from the form.

Note Do not click **OK**. You must click **Reload** or **Refresh** to clear the username and password and to reload the form before attempting to log in again.

Step 6 Enter the username and password again.

If the authentication is successful, a window appears displaying a successful authentication message. If the window displays a failed authentication message, go to Step 4.

Step 7 Click **Close** on the browser **File** menu.

Step 8 From the original authentication proxy login page, click **Reload (Refresh** for Internet Explorer) on the browser toolbar.

The authentication proxy completes the authenticated connection with the web server.

Monitoring and Maintaining the Authentication Proxy

This section describes how to view dynamic access list entries and how to manually remove authentication entries.

Displaying Dynamic ACL Entries

You can display dynamic access list entries when they are in use. After an authentication proxy entry is cleared by you or by the idle timeout parameter, you can no longer display it. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established by the authentication proxy, use the **show ip access-lists** command in privileged EXEC mode:

Command	Purpose
router# show ip access-lists	Display the standard and extended access lists configured on the firewall, including dynamic ACL entries.

Consider the following example where ACL 105 is applied inbound at the input interface where you configure authentication proxy. The initial display shows the contents of the ACLs prior to authentication. The second display shows the same displays after user authentication with the AAA server.

Note If NAT is configured, the **show ip access list** command might display the translated host IP address for the dynamic ACL entry. This depends on whether the ACL is applied on the interface where NAT is applied **inside** or **outside**. If the ACL is applied on the NAT outside interface, the translated address is displayed. If the ACL is applied on the NAT inside interface, the IP address of the host initiating the connection is displayed. The **show ip auth-proxy cache** command always displays the IP address of the host initiating the connection.

Initial ACL entries prior to the authentication proxy:

```
Router# show ip access-lists
.
.
.
Extended IP access list 105
  deny tcp any any eq telnet
  deny udp any any
  permit tcp any any (28 matches)
  permit ip any any
```

The ACL entries following user authentication are shown in boldface type:

```
Router# show ip access-lists
.
.
.
Extended IP access list 105
  permit tcp host 192.168.25.215 any eq 26
  permit icmp host 192.168.25.215 host 60.0.0.2
  permit tcp host 192.168.25.215 any eq telnet
  permit tcp host 192.168.25.215 any eq ftp
  permit tcp host 192.168.25.215 any eq ftp-data
  permit tcp host 192.168.25.215 any eq smtp
  deny tcp any any eq telnet
  deny udp any any
  permit tcp any any (76 matches)
  permit ip any any
```

Deleting Authentication Proxy Cache Entries

When the authentication proxy is in use, dynamic access lists dynamically grow and shrink as authentication entries are added and deleted. To display the list of authentication entries, use the **show ip auth-proxy cache** command. To manually delete an authentication entry, use the **clear ip auth-proxy cache** command in privileged EXEC:

Command	Purpose
router# clear ip auth-proxy cache { * host ip address }	Delete authentication proxy entries from the firewall before they time out. Use an asterisk to delete all authentication cache entries. Enter a specific IP address to delete an entry for a single host.

Configuration Examples

Configuring the authentication proxy feature requires configuration changes on both the router and the AAA server. This section provides the following examples:

- Authentication Proxy Configuration
- Authentication Proxy, IPSec, and CBAC Configuration
- Authentication Proxy, IPSec, NAT, and CBAC Configuration
- AAA Server User Profile

Throughout these examples, the “!” symbol indicates a comment line. Comment lines precede the configuration entries being described.

Authentication Proxy Configuration

The following examples highlight the specific authentication proxy configuration entries. These examples do not represent a complete router configuration. Complete router configurations using the authentication proxy are included later in this document.

AAA Configuration

```
aaa new-model
aaa authentication login default tacacs+ radius
!Set up the aaa new model to use the authentication proxy.
aaa authorization auth-proxy default tacacs+ radius
!Define the AAA servers used by the router
tacacs-server host 172.31.54.143
tacacs-server key cisco
radius-server host 172.31.54.143
radius-server key cisco
```

HTTP Server Configuration

```
! Enable the HTTP server on the router:
ip http server
! Set the HTTP server authentication method to AAA:
ip http authentication aaa
!Define standard access list 61 to deny any host.
access-list 61 deny any
! Use ACL 61 to deny connections from any host to the HTTP server.
ip http access-class 61
```

Authentication Proxy Configuration

```
!set the global authentication proxy timeout value.
ip auth-proxy auth-cache-time 60
!Apply a name to the authentication proxy configuration rule.
ip auth-proxy name HQ_users http
```

