AAA Rules for Network Access

This chapter describes how to enable AAA (pronounced “triple A”) for network access.

AAA Performance

The ASA uses “cut-through proxy” to significantly improve performance compared to a traditional proxy server. The performance of a traditional proxy server suffers because it analyzes every packet at the application layer of the OSI model. The ASA cut-through proxy challenges a user initially at the application layer and then authenticates with standard AAA servers or the local database. After the ASA authenticates the user, it shifts the session flow, and all traffic flows directly and quickly between the source and destination while maintaining session state information.

Licensing Requirements for AAA Rules

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines

In clustering, this feature is only supported on the master unit.

Configuring Authentication for Network Access

This section includes the following topics:

Information About Authentication

The ASA lets you configure network access authentication using AAA servers. This section includes the following topics:

One-Time Authentication

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (**CLI**: See the **timeout uauth** command in the command reference for timeout values.) (**ASDM**: See the Configuration > Firewall > Advanced > Global Timeouts pane for timeout values.) For example, if you configure the ASA to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the authentication session exists, the user does not also have to authenticate for FTP.

Applications Required to Receive an Authentication Challenge

Although you can configure the ASA to require authentication for network access to any protocol or service, users can authenticate directly with HTTP, HTTPS, Telnet, or FTP only. A user must first authenticate with one of these services before the ASA allows other traffic requiring authentication.

The authentication ports that the ASA supports for AAA are fixed as follows:

- Port 21 for FTP
- Port 23 for Telnet
- Port 80 for HTTP
- Port 443 for HTTPS

ASA Authentication Prompts

For Telnet and FTP, the ASA generates an authentication prompt.

For HTTP, the ASA uses basic HTTP authentication by default, and provides an authentication prompt. You can optionally configure the ASA to redirect users to an internal web page where they can enter their username and password (**CLI**: configured with the **aaa authentication listener** command) (**ASDM**: configured in the

Configuration > Firewall > AAA Rules > Advanced > AAA Rules Advanced Options dialog box; see the (ASDM) Enabling the Redirection Method of Authentication for HTTP and HTTPS).

For HTTPS, the ASA generates a custom login screen. You can optionally configure the ASA to redirect users to an internal web page where they can enter their username and password (**CLI**: configured with the **aaa authentication listener** command) (**ASDM**: configured in the Configuration > Firewall > AAA Rules > Advanced > AAA Rules Advanced Options dialog box; see the (ASDM) Enabling the Redirection Method of Authentication for HTTP and HTTPS).

Redirection is an improvement over the basic method because it provides an improved user experience during authentication, and an identical user experience for HTTP and HTTPS in both Easy VPN and firewall modes. It also supports authentication directly with the ASA.

You might want to continue to use basic HTTP authentication for the following reasons:

- You do not want the ASA to open listening ports.
- You use NAT on a router and you do not want to create a translation rule for the web page served by the ASA.
- Basic HTTP authentication might work better with your network.

For example non-browser applications, as when a URL is embedded in e-mail, might be more compatible with basic authentication.

After you authenticate correctly, the ASA redirects you to your original destination. If the destination server also has its own authentication, the user enters another username and password. If you use basic HTTP authentication and need to enter another username and password for the destination server, then you need to configure virtual HTTP (**CLI**: the **virtual http** command) (**ASDM**: see the Configuration >Firewall > Advanced Options > Virtual Access pane).

Note If you use HTTP authentication, by default the username and password are sent from the client to the ASA in clear text; in addition, the username and password are sent on to the destination web server as well. See the Enabling Secure Authentication of Web Clients for information to secure your credentials.

For FTP, a user has the option of entering the ASA username followed by an at sign (@) and then the FTP username (name1@name2). For the password, the user enters the ASA password followed by an at sign (@) and then the FTP password (password1@password2). For example, enter the following text:

```
name> name1@name2
password> password1@password2
```

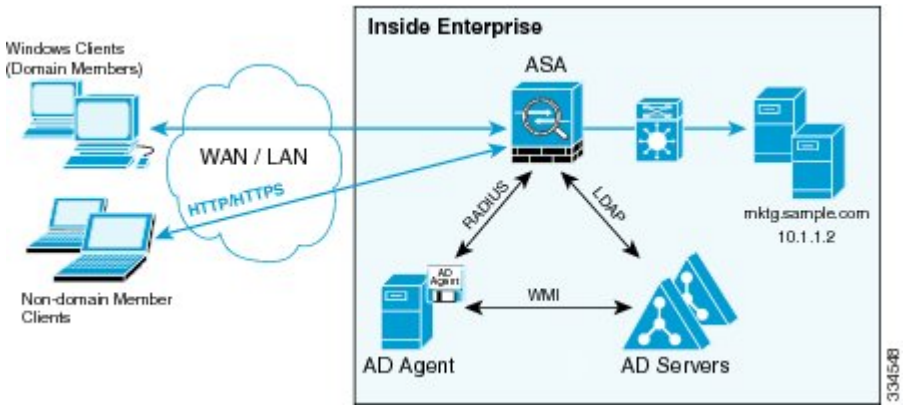
This feature is useful when you have cascaded firewalls that require multiple logins. You can separate several names and passwords by multiple at signs (@).

AAA Prompts and Identity Firewall

In an enterprise, some users log into the network by using other authentication mechanisms, such as authenticating with a web portal (cut-through proxy). For example, users with a Mac and Linux client might log into a web portal (cut-through proxy). Therefore, you must configure the identity firewall to allow these types of authentication in connection with identity-based access policies.

The following figure shows a deployment to support a cut-through proxy authentication captive portal. Active Directory servers and the AD Agent are installed on the main site LAN. However, the identity firewall is configured to support authentication of clients that are not part of the Active Directory domain.

Figure 1: Deployment Supporting Cut-through Proxy Authentication



The ASA designates users logging in through a web portal (cut-through proxy) as belonging to the Active Directory domain with which they authenticated.

The ASA reports users logging in through a web portal (cut-through proxy) to the AD Agent, which distributes the user information to all registered ASA devices. In this case, the identity firewall can associate the users with their Active Directory domain. Specifically, the user identity-IP address mappings of authenticated users are forwarded to all ASA contexts that contain the input interface where packets are received and authenticated.

Users can log in by using HTTP/HTTPS, FTP, Telnet, or SSH. When users log in with these authentication methods, the following guidelines apply:

- For HTTP/HTTPS traffic, an authentication window appears for unauthenticated users.
- For Telnet and FTP traffic, users must log in through the cut-through proxy server and again to the Telnet and FTP servers.
- A user can specify an Active Directory domain while providing login credentials (in the format, domain\username). The ASA automatically selects the associated AAA server group for the specified domain.
- If a user specifies an Active Directory domain while providing login credentials (in the format, domain\username), the ASA parses the domain and uses it to select an authentication server from the AAA servers that have been configured for the identity firewall. Only the username is passed to the AAA server.
- If the backslash (\) delimiter is not found in the login credentials, the ASA does not parse the domain and authentication is conducted with the AAA server that corresponds to the default domain configured for the identity firewall.
- If a default domain or a server group is not configured for that default domain, the ASA rejects the authentication.
- If the domain is not specified, the ASA selects the AAA server group for the default domain that is configured for the identity firewall.

