CHAPTER 38

Configuring AAA Rules for Network Access

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

For information about AAA for management access, see the “Configuring AAA for System Administrators” section on page 37-13.

This chapter includes the following sections:

- AAA Performance, page 38-1
- Licensing Requirements for AAA Rules, page 38-1
- Guidelines and Limitations, page 38-2
- Configuring Authentication for Network Access, page 38-2
- Configuring Authorization for Network Access, page 38-11
- Configuring Accounting for Network Access, page 38-18
- Using MAC Addresses to Exempt Traffic from Authentication and Authorization, page 38-20
- Feature History for AAA Rules, page 38-21

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.

Configuring Authentication for Network Access

This section includes the following topics:

- Information About Authentication, page 38-2
- Configuring Network Access Authentication, page 38-4
- Enabling Secure Authentication of Web Clients, page 38-6
- Authenticating Directly with the ASA, page 38-7

Information About Authentication

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

- One-Time Authentication, page 38-2
- Applications Required to Receive an Authentication Challenge, page 38-2
- ASA Authentication Prompts, page 38-3
- Static PAT and HTTP, page 38-4

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. (See the **timeout uauth** command in the command reference 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 (configured with the **aaa authentication listener** command).

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 (configured with the **aaa authentication listener** command).

Redirection is an improvement over the basic method because it provides an improved user experience when authenticating, and an identical user experience for HTTP and HTTPS in both Easy VPN and firewall modes. It also supports authenticating 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 the **virtual http** command.

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" section on page 38-6 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 (@).

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 access lists 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.

Configuring Network Access Authentication

To configure network access authentication, perform the following steps:

	Command	Purpose
Step 1	**aaa-server** **Example:** `hostname(config)# aaa-server AuthOutbound protocol tacacs+`	Identifies your AAA servers. If you have already identified them, continue to the next step. For more information about identifying AAA servers, see the "Configuring AAA Server Groups" section on page 35-11.
Step 2	**access-list** **Example:** `hostname(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp`	Creates an access list that identifies the source addresses and destination addresses of traffic you want to authenticate. For details, see Chapter 15, "Adding an Extended Access List." 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 access list, because the user must authenticate with one of these services before other services are allowed through the ASA.

	Command	Purpose
Step 3	**aaa authentication match** *acl_name interface_name server_group* **Example:** `hostname(config)# aaa authentication match MAIL_AUTH inside AuthOutbound`	Configures authentication. The *acl_name* argument is the name of the access list 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.
Step 4	**aaa authentication listener http**[**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `hostname(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:** `hostname(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.