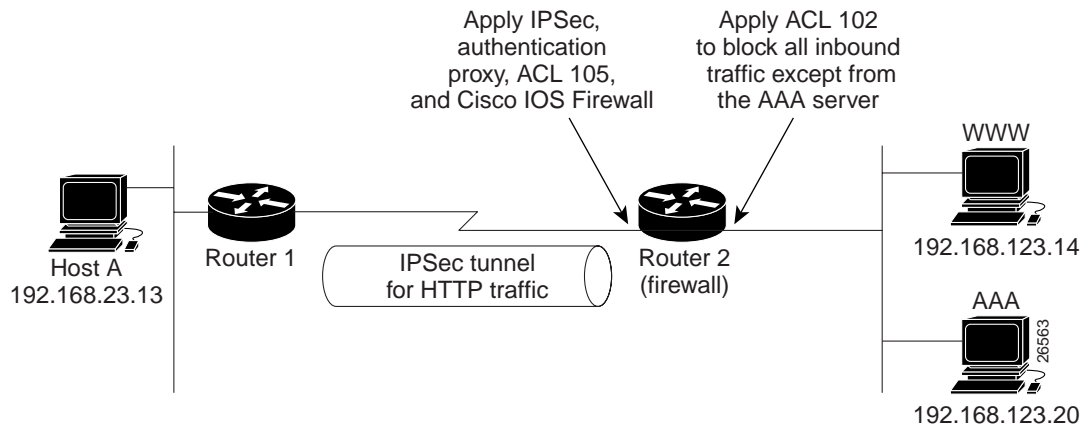
Interface Configuration

```
! Apply the authentication proxy rule at an interface.
interface e0
 ip address 10.1.1.210 255.255.255.0
 ip auth-proxy HQ_users
```

Authentication Proxy, IPSec, and CBAC Configuration

This example shows a router configuration with the authentication proxy, IPSec, and CBAC features. Figure 6 illustrates the configuration.

Figure 6 Authentication Proxy, IPSec, and CBAC Configuration Example



In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between Router 1 and Router 2 is encrypted using IPSec. The authentication proxy, IPSec, and CBAC are configured at interface Serial0 on Router 2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial0. ACL 102 is applied at interface Ethernet0 on Router 2 to block traffic on that interface, except traffic from the AAA server.

When Host A initiates an HTTP connection with the web server, the authentication proxy prompts the user at Host A for a username and password. These credentials are verified with the AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness.

Router 1 Configuration

```
! Configure Router 1 for IPSec.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
logging buffered 4096 debugging
no logging console
enable secret 5 $1$E00B$AQF1vFZM3fLr3LQA0sudL/
enable password junk
!
username Router2 password 0 welcome
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.2
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
!
crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.2
 set transform-set rule_1
 match address 155
!
interface Ethernet0/0
 ip address 192.168.23.2 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation PPP
 ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
 clockrate 56000
 crypto map testtag
!
!
ip classless
ip route 192.168.123.0 255.255.255.0 10.0.0.2
!Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.23.13 host 192.168.123.14 eq www
access-list 155 permit tcp host 192.168.23.13 eq www host 192.168.123.14
```

Router 2 Configuration

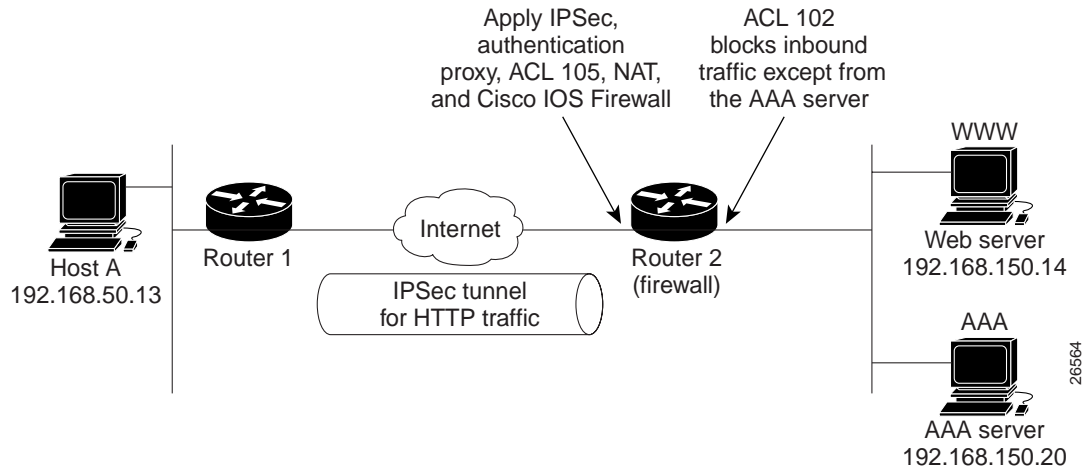
```
!Configure Router 2 as the firewall, using the authentication proxy, IPSec, and CBAC.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router2
!
logging buffered 4096 debugging
aaa new-model
aaa authentication login default group tacacs+
aaa authentication login console_line none
aaa authentication login special none
aaa authentication ppp default group tacacs+
aaa authorization exec default group tacacs+
!Configure AAA for the authentication proxy
aaa authorization auth-proxy default group tacacs+
enable password junk
!
!Create the CBAC inspection rule HTTP_TEST.
ip inspect name rule22 http
ip inspect name rule22 tcp
ip inspect name rule22 ftp
ip inspect name rule22 smtp
!
!Create the authentication proxy rule PXY.
ip auth-proxy name pxy http
!Turn on display of the router name in the authentication proxy login page.
ip auth-proxy auth-proxy-banner
ip audit notify log
ip audit po max-events 100
!
!Configure IPSec.
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.1
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set rule_1
 match address 155
!
!Apply the CBAC inspection rule and the authentication proxy rule at interface
!Serial0/0.
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip inspect rule22 in
 ip auth-proxy pxy
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
 crypto map testtag
!
interface Ethernet0/1
 ip address 192.168.123.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
```

```
ip route-cache
no ip mroute-cache
!
no ip classless
ip route 192.168.23.0 255.255.255.0 10.0.0.1
ip route 192.168.50.0 255.255.255.0 16.0.0.1
!Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
!Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
!Create ACL 102 to block all traffic inbound on interface Ethernet0/1 except for
!traffic from the AAA server.
access-list 102 permit tcp host 192.168.123.20 eq tacacs host 192.168.123.2
access-list 102 deny tcp any any
access-list 102 deny udp any any
access-list 102 permit ip any any
!Create ACL 105 to block all traffic inbound on interface Serial0/0. Permit only IP
!protocol traffic.
access-list 105 deny tcp any any
access-list 105 deny udp any any
access-list 105 permit ip any any
!Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.123.14 host 192.168.23.13 eq www
access-list 155 permit tcp host 192.168.123.14 eq www host 192.168.23.13
!
!Define the AAA server host and encryption key.
tacacs-server host 192.168.123.14
tacacs-server key cisco
!
line con 0
exec-timeout 0 0
login authentication special
transport input none
line aux 0
transport input all
speed 38400
flowcontrol hardware
line vty 0 4
password lab
```