AAA Rules as a Backup Authentication Method

An authentication rule (also known as “cut-through proxy”) controls network access based on the user. Because this function is very similar to an access rule plus an identity firewall, AAA rules can now be used as a backup method of authentication if a user AD login expires or a valid user has not yet logged into AD. For example,

for any user without a valid login, you can trigger a AAA rule. To ensure that the AAA rule is only triggered for users that do not have valid logins, you can specify special usernames in the extended ACL that are used for the access rule and for the AAA rule: None (users without a valid login) and Any (users with a valid login). In the access rule, configure your policy as usual for users and groups, but then include a rule that permits all None users before deny any any; you must permit these users so they can later trigger a AAA rule. Then, configure a AAA rule that does not match Any users (these users are not subject to the AAA rule, and were handled already by the access rule), but matches all None users only to trigger AAA authentication for these users. After the user has successfully logged in via cut-through proxy, the traffic will flow normally again.

Static PAT and HTTP

For HTTP authentication, the ASA checks real ports when static PAT is configured. If it detects traffic destined for real port 80, regardless of the mapped port, the ASA intercepts the HTTP connection and enforces authentication.

For example, assume that outside TCP port 889 is translated to port 80 and that any relevant ACLs permit the traffic:

```
object network obj-192.168.123.10-01
             host 192.168.123.10
             nat (inside,outside) static 10.48.66.155 service tcp 80 889
```

Then when users try to access 10.48.66.155 on port 889, the ASA intercepts the traffic and enforces HTTP authentication. Users see the HTTP authentication page in their web browsers before the ASA allows HTTP connection to complete.

If the local port is different than port 80, as in the following example:

```
object network obj-192.168.123.10-02
             host 192.168.123.10
             nat (inside,outside) static 10.48.66.155 service tcp 111 889
```

Then users do not see the authentication page. Instead, the ASA sends an error message to the web browser, indicating that the user must be authenticated before using the requested service.

When a mapped address is used for static PAT, it is automatically placed into the dynamic PAT pool.

For instance, this configuration,

```
object network my-ftp-server
             host <real-server>
             nat (inside,outside) static <mapped-server> ftp ftp
```

is equivalent to

```
object network my-ftp-server
             host <real-server>
             nat (inside,outside) static <mapped-server> ftp ftp

object network <internal>
             nat (inside,outside) dynamic <mapped-server>
```

The second line ensures that all PAT bindings are accounted for.This accounting is necessary to avoid connection failure from port collision.

As the the mapped address is placed under dynamic PAT, any additional service that is to be accessed through the mapped address, must also be explicitly configured.

For example, the following is the correct configuration for three services through address 192.150.49.10. Additionally, the SMTP and HTTP services also reside at a host with the same address as the mapped address, 192.150.49.10.

```
object network my-ftp-server
             host <real-server>
             nat (inside,outside) static <mapped-server> ftp ftp

object network my-ftp-server
             host "192.150.49.10"
             nat (inside,outside) static 192.150.49.10 smtp smtp

object network my-ftp-server
             host "192.150.49.10"
             nat (inside,outside) static 192.150.49.10 http http
```

Configuring Network Access Authentication

CLI

Procedure

	Command or Action	Purpose
Step 1	**aaa-server** **Example:** `ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+`	Identifies your AAA servers. If you have already identified them, continue to the next step.
Step 2	**access-list** *access_list_name* **extended** { **deny** | **permit** } { **tcp** | **udp** } [*user_argument*] [*security_group_argument*] *source_address_argument* [*port_argument*] [*security_group_argument*] *dest_address_argument* [*port_argument*] **Example:** `ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp`	Creates an ACL that identifies the source addresses and destination addresses of traffic that you want to authenticate. The syntax shown here is just an example. If you specify identity firewall arguments in the ACL, then the following keywords in the ACL are specifically relevant to AAA rules. The keywords **user-group any** and **user-group none** can be specified to support cut-through proxy authentication. • **any** — The ACL matches any IP addresses that has already been associated with any users. • **none** — The ACL matches any IP addresses that has not been associated with any IP address.
Step 3	**aaa authentication match** *acl_name interface_name server_group* [**user-identity**] **Example:** `ciscoasa(config)# aaa authentication match MAIL_AUTH inside AuthOutbound`	Configures authentication. The *acl_name* argument is the name of the ACL that you created in Step 2. The *interface_name* argument is the name of the interface specified with the **nameif** command. The *server_group* argument is the AAA server group that you created in Step 1. **Note** You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the command reference for more information.

	Command or Action	Purpose
		The **user-identity** keyword matches authentication to the identity firewall.
Step 4	**aaa authentication listener http** [**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `ciscoasa(config)# aaa authentication listener http inside redirect`	(Optional) Enables the redirection method of authentication for HTTP or HTTPS connections. The *interface_name* argument is the interface on which you want to enable listening ports. The **port** *portnum* argument specifies the port number on which the ASA listens; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually. Enter this command separately for HTTP and for HTTPS.
Step 5	**aaa local authentication attempts max-fail** *number* **Example:** `ciscoasa(config)# aaa local authentication attempts max-fail 7`	(Optional) Uses the local database for network access authentication and limits the number of consecutive failed login attempts that the ASA allows any given user account (with the exception of users with a privilege level of 15. This feature does not affect level 15 users). The *number* argument value is between 1 and 16. **Tip** To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.

ASDM

Step 1 In the Configuration > Firewall > AAA Rules pane, choose **Add** > **Add Authentication Rule**.

The Add Authentication Rule dialog box appears.

Step 2 In the Interface drop-down list, choose the interface for applying the rule.

Tip In the Action field, click one of the following, depending on the implementation:

- **Authenticate**
- **Do not Authenticate**

Step 3 In the AAA Server Group drop-down list, choose a server group. To add a AAA server to the server group, click **Add Server**.

If you chose LOCAL for the AAA server group, you can optionally add a new user by clicking **Add User**. See the .

Step 4 In the Source field, add the source IP address, or click the ellipsis (...) to choose an IP address already defined in ASDM.

Step 5 In the Destination field, enter the destination IP address, or click the ellipsis (...) to choose an IP address already defined in ASDM.

Step 6 In the Service field, enter an IP service name or number for the destination service, or click the ellipsis (...) to choose a service.

Step 7 (Optional) In the Description field, enter a description.

Step 8 (Optional) Click **More Options** to do any of the following:

- To specify a source service for TCP or UDP, enter a TCP or UDP service in the Source Service field.
- The destination service and source service must be the same. Copy and paste the destination Service field to the Source Service field.
- To make the rule inactive, clear the **Enable Rule** check box.

  You may not want to remove a rule, but instead turn it off.
- To set a time range for the rule, In the Time Range drop-down list, choose an existing time range. To add a new time range, click the ellipsis (...).

Step 9 Click **OK**.

The Add Authentication Rule dialog box closes and the rule appears in the AAA Rules table.

Step 10 Click **Apply**.

The changes are saved to the running configuration.