Examples

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

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

```
hostname(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):

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

For more information about authentication, see the "Information About Authentication" section on page 38-2.

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—Use the **aaa authentication listener** command with the **redirect** keyword. This method prevents the authentication credentials from continuing to the destination server. See the "ASA Authentication Prompts" section on page 38-3 for more information about the redirection method compared to the basic method.
- Enable virtual HTTP—Use the **virtual http** command to authenticateseparately 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" section on page 38-8 for more information.

  Enable the exchange of usernames and passwords between a web client and the ASA with HTTPS—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. 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 16 concurrent HTTPS authentication sessions are allowed. If all 16 HTTPS authentication processes are running, a new connection requiring authentication will not succeed.

  – When **uauth timeout 0** is configured (the **uauth timeout** is set to 0),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 1 second with the **timeout uauth 0:0:1** command. 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-list** command statement 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.

This section includes the following topics:

- Authenticating HTTP(S) Connections with a Virtual Server, page 38-8
- Authenticating Telnet Connections with a Virtual Server, page 38-9

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" section on page 38-4, then you have also automatically enabled direct authentication.

When you use HTTP authentication on the ASA (see the"Configuring Network Access Authentication" section on page 38-4), the ASA uses basic HTTP authentication by default.

To continue to use basic HTTP authentication, and to enable direct authentication for HTTP and HTTPS, enter the following command:

Command	Purpose
aaa authentication listener http[**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `hostname(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.

If the destination HTTP server requires authentication in addition to the ASA, then to authenticate separately with the ASA (via a AAA server) and with the HTTP server, enter the following command:

Command	Purpose
virtual http **Example:** `hostname(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 access list 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 access list 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** 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.

Authenticating Telnet Connections with a Virtual Server

Although you can configure network access authentication for any protocol or service (see the **aaa authentication match** or **aaa authentication include** command), 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.

To configure a virtual Telnet server, enter the following command:

Command	Purpose
virtual telnet *ip_address* **Example:** `hostname(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 access list 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 access list 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.

Examples

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

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

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, page 38-11
- Configuring RADIUS Authorization, page 38-14

Configuring TACACS+ Authorization

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

Tip Using access lists 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 access list 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.

To configure TACACS+ authorization, perform the following steps:

	Command	Purpose
Step 1	**aaa-server** **Example:** `hostname(config)# aaa-server AuthOutbound protocol tacacs+`	Identifies your AAA servers. If you have already identified them, continue to the next step. For more information about identifying AAA servers, see the "Configuring AAA Server Groups" section on page 35-11.
Step 2	**access-list** **Example:** `hostname(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp`	Creates an access list that identifies the source addresses and destination addresses of traffic you want to authenticate. For details, see Chapter 15, "Adding an Extended Access List." 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 access list, 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:** `hostname(config)# aaa authentication match MAIL_AUTH inside AuthOutbound`	Configures authentication. The *acl_name* argument is the name of the access list 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. **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:** `hostname(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.

	Command	Purpose
Step 5	**aaa local authentication attempts max-fail** *number* **Example:** `hostname(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:** `hostname(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet`	Create an access list that identifies the source addresses and destination addresses of traffic that you want to authorize. For instructions, see Chapter 15, "Adding an Extended Access List." The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization. The access list that you use for authorization matching should include rules that are equal to or a subset of the rules in the access list used for authentication matching. **Note** If you have configured authentication and want to authorize all the traffic being authenticated, you can use the same access list that you created for use with the **aaa authentication match** command.
Step 7	**aaa authorization match** *acl_name interface_name server_group* **Example:** `hostname(config)# aaa authentication match TELNET_AUTH inside AuthOutbound`	Enables authorization. The *acl_name* argument is the name of the access list you created in Step 6, the *interface_name* argument is the name of the 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.

Examples

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.

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

```
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname(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" section on page 38-4.

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 access list information received from RADIUS servers.

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

Note If you have used the **access-group** command to apply access lists to interfaces, be aware of the following effects of the **per-user-override** keyword on authorization by user-specific access lists:

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

For more information, see the **access-group** command entry in the command reference.

This section includes the following topics:

- Configuring a RADIUS Server to Send Downloadable Access Control Lists, page 38-14
- Configuring a RADIUS Server to Download Per-User Access Control List Names, page 38-18

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 Access List Feature and Cisco Secure ACS, page 38-14
- Configuring Cisco Secure ACS for Downloadable Access Lists, page 38-16
- Configuring Any RADIUS Server for Downloadable Access Lists, page 38-17
- Converting Wildcard Netmask Expressions in Downloadable Access Lists, page 38-18

About the Downloadable Access List Feature and Cisco Secure ACS

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

- Unlimited access list size—Downloadable access lists are sent using as many RADIUS packets as required to transport the full access list from Cisco Secure ACS to the ASA.

- Simplified and centralized management of access lists—Downloadable access lists enable you to write a set of access lists 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 access list 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 access lists of any size.