Authentication Proxy, IPSec, NAT, and CBAC Configuration

This example provides a router configuration with the authentication proxy, IPSec, NAT, and CBAC features. Figure 7 illustrates the configuration.

Figure 7 Authentication Proxy, IPSec, and CBAC Configuration Example



In this example, Host A initiates an HTTP connection with web server (WWW). The HTTP traffic between Router 1 (interface BRI0) and Router 2 (interface Serial2) is encrypted using IPSec. The authentication proxy is configured on Router 2, which is acting as the firewall. The authentication proxy, NAT, and CBAC are configured at interface Serial2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial2. ACL 102 is applied at interface Ethernet0 on Router 2 to block traffic on that interface, except traffic from the AAA server. In this example, the authentication proxy use standard ACL 10 to specify the hosts using the authentication proxy feature.

When any host in ACL 10 initiates an HTTP connection with the web server, the authentication proxy prompts the user at that host for a username and password. These credentials are verified with AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness.

Router 1 Configuration

```
! Configure Router 1 for IPSec.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
logging buffered 4096 debugging
no logging console
!
isdn switch-type basic-5ess
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 16.0.0.2
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
!
crypto map testtag 10 ipsec-isakmp
 set peer 16.0.0.2
 set transform-set rule_1
 match address 155
!
!
process-max-time 200
!
interface BRI0
 ip address 16.0.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer idle-timeout 5000
 dialer map ip 16.0.0.2 name router2 broadcast 50006
 dialer-group 1
 isdn switch-type basic-5ess
 crypto map testtag
!
interface FastEthernet0
 ip address 192.168.50.2 255.255.255.0
 no ip directed-broadcast
!
ip classless
ip route 192.168.150.0 255.255.255.0 16.0.0.2
no ip http server
!Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.50.13 host 192.168.150.100 eq www
access-list 155 permit tcp host 192.168.50.13 eq www host 192.168.150.100
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password lab
 login
!
```

Router 2 Configuration

```

!Configure Router 2 as the firewall, using the authentication proxy, IPSec, NAT and
!CBAC.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname router2
!
logging buffered 4096 debugging
aaa new-model
aaa authentication login default group tacacs+
aaa authentication login console_line none
aaa authorization exec default group tacacs+
!Configure AAA for the authentication proxy
aaa authorization auth-proxy default group tacacs+
!
!Create the CBAC inspection rule "rule44".
ip inspect name rule44 http java-list 5
ip inspect name rule44 tcp
ip inspect name rule44 ftp
ip inspect name rule44 smtp
!
!Create the authentication proxy rule "pxy.". Set the timeout value for rule
!pxy to three minutes. Standard ACL 10 is applied to the rule.
ip auth-proxy name pxy http list 10 auth-cache-time 3
isdn switch-type primary-5ess
!
!Configure IPSec.
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 16.0.0.1
!
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
!
crypto map testtag 10 ipsec-isakmp
 set peer 16.0.0.1
 set transform-set rule_1
 match address 155
!
controller T1 2/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
!Apply ACL 102 inbound at interface Ethernet0/1 and configure NAT.
interface Ethernet0/1
 ip address 192.168.150.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip nat inside
 no ip mroute-cache
!
!Apply the authentication proxy rule PXY, CBAC inspection rule HTTP_TEST, NAT, and
!and ACL 105 at interface Serial2/0:23.
interface Serial2/0:23
 ip address 16.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip nat outside
 ip inspect rule44 in