Example

The following example authenticates all inside HTTP traffic and SMTP traffic:

```
ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp
ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq www
ciscoasa(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
ciscoasa(config)# aaa authentication listener http inside redirect
```

The following example authenticates Telnet traffic from the outside interface to a particular server (209.165.201.5):

```
ciscoasa(config)# aaa-server AuthInbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthInbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
ciscoasa(config)# aaa authentication match TELNET_AUTH outside AuthInbound
```

The following example shows a typical cut-through proxy configuration to allow a user to log in through the ASA. In this example, the following conditions apply:

- The ASA IP address is 192.168.123.10.
- The Active Directory domain controller has the IP address 10.1.2.10.

- The end user client has the IP address 192.168.123.10 and uses HTTPS to log in through a web portal.
- The user is authenticated by the Active Directory domain controller via LDAP.
- The ASA uses the inside interface to connect to the Active Directory domain controller on the corporate network.

```
hostname(config)# access-list AUTH extended permit tcp any 192.168.123.10 255.255.255.0 eq
 http
hostname(config)# access-list AUTH extended permit tcp any 192.168.123.10 255.255.255.0 eq
 https
hostname(config)# aaa-server LDAP protocol ldap
hostname(config-aaa-server-group)# aaa-server LDAP (inside) host 10.1.2.10
hostname(config-aaa-server-host)# ldap-base-dn DC=cisco,DC=com
hostname(config-aaa-server-host)# ldap-group-base-dn DC=cisco,DC=com
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)# ldap-login-dn cn=kao,OU=Employees,OU=Cisco
Users,DC=cisco,DC=com
hostname(config-aaa-server-host)# ldap-login-password *****
hostname(config-aaa-server-host)# ldap-over-ssl enable
hostname(config-aaa-server-host)# server-type microsoft
hostname(config-aaa-server-host)# aaa authentication match AUTH inside LDAP
hostname(config)#
hostname(config)# http server enable
hostname(config)# http 0.0.0.0 0.0.0.0 inside
hostname(config)#
hostname(config)# auth-prompt prompt Enter Your Authentication
hostname(config)# auth-prompt accept You are Good
hostname(config)# auth-prompt reject Goodbye
```

In this example, the following guidelines apply:

- In **access-list** commands, you should configure permit user NONE rules before entering the **access-list 100 ex deny any any** command to allow unauthenticated incoming users to trigger AAA cut-through proxy.
- In **access-list AUTH** commands, permit user NONE rules specify that only unauthenticated users can trigger AAA cut-through proxy.

```
hostname(config)# access-list listenerAuth extended permit tcp any any
hostname(config)# aaa authentication match listenerAuth inside ldap
hostname(config)# aaa authentication listener http inside port 8888
hostname(config)# access-list 100 ex permit ip user SAMPLE\user1 any any
hostname(config)# access-list 100 ex deny ip user SAMPLE\user2 any any
hostname(config)# access-list 100 ex permit ip user NONE any any
hostname(config)# access-list 100 ex deny any any
hostname(config)# access-group 100 in interface inside
hostname(config)# aaa authenticate match 100 inside user-identity
```

The following example shows how you can use AAA rules plus identity firewall (cut-through proxy) to authenticate successfully:

```
hostname(config)# access-list 100 ex permit ip user CISCO\xyz any any
hostname(config)# access-list 100 ex deny ip user CISCO\abc any any
hostname(config)# access-list 100 ex permit ip user NONE any any
hostname(config)# access-list 100 ex deny any any
hostname(config)# access-group 100 in interface inside
hostname(config)# access-list 200 ex permit user NONE any any
hostname(config)# aaa authenticate match 200 inside user-identity
```

(ASDM) Enabling the Redirection Method of Authentication for HTTP and HTTPS

This method of authentication enables HTTP(S) listening ports to authenticate network users. When you enable a listening port, the ASA serves an authentication page for direct connections and, by enabling redirection, for through traffic. This method also prevents the authentication credentials from continuing to the destination server. See the ASA Authentication Prompts for more information about the redirection method compared to the basic method.

Step 1 In the Configuration > Firewall > AAA Rules pane, click **Advanced**.

The AAA Rules Advanced Options dialog box appears.

Step 2 Under Interactive Authentication, click **Add**.

The Add Interactive Authentication Entry dialog box appears.

Step 3 For the Protocol, choose either **HTTP** or **HTTPS**. You can enable both by repeating this procedure and creating two separate rules.

Step 4 In the Interface drop-down list, choose the interface on which you want to enable the listener.

Step 5 In the Port drop-down list, choose the port or enter a number.

This is the port that the ASA listens on for direct or redirected traffic; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually.

Step 6 (Optional) Check **Redirect network users for authentication request**.

This option redirects through traffic to an authentication web page served by the ASA. Without this option, only traffic directed to the ASA interface can access the authentication web pages.

Note If you enable the redirect option, you cannot also configure static PAT for the same interface where you translate the interface IP address and the same port that is used for the listener; NAT succeeds, but authentication fails.

Step 7 Click **OK**, and then click **OK** again to close the AAA Rules Advanced Options dialog box.

Step 8 Click **Apply**.

The changes are saved to the running configuration.

Enabling Secure Authentication of Web Clients

If you use HTTP authentication, by default the username and password are sent from the client to the ASA in clear text; in addition, the username and password are sent to the destination web server as well.

The ASA provides the following methods for securing HTTP authentication:

- Enable the redirection method of authentication for HTTP—**CLI**: Use the **aaa authentication listener** command with the **redirect** keyword. **ASDM**: See the (ASDM) Enabling the Redirection Method of Authentication for HTTP and HTTPS. This method prevents the authentication credentials from continuing

to the destination server. See the ASA Authentication Prompts for more information about the redirection method compared to the basic method.

- Enable virtual HTTP—Virtual HTTP lets you authenticate separately with the ASA and with the HTTP server. Even if the HTTP server does not need a second authentication, this command achieves the effect of stripping the basic authentication credentials from the HTTP GET request. See the Authenticating HTTP(S) Connections with a Virtual Server for more information.
- Enable the exchange of usernames and passwords between a web client and the ASA with HTTPS—**CLI**: Use the **aaa authentication secure-http-client** command to enable the exchange of usernames and passwords between a web client and the ASA with HTTPS. **ASDM**: To enable the exchange of usernames and passwords between a web client and the ASA with HTTPS, perform the following steps:

  1. In the Configuration > Firewall > AAA Rules pane, click **Advanced**. The AAA Rules Advanced Options dialog box appears.
  2. Under Secure HTTP, click **Enable Secure HTTP**.
  3. Click **OK**, and then click **OK** again to close the AAA Rules Advanced Options dialog box.
  4. Click **Apply**.

  This is the only method that protects credentials between the client and the ASA, as well as between the ASA and the destination server. You can use this method alone, or in conjunction with either of the other methods so you can maximize your security.

  After enabling this feature, when a user requires authentication when using HTTP, the ASA redirects the HTTP user to an HTTPS prompt. After you authenticate correctly, the ASA redirects you to the original HTTP URL.

  Secured, web-client authentication has the following limitations:

  – A maximum of 64 concurrent HTTPS authentication sessions are allowed. If all 64 HTTPS authentication processes are running, a new connection requiring authentication will not succeed.