The ASA receives downloadable access lists 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 access list. The Cisco IOS cisco-av-pair RADIUS VSA (vendor 9, attribute 1) includes the following attribute-value pair to identify the downloadable access list set:

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

   where *acl-set-name* is the internal name of the downloadable access list, which is a combination of the name assigned to the access list by the Cisco Secure ACS administrator and the date and time that the access list was last modified.
3. The ASA examines the name of the downloadable access list and determines if it has previously received the named downloadable access list.
   - If the ASA has previously received the named downloadable access list, communication with Cisco Secure ACS is complete and the ASA applies the access list to the user session. Because the name of the downloadable access list includes the date and time that it was last modified, matching the name sent by Cisco Secure ACS to the name of an access list previously downloaded means that the ASA has the most recent version of the downloadable access list.
   - If the ASA has not previously received the named downloadable access list, it may have an out-of-date version of the access list or it may not have downloaded any version of the access list. In either case, the ASA issues a RADIUS authentication request using the downloadable access list 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 access list, 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 access list 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 access list required is less than approximately 4 KB in length, Cisco Secure ACS responds with an access-accept message that includes the access list. The largest access list 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 access list in a cisco-av-pair RADIUS VSA. The access list 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
```

The following example is of an attribute-value pair:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

6. If the access list required is more than approximately 4 KB in length, Cisco Secure ACS responds with an access-challenge message that includes a portion of the access list, 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 access list received and responds with another access-request message that includes the same attributes as the first request for the downloadable access list, 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 access list in an access-accept message.

Configuring Cisco Secure ACS for Downloadable Access Lists

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

The access list 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 access list 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 access lists and associating them with users, see the user guide for your version of Cisco Secure ACS.

On the ASA, the downloaded access list 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 access list 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 Access Lists

You can configure any RADIUS server that supports Cisco IOS RADIUS VSAs to send user-specific access lists 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 access list 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 access lists that are sent in the cisco-av-pair attribute, see the documentation for your RADIUS server.

On the ASA, the downloaded access list name has the following format:

AAA-user-*username*

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

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

Converting Wildcard Netmask Expressions in Downloadable Access Lists

If a RADIUS server provides downloadable access lists 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 access lists on your RADIUS servers. Translation of wildcard netmask expressions means that downloadable access lists written for Cisco VPN 3000 series concentrators can be used by the ASA without altering the configuration of the downloadable access lists on the RADIUS server.

You configure access list netmask conversion on a per-server basis using the **acl-netmask-convert** command, available in the aaa-server configuration mode. For more information about configuring a RADIUS server, see the “Configuring AAA Server Groups” section on page 35-11. For more information about the **acl-netmask-convert** command, see the command reference.

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

To download a name for an access list 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.

To create an access list on the ASA, see Chapter 15, “Adding an Extended Access List.”

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.

To configure accounting, perform the following steps:

	Command	Purpose
Step 1	**access-list** **Example:** `hostname(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" section on page 38-4. If you want the ASA to provide accounting data per IP address, enabling authentication is not necessary. Creates an access list that identifies the source addresses and destination addresses of traffic for which you want accounting data. For instructions, see Chapter 15, "Adding an Extended Access List." 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 access list that you created for use with the **aaa authentication match** command.
Step 2	**aaa accounting match** *acl_name interface_name server_group* **Example:** `hostname(config)# aaa accounting match SERVER_AUTH inside AuthOutbound`	Enables accounting. The *acl_name* argument is the access list 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.

Examples

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.

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

To use MAC addresses to exempt traffic from authentication and authorization, perform the following steps:

	Command	Purpose
Step 1	**mac-list** *id* {**deny** | **permit**} *mac macmask* **Example:** `hostname(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 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. 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:** `hostname(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. You can only enter one instance of the **aaa mac-exempt match** command.

Examples

The following example bypasses authentication for a single MAC address:

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

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

```
hostname(config)# mac-list acd permit 0003.E300.0000 FFFF.FF00.0000
hostname(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.

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

Feature History for AAA Rules

Table 38-1 lists each feature change and the platform release in which it was implemented.

Table 38-1 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.**