```

```
ip auth-proxy pxy
encapsulation ppp
ip mroute-cache
dialer idle-timeout 5000
dialer map ip 16.0.0.1 name router1 broadcast 71011
dialer-group 1
isdn switch-type primary-5ess
fair-queue 64 256 0
crypto map testtag
!
!Use NAT to translate the Web server address.
ip nat inside source static 192.168.150.14 192.168.150.100
ip classless
ip route 192.168.50.0 255.255.255.0 16.0.0.1
!Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
!Create standard ACL 5 to specify the list of hosts from which to accept java applets.
!ACL 5 is used to block Java applets in the CBAC inspection rule named "rule44," which
!is applied at interface Serial2/0:23.
access-list 5 permit any
!Create standard ACL 10 to specify the hosts using the authentication proxy. This ACL
!used in the authentication proxy rule named "PXY", which is applied at interface
!Serial2/0:23.
access-list 10 permit any
!Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
!Create extended ACL 102 to block all traffic inbound on interface Ethernet0/1
!except for traffic from the AAA server.
access-list 102 permit tcp host 192.168.150.20 eq tacacs 192.168.150.2
access-list 102 deny tcp any any
access-list 102 deny udp any any
access-list 102 permit ip any any
!Create extended ACL 105 to block all TCP and UDP traffic inbound on interface
!Serial2/0:23.
access-list 105 deny tcp any any
access-list 105 deny udp any any
access-list 105 permit ip any any
!Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.150.100 host 192.168.50.13 eq www
access-list 155 permit tcp host 192.168.150.100 eq www host 192.168.50.13
dialer-list 1 protocol ip permit
!Define the AAA server host and encryption key.
tacacs-server host 192.168.126.14
tacacs-server key cisco
!
line con 0
exec-timeout 0 0
!Define the AAA server host and encryption key.
login authentication console_line
transport input none
line aux 0
line vty 0 4
password lab
!
!
end
```

AAA Server User Profile

This section includes examples of the authentication proxy user profiles entries on the AAA servers.

The **proxyacl** entries define the user's access privileges. After successfully using the authentication proxy to log in, these entries are transferred to the firewall router. Each entry in the profile must specify **permit** access for the service or application. The source address in each entry is set to "any," which is replaced with the IP address of the authenticating host when the profile is downloaded to the firewall. The privilege level must be set to 15 for all AAA users.

CiscoSecure ACS 2.3 for Windows NT

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for Windows NT. For detailed information about CiscoSecure ACS, refer to the documentation for that product.

The following sample configuration is for the TACACS+ service of CiscoSecure ACS for Windows NT.

- Step 1** Click the Interface Configuration icon and click **TACACS+ (Cisco)**.
- (a) Scroll down to New Services.
 - (b) Add a new service, "auth-proxy" in the Service field. Leave the Protocol field empty.
 - (c) Select both the User and Group check boxes for the new service.
 - (d) Scroll down to Advance Configuration Options and check the Per-user Advance TACACS+ features.
 - (e) Click **Submit**.
- Step 2** Click the Network Configuration icon.
- (a) Click the Add Entry icon for Network Access Servers and fill in the Network Access Server Hostname, IP address, and the key (the key configured on the router) fields.
 - (b) Select TACACS+ (Cisco) for the Authenticate Using option.
 - (c) Click the Submit + Restart icon.
- Step 3** Click the Group Setup icon.
- (a) Select a user group from the drop-down menu.
 - (b) Select the Users in Group check box.
 - (c) Select a user from the user list.
 - (d) In the User Setup list, scroll down to TACACS+ Settings and select the "auth-proxy" check box.
 - (e) Select the Custom Attributes check box.
 - (f) Add the profiles entries (do not use single or double quotes around the entries) and set the privilege level to 15.

```
priv-lvl=15
proxyacl#1=permit tcp any any eq 26
proxyacl#2=permit icmp any host 60.0.0.2
proxyacl#3=permit tcp any any eq ftp
proxyacl#4=permit tcp any any eq ftp-data
proxyacl#5=permit tcp any any eq smtp
proxyacl#6=permit tcp any any eq telnet
```

(g) Click **Submit**.

Step 4 Click the User Setup icon.

(a) Click **List All Users**.

(b) Add a username.

(c) Scroll down to User Setup Password Authentication.

(d) Select SDI SecurID Token Card from the Password Authentication drop-down menu.

(e) Select the previous configured user group 1.

(f) Click **Submit**.

Step 5 Click Group Setup icon again.

(a) Select the user group 1.

(b) Click **Users in Group**.

(c) Click **Edit Settings**.

(d) Click the Submit + Restart icon to make sure the latest configuration is updated and sent to the AAA server.

CiscoSecure ACS 2.3 for UNIX

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for UNIX. For detailed information regarding CiscoSecure ACS, refer to the documentation for that product.

To manage the CiscoSecure ACS using the Administrator program, you need a web browser that supports Java and JavaScript. You must enable Java in the browser application. You can start the Java-based CiscoSecure Administrator advanced configuration program from any of the CiscoSecure ACS Administrator web pages.

The following sample configuration procedure is for the TACACS+ service of CiscoSecure ACS 2.3 for UNIX.

Step 1 On the CiscoSecure ACS web menu bar of the CiscoSecure ACS web interface, click **Advanced** and then click **Advanced** again.

The Java-based CiscoSecure Administrator advanced configuration program appears. It might require a few minutes to load.

Step 2 In the CiscoSecure Administrator advanced configuration program, locate and deselect Browse in the Navigator pane of the tabbed Members page.

This displays the Create New Profile icon.

Step 3 In the Navigator pane, do one of the following:

- Locate and click the group to which the user will belong
- If you do not want the user to belong to a group, click the [Root] folder icon.