  – When the uauth timeout is set to unlimited (**CLI**: the **uauth timeout 0** command), HTTPS authentication might not work. If a browser initiates multiple TCP connections to load a web page after HTTPS authentication, the first connection is let through, but the subsequent connections trigger authentication. As a result, users are continuously presented with an authentication page, even if the correct username and password are entered each time. To work around this, set the uauth timeout to one second (**CLI**: the **timeout uauth 0:0:1** command) (**ASDM**: see the Configuration > Firewall > Advanced > Global Timeouts pane). However, this workaround opens a 1-second window of opportunity that might allow unauthenticated users to go through the firewall if they are coming from the same source IP address.

  Because HTTPS authentication occurs on the SSL port 443, users must not configure an access rule to block traffic from the HTTP client to the HTTP server on port 443. Furthermore, if static PAT is configured for web traffic on port 80, it must also be configured for the SSL port.

  – In the following example, the first set of commands configures static PAT for web traffic, and the second set of commands must be added to support the HTTPS authentication configuration:

```
object network obj-10.130.16.10-01
   host 10.130.16.10
   nat (inside,outside) static 10.132.16.200 service tcp 80 80
object network obj-10.130.16.10-02
   host 10.130.16.10
   nat (inside,outside) static 10.132.16.200 service tcp 443 443
```

Authenticating Directly with the ASA

If you do not want to allow HTTP, HTTPS, Telnet, or FTP through the ASA but want to authenticate other types of traffic, you can authenticate with the ASA directly using HTTP, HTTPS, or Telnet.

Authenticating HTTP(S) Connections with a Virtual Server

If you enabled the redirection method of HTTP and HTTPS authentication in the Configuring Network Access Authentication, then you have also automatically enabled direct authentication.

When you use HTTP authentication on the ASA (see the Configuring Network Access Authentication), the ASA uses basic HTTP authentication by default.

You can change the authentication method so that the ASA redirects HTTP connections to web pages generated by the ASA itself using the (ASDM) Enabling the Redirection Method of Authentication for HTTP and HTTPS .

However, if you continue to use basic HTTP authentication, then you might need the virtual HTTP server when you have cascading HTTP authentications.

If the destination HTTP server requires authentication in addition to the ASA, then virtual HTTP lets you authenticate separately with the ASA (via a AAA server) and with the HTTP server. Without virtual HTTP, the same username and password that you used to authenticate with the ASA is sent to the HTTP server; you are not prompted separately for the HTTP server username and password. Assuming the username and password are not the same for the AAA and HTTP servers, then the HTTP authentication fails.

This feature redirects all HTTP connections that require AAA authentication to the virtual HTTP server on the ASA. The ASA prompts for the AAA server username and password. After the AAA server authenticates the user, the ASA redirects the HTTP connection back to the original server, but it does not include the AAA server username and password. Because the username and password are not included in the HTTP packet, the HTTP server prompts the user separately for the HTTP server username and password.

For inbound users (from lower security to higher security), you must also include the virtual HTTP address as a destination interface in the access rule applied to the source interface. Moreover, you must add a static NAT rule for the virtual HTTP IP address, even if NAT is not required. An identity NAT rule is typically used (where you translate the address to itself).

For outbound users, there is an explicit permit for traffic, but if you apply an access rule to an inside interface, be sure to allow access to the virtual HTTP address. A static NAT rule is not required.

Note Do not set the uauth timeout duration to 0 seconds when using virtual HTTP, because this setting prevents HTTP connections to the real web server.

You can authenticate directly with the ASA at the following URLs when you enable AAA for the interface:

```
http://interface_ip[:port]/netaccess/connstatus.html
https://interface_ip[:port]/netaccess/connstatus.html
```

CLI

Procedure

	Command or Action	Purpose
Step 1	**aaa authentication listener http** [**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `ciscoasa(config)# aaa authentication listener http inside redirect`	(Optional) Enables the redirection method of authentication for HTTP or HTTPS connections. The *interface_name* argument is the interface on which you want to enable listening ports. The **port** *portnum* argument specifies the port number on which the ASA listens; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually. Enter this command separately for HTTP and for HTTPS.

Step 2	**virtual http**
	Example:
	`ciscoasa(config)# virtual http`
		Redirects all HTTP connections that require AAA authentication to the virtual HTTP server on the ASA. The ASA prompts for the AAA server username and password. After the AAA server authenticates the user, the ASA redirects the HTTP connection back to the original server, but it does not include the AAA server username and password. Because the username and password are not included in the HTTP packet, the HTTP server prompts the user separately for the HTTP server username and password.
		For inbound users (from lower security to higher security), you must also include the virtual HTTP address as a destination interface in the ACL applied to the source interface. In addition, you must add a static NAT command for the virtual HTTP IP address, even if NAT is not required. An identity NAT command is typically used (where you translate the address to itself).
		For outbound users, there is an explicit permit for traffic, but if you apply an ACL to an inside interface, be sure to allow access to the virtual HTTP address. A static statement is not required.
		Note Do not set the timeout uauth command duration to 0 seconds when using the virtual http command, because this setting prevents HTTP connections to the actual web server.
		You can authenticate directly with the ASA at the following URLs when you enable AAA for the interface:
		`http://interface_ip[:port]/netaccess/connstatus.html`
		`https://interface_ip[:port]/netaccess/connstatus.html`

	Command or Action	Purpose
		Without virtual HTTP, the same username and password that you used to authenticate with the ASA are sent to the HTTP server; you are not prompted separately for the HTTP server username and password. Assuming the username and password are not the same for the AAA and HTTP servers, then the HTTP authentication fails.

ASDM

Step 1 In the Configuration > Firewall > Advanced > Virtual Access > Virtual HTTP Server area, check the **Enable** check box.

Step 2 In the Virtual HTTP Server field, add the IP address of the virtual HTTP server.

Make sure this address is an unused address that is routed to the ASA. For example, if you perform NAT for inside addresses accessing an outside server, and you want to provide outside access to the virtual HTTP server, you can use one of the global NAT addresses for the virtual HTTP server address.

Step 3 (Optional) If you are using text-based browsers, where redirection does not happen automatically, check the **Display redirection warning** check box. This enables an alert to notify users when the HTTP connection is being redirected.

Step 4 Click **Apply**.

The virtual server is added and the changes are saved to the running configuration.

Authenticating Telnet Connections with a Virtual Server

Although you can configure network access authentication for any protocol or service (**CLI**: see the **aaa authentication match** or **aaa authentication include** command) (**ASDM**: see the Configuring Network Access Authentication), you can authenticate directly with HTTP, Telnet, or FTP only. A user must first authenticate with one of these services before other traffic that requires authentication is allowed through. If you do not want to allow HTTP, Telnet, or FTP traffic through the ASA, but want to authenticate other types of traffic, you can configure virtual Telnet; the user Telnets to a given IP address configured on the ASA, and the ASA issues a Telnet prompt.

When an unauthenticated user connects to the virtual Telnet IP address, the user is challenged for a username and password, and then authenticated by the AAA server. After the user is authenticated, the message "Authentication Successful" appears. Then the user can successfully access other services that require authentication.

For inbound users (from lower security to higher security), you must also include the virtual Telnet address as a destination interface in the access rule applied to the source interface. In addition, you must add a static NAT rule for the virtual Telnet IP address, even if NAT is not required. An identity NAT rule is typically used (where you translate the address to itself).

For outbound users, there is an explicit permit for traffic, but if you apply an access rule to an inside interface, be sure to allow access to the virtual Telnet address. A static NAT rule is not required.

To log out from the ASA, reconnect to the virtual Telnet IP address; you are prompted to log out.

CLI

Procedure

	Command or Action	Purpose
Step 1	**virtual telnet** *ip_address* **Example:** `ciscoasa(config)# virtual telnet 209.165.202.129`	Configures a virtual Telnet server. The *ip_address* argument sets the IP address for the virtual Telnet server. Make sure this address is an unused address that is routed to the ASA. You must configure authentication for Telnet access to the virtual Telnet address as well as the other services that you want to authenticate using the **authentication match** or **aaa authentication include** command. When an unauthenticated user connects to the virtual Telnet IP address, the user is challenged for a username and password, and then authenticated by the AAA server. Once authenticated, the user sees the message "Authentication Successful." Then, the user can successfully access other services that require authentication. For inbound users (from lower security to higher security), you must also include the virtual Telnet address as a destination interface in the ACL applied to the source interface. In addition, you must add a static NAT command for the virtual Telnet IP address, even if NAT is not required. An identity NAT command is typically used (where you translate the address to itself). For outbound users, there is an explicit permit for traffic, but if you apply an ACL to an inside interface, be sure to allow access to the virtual Telnet address. A static statement is not required. To log out from the ASA, reconnect to the virtual Telnet IP address; you are then prompted to log out.

ASDM

Step 1 In the Configuration > Firewall > Advanced > Virtual Access > Virtual Telnet Server area, check the **Enable** check box.

Step 2 In the Virtual Telnet Server field, enter the IP address of the virtual Telnet server.

Make sure that this address is an unused address that is routed to the ASA. For example, if you perform NAT for inside addresses accessing an outside server, and you want to provide outside access to the virtual HTTP server, you can use one of the global NAT addresses for the virtual HTTP server address.

Step 3 Click **Apply**.

The virtual server is added and the changes are saved to the running configuration.

Example

The following example shows how to enable virtual Telnet together with AAA authentication for other services:

```
ciscoasa(config)# virtual telnet 209.165.202.129
ciscoasa(config)# access-list ACL-IN extended permit tcp any host 209.165.200.225 eq smtp
ciscoasa(config)# access-list ACL-IN remark This is the SMTP server on the inside
ciscoasa(config)# access-list ACL-IN extended permit tcp any host 209.165.202.129 eq
telnet
ciscoasa(config)# access-list ACL-IN remark This is the virtual Telnet address
ciscoasa(config)# access-group ACL-IN in interface outside
ciscoasa(config)# network object obj-209.165.202.129-01
ciscoasa(config-network-object)# host 209.165.202.129
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.202.129
ciscoasa(config)# access-list AUTH extended permit tcp any host 209.165.200.225 eq smtp
ciscoasa(config)# access-list AUTH remark This is the SMTP server on the inside
ciscoasa(config)# access-list AUTH extended permit tcp any host 209.165.202.129 eq telnet
ciscoasa(config)# access-list AUTH remark This is the virtual Telnet address
ciscoasa(config)# aaa authentication match AUTH outside tacacs+
```

(ASDM) Configuring the Authentication Proxy Limit

You can manually configure the uauth session limit by setting the maximum number of concurrent proxyconnections allowed per user.

Step 1 Choose **Configuration** > **Firewall** > **AAA Rules**, then click **Advanced**.

The AAA Rules Advanced Options dialog box appears.

Step 2 In the Proxy Limit area, check the **Enable Proxy Limit** check box.

Step 3 In the Proxy Limit field, enter the number of concurrent proxy connections allowed per user, from 1 to 128.

Step 4 Click **OK**, then click **Apply**.

The changes are saved to the running configuration.

Configuring Authorization for Network Access

After a user authenticates for a given connection, the ASA can use authorization to further control traffic from the user.

This section includes the following topics:

Configuring TACACS+ Authorization

You can configure the ASA to perform network access authorization with TACACS+. **CLI**: You identify the traffic to be authorized by specifying ACLs that authorization rules must match. Alternatively, you can identify the traffic directly in authorization rules themselves.

Note Using ACLs to identify traffic to be authorized can greatly reduced the number of authorization commands that you must enter. This is because each authorization rule that you enter can specify only one source and destination subnet and service, whereas an ACL can include many entries.

Authentication and authorization statements are independent; however, any unauthenticated traffic matched by an authorization rule will be denied. For authorization to succeed:

1. A user must first authenticate with the ASA.

   Because a user at a given IP address only needs to authenticate one time for all rules and types, if the authentication session has not expired, authorization can occur even if the traffic is not matched by an authentication rule.

2. After a user authenticates, the ASA checks the authorization rules for matching traffic.
3. If the traffic matches the authorization rule, the ASA sends the username to the TACACS+ server.
4. The TACACS+ server responds to the ASA with a permit or a deny for that traffic, based on the user profile.
5. The ASA enforces the authorization rule in the response.

See the documentation for your TACACS+ server for information about configuring network access authorizations for a user.

CLI

Procedure

	Command or Action	Purpose
Step 1	**aaa-server** **Example:** `ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+`	Identifies your AAA servers. If you have already identified them, continue to the next step.
Step 2	**access-list** **Example:** `ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp`	Creates an ACL that identifies the source addresses and destination addresses of traffic you want to authenticate. The **permit** ACEs mark matching traffic for authentication, while **deny** entries exclude matching traffic from authentication. Be sure to include the destination ports for either HTTP, HTTPS, Telnet, or FTP in the ACL, because the user must authenticate with one of these services before other services are allowed through the ASA.
Step 3	**aaa authentication match** *acl_name interface_name server_group* **Example:** `ciscoasa(config)# aaa authentication match MAIL_AUTH inside AuthOutbound`	Configures authentication. The *acl_name* argument is the name of the ACL that you created in Step 2., The *interface_name* argument is the name of the interface specified with the **nameif** command, and the *server_group* argument is the AAA server group that you created in Step 1.

	Command or Action	Purpose
		Note You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the command reference for more information.
Step 4	**aaa authentication listener http** [**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `ciscoasa(config)# aaa authentication listener http inside redirect`	(Optional) Enables the redirection method of authentication for HTTP or HTTPS connections. The *interface_name* argument is the interface on which you want to enable listening ports. The **port** *portnum* argument specifies the port number on which the ASA listens; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually. Enter this command separately for HTTP and for HTTPS.
Step 5	**aaa local authentication attempts max-fail** *number* **Example:** `ciscoasa(config)# aaa local authentication attempts max-fail 7`	(Optional) Uses the local database for network access authentication and limits the number of consecutive failed login attempts that the ASA allows any given user account (with the exception of users with a privilege level of 15. This feature does not affect level 15 users). The *number* argument value is between 1 and 16. **Tip** To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.
Step 6	**access-list** **Example:** `ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet`	Create an ACL that identifies the source addresses and destination addresses of traffic that you want to authorize. The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization. The ACL that you use for authorization matching should include rules that are equal to or a subset of the rules in the ACL used for authentication matching. **Note** If you have configured authentication and want to authorize all the traffic being authenticated, you can use the same ACL that you created for use with the **aaa authentication match** command.
Step 7	**aaa authorization match** *acl_name interface_name server_group* **Example:**	Enables authorization. The *acl_name* argument is the name of the ACL you created in Step 6, the *interface_name* argument is the name of the

	Command or Action	Purpose
	`ciscoasa(config)# aaa authentication match TELNET_AUTH inside AuthOutbound`	interface as specified with the **nameif** command or by default, and the *server_group* argument is the AAA server group that you created when you enabled authentication. **Note** Alternatively, you can use the **aaa authorization include** command (which identifies traffic within the command) but you cannot use both methods in the same configuration. See the command reference for more information.