Step 4 Click **Create Profile** to display the New Profile dialog box.

Step 5 Make sure the Group check box is cleared.

Step 6 Enter the name of the user you want to create and click **OK**. The new user appears in the tree.

- Step 7** Click the icon for the group or user profile in the tree that is displayed in the Navigator pane of the tabbed Members page.
- Step 8** If necessary, in the Profile pane, click the Profile icon to expand it.
A list or dialog box that contains attributes applicable to the selected profile or service appears in the window at the bottom right of the screen. The information in this window changes depending on what you have selected in the Profile pane.
- Step 9** Click **Service-String**.
- Step 10** Click **string**, enter **auth-proxy** in the text field, and click **Apply**.
- Step 11** Select the **Option** menu.
- Step 12** On the **Option** menu, click **Default Attributes**.
- Step 13** Change the attribute from Deny to **Permit**.
- Step 14** Click **Apply**.
- Step 15** On the **Option** menu, click **Attribute** and enter the privilege level in the text field:
`priv-lvl=15`
- Step 16** On the **Option** menu, click **Attribute** and enter the **proxyacl** entries in the text field:
`proxyacl#1="permit tcp any any eq 26"`

Repeat this step for each additional service or protocol to add:

`proxyacl#2="permit icmp any host 60.0.0.2"`
`proxyacl#3="permit tcp any any eq ftp"`
`proxyacl#4="permit tcp any any eq ftp-data"`
`proxyacl#5="permit tcp any any eq smtp"`
`proxyacl#6="permit tcp any any eq telnet"`
- Step 17** When you have finished making all your changes, click Submit.

TACACS+ Server

```

default authorization = permit
key = cisco
user = Brian {
  login = cleartext cisco
  service = auth-proxy
  {
    priv-lvl=15
    proxyacl#1="permit tcp any any eq 26"
    proxyacl#2="permit icmp any host 60.0.0.2"
    proxyacl#3="permit tcp any any eq ftp"
    proxyacl#4="permit tcp any any eq ftp-data"
    proxyacl#5="permit tcp any any eq smtp"
    proxyacl#6="permit tcp any any eq telnet"
  }
}

```

Livingston Radius Server

```
Bob Password = "cisco" User-Service-Type=Outbound-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Ascend Radius Server

```
Alice Password = "cisco" User-Service = Dialout-Framed-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications.

- **aaa authorization**
- **clear ip auth-proxy cache**
- **ip auth-proxy**
- **ip auth-proxy auth-cache-time**
- **ip auth-proxy auth-proxy-banner**
- **ip auth-proxy name**
- **show ip auth-proxy**

In Cisco IOS Release 12.0(1)T or later, you can search and filter the output for **show** and **more** commands. This functionality is useful when you need to sort through large amounts of output, or if you want to exclude output that you do not need to see.

To use this functionality, enter a **show** or **more** command followed by the “pipe” character (`|`), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search or filter on:

```
command / {begin | include | exclude} regular-expression
```

Following is an example of the **show atm vc** command in which you want the command output to begin with the first line where the expression “PeakRate” appears:

```
show atm vc /begin PeakRate
```

For more information on the search and filter functionality, refer to the Cisco IOS Release 12.0(1)T feature module titled *CLI String Search*.

aaa authorization

To set parameters that restrict a user's network access, use the **aaa authorization** global configuration command. To disable authorization for a function, use the **no** form of this command.

```
aaa authorization {network | exec | commands level | reverse-access | auth-proxy} {default | list-name} [method1 [method2...]]
```

```
no aaa authorization {network | exec | commands level | reverse-access | auth-proxy}
```

Syntax Description

network	Runs authorization for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARA.
exec	Runs authorization to determine if the user is allowed to run an EXEC shell. This facility might return user profile information such as autocommand information.
commands	Runs authorization for all commands at the specified privilege level.
<i>level</i>	Specific command level that should be authorized. Valid entries are 0 through 15.
reverse-access	Runs authorization for reverse access connections, such as reverse Telnet.
auth-proxy	Runs authorization for the authentication proxy.
default	Uses the listed authorization methods that follow this argument as the default list of methods for authorization.
<i>list-name</i>	Character string used to name the list of authorization methods.
<i>method1</i> [<i>method2</i> ...]	One of the keywords listed in Table 3.

Defaults

Authorization is disabled for all actions (equivalent to the method keyword **none**). 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 (where this authorization type applies) 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 authorization takes place.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(5)T	The auth-proxy keyword was introduced.

Usage Guidelines

Use the **aaa authorization** command to enable authorization and to create named methods lists, defining authorization methods that can be used when a user accesses the specified function. Method lists for authorization define the ways 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 RADIUS or TACACS+), in sequence. Method lists enable you to designate 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.

Use the **aaa authorization** command to create a list by entering the *list-name* and the *method*, where *list-name* is any character string used to name this list (excluding all method names) and *method* identifies the list of authorization method(s) tried in the given sequence.

Method keywords are described in Table 3.

Table 3 AAA Authorization Methods

Method Keyword	Description
tacacs+	Requests authorization information from the TACACS+ server for the supported authorization type.
if-authenticated	Allows the user to access the requested function if the user is authenticated.
none	No authorization is performed.
local	Uses the local database for authorization.
radius	Uses RADIUS to get authorization information.
krb5-instance	Uses the instance defined by the kerberos instance map command.

Cisco IOS software supports the following six methods for 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 (AV) 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. Not supported with the **auth-proxy** authorization type.
- **None**—The network access server does not request authorization information; authorization is not performed over this line/interface. Not supported with the **auth-proxy** authorization type.
- **Local**—The router or access server consults its local database, as defined by the **username** command, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database. Not supported with the **auth-proxy** authorization type.
- **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.
- **Kerberos Instance Map**—The network access server uses the instance defined by the **kerberos instance map** command for authorization. Not supported with the **auth-proxy** authorization type.

Method lists are specific to the type of authorization being requested. AAA supports five different types of authorization:

- **Network**—Applies to network connections. This can include a PPP, SLIP, or ARA connection.
- **EXEC**—Applies to the attributes associated with a user EXEC terminal session.
- **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.
- **Reverse Access**—Applies to reverse Telnet sessions.
- **Auth-proxy**—Applies to HTTP sessions that trigger the authentication proxy feature.

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type.

Once defined, method lists must be applied to specific lines or interfaces before any of the defined methods will be performed.

The authorization command causes a request packet containing a series of AV pairs to be sent to the RADIUS or TACACS server as part of the authorization process. The server can do one of the following:

- Accept the request as is.
- Make changes to the request.
- Refuse the request and refuse authorization.

For a list of supported RADIUS attributes, refer to the RADIUS attributes appendix in the *Security Configuration Guide*. For a list of supported TACACS+ AV pairs, refer to the TACACS+ attribute-value pairs appendix in the Cisco IOS Release 12.0 *Security Configuration Guide*.

Note There are five commands associated with privilege level 0: **disable**, **enable**, **exit**, **help**, and **logout**. If you configure AAA authorization for a privilege level greater than 0, these five commands will not be included in the privilege level command set.

When you enable the authentication proxy, the AAA server, the database used for authentication, and the authentication proxy service must be configured for authorization.

Note Use the **ip auth-proxy name** command in conjunction with the **aaa authorization auth-proxy** command. Together these commands set up the authorization policy to be downloadable by the firewall.

Examples

In this example, the first method of authorization using the authentication proxy is TACACS+, and the secondary method is RADIUS.

```
aaa authorization auth-proxy default tacacs+ radius
```

In this example, the first method of authorization using the authentication proxy is RADIUS, and the secondary method is TACACS+.

```
aaa authorization auth-proxy default radius tacacs+
```

The following example shows the **aaa authorization auth-proxy** command as part of an AAA new model configuration. Use these AAA configuration commands to secure the router when the authentication proxy is enabled. Failure to configure the router properly could result in security holes.

```
aaa new-model  
aaa authentication login default radius tacacs+  
aaa authorization auth-proxy default radius tacacs+
```

Related Commands

Command	Description
ip auth-proxy name	Create an authentication proxy rule.

clear ip auth-proxy cache

To clear authentication proxy entries from the router, use the **clear ip auth-proxy cache** global configuration command.

```
clear ip auth-proxy cache {* | host ip address}
```

Syntax Description

<i>*</i>	Clears all authentication proxy entries, including user profiles and dynamic access lists.
<i>host ip address</i>	Clears the authentication proxy entry, including user profiles and dynamic access lists, for the specified host.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use this command to clear entries from the translation table before they time out.

Examples

The following command deletes all authentication proxy entries:

```
clear ip auth-proxy cache *
```

This command deletes the authentication proxy entry for the host with IP address 192.168.4.5:

```
clear ip auth-proxy cache 192.168.4.5
```

Related Commands

Command	Description
show ip auth-proxy cache	Display the running authentication proxy configuration.

ip auth-proxy

To apply an authentication proxy rule at a firewall interface, use the **ip auth-proxy** interface configuration command. To remove the authentication proxy rules, use the **no** form of this command.

ip auth-proxy *auth-proxy-name*
no ip auth-proxy *auth-proxy-name*

Syntax Description

auth-proxy-name Specifies the name of the authentication proxy rule to apply to the interface configuration. The authentication proxy rule is established with the **authentication proxy name** command.

Defaults

No default behavior or values.

Command Modes

Interface configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use the **ip auth-proxy** command to enable the named authentication proxy rule at the firewall interface. Traffic passing through the interface from hosts with an IP address matching the standard access list and protocol type (HTTP) is intercepted for authentication if no corresponding authentication cache entry exists. If no access list is defined, the authentication proxy intercepts traffic from all hosts whose connection initiating packets are received at the configured interface.

Use the **no** form of this command with a rule name to disable the authentication proxy for a given rule on a specific interface. If a rule is not specified, the no form of this command disables the authentication proxy on the interface.

Examples

The following example configures interface Ethernet0 with the HQ_users rule:

```
interface e0
 ip address 172.21.127.210 255.255.255.0
 ip access-group 111 in
 ip auth-proxy HQ_users
 ip nat inside
```

Related Commands

Command	Description
ip auth-proxy name	Create an authentication proxy rule.

ip auth-proxy auth-cache-time

To set the authentication proxy idle timeout value (the length of time an authentication cache entry, along with its associated dynamic user ACL, is managed after a period of inactivity), use the **ip auth-proxy auth-cache-time** global configuration command. To set the default value, use the **no** form of this command.

ip auth-proxy auth-cache-time *min*

no ip auth-proxy auth-cache-time

Syntax Description

min Specifies the length of time in minutes that an authentication cache entry, along with its associated dynamic user ACL, is managed after a period of inactivity. Enter a value in the range 1 to 2,147,483,647. The default value is 60 minutes.