ASDM

Step 1 Enable authentication. For more information, see the Configuring Network Access Authentication. If you have already enabled authentication, continue to the next step.

Step 2 In the Configuration > Firewall > AAA Rules pane, choose **Add** > **Add Authorization Rule**.

Step 3 In the Interface drop-down list, choose the interface for applying the rule.

Step 4 In the Action field, click one of the following, depending on the implementation:

- **Authorize**
- **Do not Authorize**

Step 5 In the AAA Server Group drop-down list, choose a server group. To add a AAA server to the server group, click **Add Server**.

Only TACACS+ servers are supported.

Step 6 In the Source field, add the source IP address, or click the ellipsis (...) to choose an IP address already defined in ASDM.

Step 7 In the Destination field, enter the destination IP address, or click the ellipsis (...) to choose an IP address already defined in ASDM.

Step 8 In the Service field, enter an IP service name or number for the destination service, or click the ellipsis (...) to choose a service.

Step 9 (Optional) In the Description field, enter a description.

Step 10 (Optional) Click **More Options** to do any of the following:

- To specify a source service for TCP or UDP, enter a TCP or UDP service in the Source Service field.
- The destination service and source service must be the same. Copy and paste the Destination Service field content into the Source Service field.
- To make the rule inactive, clear the **Enable Rule** check box.

  You may not want to remove a rule, but instead turn it off.
- To set a time range for the rule, in the Time Range drop-down list, choose an existing time range.

  To add a new time range, click the ellipsis (...).

Step 11 Click **OK**.

The Add Authorization Rule dialog box closes, and the rule appears in the AAA Rules table.

Step 12 Click **Apply**.

The changes are saved to the running configuration.

Example

The following example authenticates and authorizes inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization.

```
ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
ciscoasa(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
ciscoasa(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
```

Configuring RADIUS Authorization

When authentication succeeds, the RADIUS protocol returns user authorizations in the access-accept message sent by a RADIUS server. For more information about configuring authentication, see the Configuring Network Access Authentication.

When you configure the ASA to authenticate users for network access, you are also implicitly enabling RADIUS authorizations; therefore, this section contains no information about configuring RADIUS authorization on the ASA. It does provide information about how the ASA handles ACL information received from RADIUS servers.

You can configure a RADIUS server to download an ACL to the ASA or an ACL name at the time of authentication. The user is authorized to do only what is permitted in the user-specific ACL.

Note If you have enabled the Per User Override Setting (**ASDM**: see the **Configuration** > **Firewall** > **Access Rules** > **Advanced** > **Access Rules Advanced Options** dialog box; **CLI**: see the **access-group per-user-override** keyword), be aware of the following effects on authorization by user-specific ACLs:

- Without the per-user-override feature, traffic for a user session must be permitted by both the interface ACL and the user-specific ACL.
- With the per-user-override feature, the user-specific ACL determines what is permitted.

This section includes the following topics:

Configuring a RADIUS Server to Send Downloadable Access Control Lists

This section describes how to configure Cisco Secure ACS or a third-party RADIUS server and includes the following topics:

About the Downloadable ACL Feature and Cisco Secure ACS

Downloadable ACLs is the most scalable means of using Cisco Secure ACS to provide the appropriate ACLs for each user. It provides the following capabilities:

- Unlimited ACL size—Downloadable ACLs are sent using as many RADIUS packets as required to transport the full ACL from Cisco Secure ACS to the ASA.
- Simplified and centralized management of ACLs—Downloadable ACLs enable you to write a set of ACLs once and apply it to many user or group profiles and distribute it to many ASAs.

This approach is most useful when you have very large ACL sets that you want to apply to more than one Cisco Secure ACS user or group; however, its ability to simplify Cisco Secure ACS user and group management makes it useful for ACLs of any size.

The ASA receives downloadable ACLs from Cisco Secure ACS using the following process:

1. The ASA sends a RADIUS authentication request packet for the user session.
2. If Cisco Secure ACS successfully authenticates the user, Cisco Secure ACS returns a RADIUS access-accept message that includes the internal name of the applicable downloadable ACL. The Cisco IOS cisco-av-pair RADIUS VSA (vendor 9, attribute 1) includes the following attribute-value pair to identify the downloadable ACL set:

   ```
   ACS:CiscoSecure-Defined-ACL=acl-set-name
   ```

   where *acl-set-name* is the internal name of the downloadable ACL, which is a combination of the name assigned to the ACL by the Cisco Secure ACS administrator and the date and time that the ACL was last modified.
3. The ASA examines the name of the downloadable ACL and determines if it has previously received the named downloadable ACL.

   – If the ASA has previously received the named downloadable ACL, communication with Cisco Secure ACS is complete and the ASA applies the ACL to the user session. Because the name of the downloadable ACL includes the date and time that it was last modified, matching the name sent by Cisco Secure ACS to the name of an ACL previously downloaded means that the ASA has the most recent version of the downloadable ACL.

   – If the ASA has not previously received the named downloadable ACL, it may have an out-of-date version of the ACL or it may not have downloaded any version of the ACL. In either case, the ASA issues a RADIUS authentication request using the downloadable ACL name as the username in the RADIUS request and a null password attribute. In a cisco-av-pair RADIUS VSA, the request also includes the following attribute-value pairs:

   ```
   AAA:service=ip-admission
   AAA:event=acl-download
   ```

   In addition, the ASA signs the request with the Message-Authenticator attribute (IETF RADIUS attribute 80).
4. After receipt of a RADIUS authentication request that has a username attribute that includes the name of a downloadable ACL, Cisco Secure ACS authenticates the request by checking the Message-Authenticator attribute. If the Message-Authenticator attribute is missing or incorrect, Cisco Secure ACS ignores the

request. The presence of the Message-Authenticator attribute prevents malicious use of a downloadable ACL name to gain unauthorized network access. The Message-Authenticator attribute and its use are defined in RFC 2869, RADIUS Extensions, available at http://www.ietf.org.

5. If the ACL required is less than approximately 4 KB in length, Cisco Secure ACS responds with an access-accept message that includes the ACL. The largest ACL that can fit in a single access-accept message is slightly less than 4 KB, because part of the message must be other required attributes.