Defaults

60 minutes.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use this command to set the global idle timeout value for the authentication proxy. You must set the **auth-cache-time** timeout value to a higher value than the idle timeout of any CBAC protocols. Otherwise, when the authentication proxy removes the user profile along associated dynamic user ACLs, there might be some idle connections monitored by CBAC. Removing these user-specific ACLs could cause those idle connections to hang. If the CBAC idle time out value is shorter, CBAC resets these connections when the CBAC idle timeout expires, which is before the authentication proxy removes the user profile.

Examples

The following example sets the authorization cache timeout to 30 minutes:

```
ip auth-proxy auth-cache-time 30
```

Related Commands

Command	Description
ip auth-proxy name	Create an authentication proxy rule.
show ip auth-proxy configuration	Display the running authentication proxy configuration.

ip auth-proxy auth-proxy-banner

To display the router name in the authentication proxy login page, use the **ip auth-proxy auth-proxy-banner** configuration command. To disable display of the router name, use the **no** form of this command.

ip auth-proxy auth-proxy-banner

no ip auth-proxy auth-proxy-banner

Defaults

Disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use the **ip auth-proxy auth-proxy-banner** command to display the banner in the authentication proxy login page.

Examples

The following example sets the authorization cache timeout to the default value:

```
ip auth-proxy auth-proxy-banner
```

Related Commands

Command	Description
ip auth-proxy name	Create an authentication proxy rule.

ip auth-proxy name

To create an authentication proxy rule, use the **ip auth-proxy name** global configuration command. To remove the authentication proxy rules, use the **no** form of this command.

```
ip auth-proxy name auth-proxy-name http [auth-cache-time min] [list std-access-list]  
no ip auth-proxy name auth-proxy-name
```

Syntax Description

<i>auth-proxy-name</i>	Associates a name with an authentication proxy rule. Enter a name of up to 16 alphanumeric characters.
http	Specifies the protocol that triggers the authentication proxy. The only supported protocol is HTTP.
auth-cache-time <i>min</i>	(Optional) Overrides the global authentication proxy cache timer for a specific the authentication proxy name, offering more control over timeout values. Enter a value in the range 1 to 2,147,483,647. The default value is equal to the value set with the ip auth-proxy auth-cache-time command.
list <i>std-access-list</i>	(Optional) Specifies a standard access list to use with the authentication proxy. With this option, the authentication proxy is applied only to those hosts in the standard access list. If no list is specified, all connections initiating HTTP traffic arriving at the interface are subject to authentication.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

This command creates a named authentication proxy rule, and it allows you to associate that rule with an access control list, providing control over which hosts use the authentication proxy. The rule is applied to an interface on a router using the **ip auth-proxy** command.

Use the **auth-cache-time** option to override the global the authentication proxy cache timer. This option provides control over timeout values for specific authentication proxy rules. The authentication proxy cache timer monitors the length of time (in minutes) that an authentication

cache entry, along with its associated dynamic user ACL, is managed after a period of inactivity. When that period of inactivity (idle time) expires, the authentication entry and the associated dynamic access lists are deleted.

Use the **list** option to associate a set of specific IP addresses with the **auth-proxy** name.

Use the **no** form of this command with a rule name to remove the authentication proxy rules. If no rule is specified, the no form of this command removes all the authentication rules on the router, and disables the proxy at all interfaces.

Note You must use the **aaa authorization auth-proxy** command together with the **ip auth-proxy name** command. Together these commands set up the authorization policy to be retrieved by the firewall. Refer to the **aaa authorization auth-proxy** command for more information.

Examples

The following example creates the HQ_users authentication proxy rule. Because no standard access list is specified in the rule, all connection initiating HTTP traffic is subjected to authentication.

```
ip auth-proxy name HQ_users http
```

This command creates the Mfg_users authentication proxy rule and applies it to hosts specified in ACL 10:

```
access-list 10 192.168.7.0 0.0.0.255  
ip auth-proxy name Mfg_users http list 10
```

This command sets the timeout value for Mfg_users to 30 minutes:

```
access-list 15 any  
ip auth-proxy name Mfg_users http auth-cache-time 30 list 15
```

The following command disables the Mfg_users rule:

```
no ip auth-proxy name Mfg_users
```

This command disables the authentication proxy at all interfaces and removes all the rules from the router configuration:

```
no ip auth-proxy
```

Related Commands

Command	Description
aaa authorization	Set parameters that restrict a user's network access.
ip auth-proxy	Apply an authentication proxy rule at a firewall interface.
ip auth-proxy auth-cache-time	Set the authentication proxy idle timeout value.
ip auth-proxy name	Create an authentication proxy rule.
show ip auth-proxy configuration	Display the running authentication proxy configuration.

show ip auth-proxy

To display the authentication proxy entries or the running authentication proxy configuration, use the **ip auth-proxy** privileged EXEC command.

```
show ip auth-proxy {cache | configuration}
```

Syntax Description

cache	Display the current list of the authentication proxy entries.
configuration	Display the running authentication proxy configuration.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use the **show ip auth-proxy** to display either the authentication proxy entries or the running authentication proxy configuration. Use the **cache** keyword option to list the host IP address, the source port number, the timeout value for the authentication proxy, and the state for connections using authentication proxy. If authentication proxy state is HTTP_ESTAB, the user authentication was successful.

Use the **show ip auth-proxy configuration** command to display all authentication proxy rules configured on the router.

Examples

The following is sample output from the **show ip auth-proxy cache** command after one user authentication using the authentication proxy.

```
router# show ip auth-proxy cache
Authentication Proxy Cache
  Client IP 192.168.25.215 Port 57882, timeout 1, state HTTP_ESTAB
```

In the following example, the **show ip auth-proxy configuration** command displays the information about the authentication proxy rule **pxy**. The global idle timeout value is 60 minutes. The idle timeout value for this named rule is 30 minutes. No host list is specified in the rule, meaning that all connection initiating HTTP traffic at the interface is subject to the authentication proxy rule:

```
router# sh ip auth-proxy configuration
Authentication cache time is 60 minutes
Authentication Proxy Rule Configuration
Auth-proxy name pxy
http list not specified auth-cache-time 30 minutes
```

Related Commands

Command	Description
clear ip auth-proxy cache	Clear authentication proxy entries from the router.
ip auth-proxy	Apply an authentication proxy rule at a firewall interface.
ip auth-proxy auth-cache-time	Set the authentication proxy idle timeout value.
ip auth-proxy name	Create an authentication proxy rule.

Debug Commands

This section documents the new **debug** command related to the authentication-proxy feature.

- **debug ip auth-proxy**

debug ip auth-proxy

To display the authentication proxy configuration information on the router, use the **show ip auth-proxy configuration** command in privileged EXEC mode.

```
debug ip auth-proxy {ftp | function-trace | http | object-creation | object-deletion | tcp | telnet | timer}
```

Syntax Description

ftp	Display FTP events related to the authentication proxy.
function-trace	Display the authentication proxy functions.
http	Display HTTP events related to the authentication proxy.
object-creation	Display additional entries to the authentication proxy cache.
object-deletion	Display deletion of cache entries for the authentication proxy.
tcp	Display TCP events related to the authentication proxy.
telnet	Display Telnet related authentication proxy events.
timer	Displays authentication proxy timer-related events.

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use the **debug ip auth-proxy** command to display authentication proxy activity. Refer to the Examples section for more information about the debug options.

Note The **function-trace** debugging information provides low-level software information for Cisco technical support representatives. No output examples are provided for this keyword option.

Examples

The following examples illustrates the output of the **debug ip auth-proxy** command. In these examples, debugging is on for object creations, object deletions, HTTP, and TCP.

In this example, the client host at 192.168.201.1 is attempting to make an HTTP connection to the web server located at 192.168.21.1. The HTTP debugging information is on for the authentication proxy. The output shows that the router is setting up an authentication proxy entry for the login request:

```
00:11:10: AUTH-PROXY creates info:
      cliaddr - 192.168.21.1, cliport - 36583
      seraddr - 192.168.201.1, serport - 80
      ip-srcaddr 192.168.21.1
      pak-srcaddr 0.0.0.0
```

Following a successful login attempt, the debugging information shows the authentication proxy entries created for the client. In this example, the client is authorized for SMTP (port 25), FTP data (port 20), FTP control (port 21), and Telnet (port 23) traffic. The dynamic ACL entries are included in the display:

```
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 61AD60CC

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 6151C7C8 -- acl item 61AD60CC
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [25]
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 6151C908

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 6187A060 -- acl item 6151C908
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [20]
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 61A40B88

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 6187A0D4 -- acl item 61A40B88
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [21]
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 61879550

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 61879644 -- acl item 61879550
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [23]
```

The next example shows the debug output following a **clear ip auth-proxy cache** command to clear the authentication entries from the router. The dynamic ACL entries are removed from the router:

```
00:12:36:AUTH-PROXY OBJ_DELETE:delete auth_proxy cache 61AD6298
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 6151C7C8 -- acl item 61AD60CC
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 6187A060 -- acl item 6151C908
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 6187A0D4 -- acl item 61A40B88
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 61879644 -- acl item 61879550
```

The following example shows the timer information for a dynamic ACL entry. All times are expressed in milliseconds. The *first laststart* is the time that the ACL entry is created relative to the start up time of the router. The *lastref* is the time of the last packet to hit the dynamic ACL relative to the start up time of the router. The *exptime* is the next expected expiration time for the dynamic ACL. The *delta* indicates the remaining time before the dynamic ACL expires. After the timer expires, the debugging information includes a message indicating that the ACL and associated authentication proxy information for the client have been removed.

```
00:19:51:first laststart 1191112

00:20:51:AUTH-PROXY:delta 54220 lastref 1245332 exptime 1251112
00:21:45:AUTH-PROXY:ACL and cache are removed
```

Related Commands

Command	Description
show debug	Display the debug options set on the router.