   Cisco Secure ACS sends the downloadable ACL in a cisco-av-pair RADIUS VSA. The ACL is formatted as a series of attribute-value pairs that each include an ACE and are numbered serially:

```
ip:inacl#1=ACE-1
ip:inacl#2=ACE-2
.
.
.
ip:inacl#n=ACE-n
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

6. If the ACL required is more than approximately 4 KB in length, Cisco Secure ACS responds with an access-challenge message that includes a portion of the ACL, formatted as described previously, and a State attribute (IETF RADIUS attribute 24), which includes control data used by Cisco Secure ACS to track the progress of the download. Cisco Secure ACS fits as many complete attribute-value pairs into the cisco-av-pair RADIUS VSA as it can without exceeding the maximum RADIUS message size.

   The ASA stores the portion of the ACL received and responds with another access-request message that includes the same attributes as the first request for the downloadable ACL, plus a copy of the State attribute received in the access-challenge message.

   This process repeats until Cisco Secure ACS sends the last of the ACL in an access-accept message.

Configuring Cisco Secure ACS for Downloadable ACLs

You can configure downloadable ACLs on Cisco Secure ACS as a shared profile component and then assign the ACL to a group or to an individual user.

The ACL definition consists of one or more ASA commands that are similar to the extended **access-list** command (see command reference), except without the following prefix:

access-list *acl_name* **extended**

The following example is a downloadable ACL definition on Cisco Secure ACS version 3.3:

```
+-----------------------------------------+
| Shared profile Components               |
|                                         |
| Downloadable IP ACLs Content            |
|                                         |
| Name: acs_ten_acl                       |
|                                         |
| ACL Definitions                         |
|                                         |
| permit tcp any host 10.0.0.254          |
| permit udp any host 10.0.0.254          |
| permit icmp any host 10.0.0.254         |
| permit tcp any host 10.0.0.253          |
| permit udp any host 10.0.0.253          |
| permit icmp any host 10.0.0.253         |
| permit tcp any host 10.0.0.252          |
| permit udp any host 10.0.0.252          |
| permit icmp any host 10.0.0.252         |
```

```
| permit ip any any                          |
+--------------------------------------------+
```

For more information about creating downloadable ACLs and associating them with users, see the user guide for your version of Cisco Secure ACS.

On the ASA, the downloaded ACL has the following name:

```
#ACSACL#-ip-acl_name-number
```

The *acl_name* argument is the name that is defined on Cisco Secure ACS (*acs_ten_acl* in the preceding example), and *number* is a unique version ID generated by Cisco Secure ACS.

The downloaded ACL on the ASA consists of the following lines:

```
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit ip any any
```

Configuring Any RADIUS Server for Downloadable ACLs

You can configure any RADIUS server that supports Cisco IOS RADIUS VSAs to send user-specific ACLs to the ASA in a Cisco IOS RADIUS cisco-av-pair VSA (vendor 9, attribute 1).

In the cisco-av-pair VSA, configure one or more ACEs that are similar to the **access-list extended** command (see command reference), except that you replace the following command prefix:

access-list *acl_name* **extended**

with the following text:

ip:inacl#nnn**=**

The *nnn* argument is a number in the range from 0 to 999999999 that identifies the order of the command statement to be configured on the ASA. If this parameter is omitted, the sequence value is 0, and the order of the ACEs inside the cisco-av-pair RADIUS VSA is used.

The following example is an ACL definition as it should be configured for a cisco-av-pair VSA on a RADIUS server:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#99=deny tcp any any
ip:inacl#2=permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#100=deny udp any any
ip:inacl#3=permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

For information about making unique per user the ACLs that are sent in the cisco-av-pair attribute, see the documentation for your RADIUS server.

On the ASA, the downloaded ACL name has the following format:

```
AAA-user-username
```

The *username* argument is the name of the user that is being authenticated.

The downloaded ACL on the ASA consists of the following lines. Notice the order based on the numbers identified on the RADIUS server.

```
access-list AAA-user-bcham34-79AD4A08 permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list AAA-user-bcham34-79AD4A08 permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list AAA-user-bcham34-79AD4A08 permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list AAA-user-bcham34-79AD4A08 deny tcp any any
access-list AAA-user-bcham34-79AD4A08 deny udp any any
```

Downloaded ACLs have two spaces between the word “access-list” and the name. These spaces serve to differentiate a downloaded ACL from a local ACL. In this example, “79AD4A08” is a hash value generated by the ASA to help determine when ACL definitions have changed on the RADIUS server.

Converting Wildcard Netmask Expressions in Downloadable ACLs

If a RADIUS server provides downloadable ACLs to Cisco VPN 3000 series concentrators as well as to the ASA, you may need the ASA to convert wildcard netmask expressions to standard netmask expressions. This is because Cisco VPN 3000 series concentrators support wildcard netmask expressions, but the ASA only supports standard netmask expressions. Configuring the ASA to convert wildcard netmask expressions helps minimize the effects of these differences on how you configure downloadable ACLs on your RADIUS servers. Translation of wildcard netmask expressions means that downloadable ACLs written for Cisco VPN 3000 series concentrators can be used by the ASA without altering the configuration of the downloadable ACLs on the RADIUS server.

You configure ACL netmask conversion on a per-server basis: **CLI**: using the **acl-netmask-convert** command, available in the aaa-server configuration mode; when you add a server to a server group in the Configuration > Device Management > Users/AAA > AAA Server Groups > AAA Server Groups area.

Configuring a RADIUS Server to Download Per-User Access Control List Names

To download a name for an ACL that you already created on the ASA from the RADIUS server when a user authenticates, configure the IETF RADIUS filter-id attribute (attribute number 11) as follows:

```
filter-id=acl_name
```

Note In Cisco Secure ACS, the values for filter-id attributes are specified in boxes in the HTML interface, omitting **filter-id=** and entering only *acl_name*.

For information about making the filter-id attribute value unique per user, see the documentation for your RADIUS server.

Configuring Accounting for Network Access

The ASA can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the ASA. If that traffic is also authenticated, then the AAA server can maintain accounting information by username. If the traffic is not authenticated, the AAA server can maintain accounting information by IP address. Accounting information includes session start and stop times, username, the number of bytes that pass through the ASA for the session, the service used, and the duration of each session.

CLI

Procedure

	Command or Action	Purpose
Step 1	**access-list** **Example:** `ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet`	If you want the ASA to provide accounting data per user, you must enable authentication. For more information, see the Configuring Network Access Authentication . If you want the ASA to provide accounting data per IP address, enabling authentication is not necessary. Creates an ACL that identifies the source addresses and destination addresses of traffic for which you want accounting data. The permit ACEs mark matching traffic for accounting, while deny entries exclude matching traffic from accounting. **Note** If you have configured authentication and want accounting data for all the traffic being authenticated, you can use the same ACL that you created for use with the **aaa authentication match** command.
Step 2	**aaa accounting match** *acl_name interface_name server_group* **Example:** `ciscoasa(config)# aaa accounting match SERVER_AUTH inside AuthOutbound`	Enables accounting. The *acl_name* argument is the ACL name set in the **access-list** command. The *interface_name* argument is the interface name set in the **nameif** command. The *server_group* argument is the server group name set in the **aaa-server** command. **Note** Alternatively, you can use the **aaa accounting include** command (which identifies traffic within the command), but you cannot use both methods in the same configuration. See the command reference for more information.

ASDM

Step 1 If you want the ASA to provide accounting data per user, you must enable authentication. For more information, see the Configuring Network Access Authentication. If you want the ASA to provide accounting data per IP address, enabling authentication is not necessary and you can continue to the next step.

Step 2 In the Configuration > Firewall > AAA Rules pane, choose **Add** > **Add Accounting Rule**.

The Add Accounting Rule dialog box appears.

Step 3 In the Interface drop-down list, choose the interface for applying the rule.

Step 4 In the Action field, click one of the following, depending on the implementation:

- **Account**
- **Do not Account**

Step 5 In the AAA Server Group drop-down list, choose a server group. To add a AAA server to the server group, click **Add Server**.

Step 6 In the Source field, enter the source IP address, or click the ellipsis (...) to choose an IP address already defined in ASDM.

Step 7 In the Destination field, enter the destination IP address, or click the ellipsis (...) to choose an IP address already defined in ASDM.

Step 8 In the Service field, enter an IP service name or number for the destination service, or click the ellipsis (...) to choose a service.

Step 9 (Optional) In the Description field, enter a description.

Step 10 (Optional) Click **More Options** to do any of the following:

- To specify a source service for TCP or UDP, enter a TCP or UDP service in the Source Service field.
- The destination service and source service must be the same. Copy and paste the Destination Service field content to the Source Service field.
- To make the rule inactive, clear the **Enable Rule** check box.

  You may not want to remove a rule, but instead turn it off.
- To set a time range for the rule, In the Time Range drop-down list, choose an existing time range. To add a new time range, click the ellipsis (...).

Step 11 Click **OK**.

The Add Accounting Rule dialog box closes and the rule appears in the AAA Rules table.

Step 12 Click **Apply**.

The changes are saved to the running configuration.

Example

The following example authenticates, authorizes, and accounts for inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization and accounting.

```
ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
ciscoasa(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
ciscoasa(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
```

```
ciscoasa(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
ciscoasa(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```

Using MAC Addresses to Exempt Traffic from Authentication and Authorization

The ASA can exempt from authentication and authorization any traffic from specific MAC addresses. For example, if the ASA authenticates TCP traffic originating on a particular network, but you want to allow unauthenticated TCP connections from a specific server, you would use a MAC exempt rule to exempt from authentication and authorization any traffic from the server specified by the rule.

This feature is particularly useful to exempt devices such as IP phones that cannot respond to authentication prompts.

Guidelines

The order of entries matters, because the packet uses the first entry it matches, instead of a best match scenario. If you have a **permit** entry, and you want to deny an address that is allowed by the **permit** entry, be sure to enter the **deny** entry before the **permit** entry.

CLI

Procedure

	Command or Action	Purpose
Step 1	**mac-list** *id* {**deny** | **permit**} *mac macmask* **Example:** `ciscoasa(config)# mac-list abc permit 00a0.c95d.0282 ffff.ffff.ffff`	Configures a MAC list. The *id* argument is the hexadecimal number that you assign to the MAC list. To group a set of MAC addresses, enter the **mac-list** command as many times as needed with the same ID value. Because you can only use one MAC list for AAA exemption, be sure that your MAC list includes all the MAC addresses that you want to exempt. You can create multiple MAC lists, but you can only use one at a time. The *mac* argument specifies the source MAC address in 12-digit hexadecimal form; that is, nnnn.nnnn.nnnn. The *macmask* argument specifies the portion of the MAC address that should be used for matching. For example, ffff.ffff.ffff matches the MAC address exactly. ffff.ffff.0000 matches only the first 8 digits.
Step 2	**aaa mac-exempt match** *id* **Example:** `ciscoasa(config)# aaa mac-exempt match 1`	Exempts traffic for the MAC addresses specified in a particular MAC list. The *id* argument is the string identifying the MAC list that includes the MAC addresses whose traffic is to be exempt from authentication and authorization.

	Command or Action	Purpose
		You can only enter one instance of the **aaa mac-exempt match** command.

ASDM

Step 1 In the Configuration > Firewall > AAA Rules pane, choose **Add** > **Add MAC Exempt Rule**.

The Add MAC Exempt Rule dialog box appears.

Step 2 In the Action drop-down list, click one of the following options, depending on the implementation:

- **MAC Exempt**
- **No MAC Exempt**

The MAC Exempt option allows traffic from the MAC address without having to authenticate or authorize. The No MAC Exempt option specifies a MAC address that is not exempt from authentication or authorization. You might need to add a **deny** entry if you permit a range of MAC addresses using a MAC address mask such as ffff.ffff.0000, and you want to force a MAC address in that range to be authenticated and authorized.

Step 3 In the MAC Address field, specify the source MAC address in 12-digit hexadecimal form; that is, nnnn.nnnn.nnnn.

Step 4 In the MAC Mask field, specify the portion of the MAC address that should be used for matching. For example, ffff.ffff.ffff matches the MAC address exactly. ffff.ffff.0000 matches only the first 8 digits.

Step 5 Click **OK**.

The Add MAC Exempt Rule dialog box closes and the rule appears in the AAA Rules table.

Step 6 Click **Apply**.

The changes are saved to the running configuration.

Example

The following example bypasses authentication for a single MAC address:

```
ciscoasa(config)# mac-list abc permit 00a0.c95d.0282 ffff.ffff.ffff
ciscoasa(config)# aaa mac-exempt match abc
```

The following example bypasses authentication for all Cisco IP Phones, which have the hardware ID 0003.E3:

```
ciscoasa(config)# mac-list acd permit 0003.E300.0000 FFFF.FF00.0000
ciscoasa(config)# aaa mac-exempt match acd
```

The following example bypasses authentication for a a group of MAC addresses except for 00a0.c95d.02b2. Enter the **deny** statement before the **permit** statement, because 00a0.c95d.02b2 matches the **permit** statement as well, and if it is first, the **deny** statement will never be matched.

```
ciscoasa(config)# mac-list 1 deny 00a0.c95d.0282 ffff.ffff.ffff
ciscoasa(config)# mac-list 1 permit 00a0.c95d.0000 ffff.ffff.0000
ciscoasa(config)# aaa mac-exempt match 1
```

Feature History for AAA Rules

Table 1: Feature History for AAA Rules lists each feature change and the platform release in which it was implemented. ASDM is backwards-compatible with multiple platform releases, so the specific ASDM release in which support was added is not listed.

Table 1: Feature History for AAA Rules

Feature Name	Platform Releases	Feature Information
AAA Rules	7.0(1)	AAA Rules describe how to enable AAA for network access. We introduced the following commands: **aaa authentication match, aaa authentication include | exclude, aaa authentication listener http[s], aaa local authentication attempts max-fail, virtual http, virtual telnet, aaa authentication secure-http-client, aaa authorization match, aaa accounting match, aaa mac-exempt match.** We introduced the following screens: Configuration > Firewall > AAA Rules Configuration > Firewall > Advanced > Virtual Access.
Authentication using Cut-Through Proxy	9.0(1)	You can authenticate using AAA rules in conjunction with the Identity Firewall feature. We modified the following command: **aaa authentication match**.