Applying AAA 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 “AAA for System Administrators” section on page 33-5.

This chapter contains the following sections:

AAA Performance

The security appliance 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 security appliance cut-through proxy challenges a user initially at the application layer and then authenticates against standard RADIUS, TACACS+, or the local database. After the security appliance 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.

Configuring Authentication for Network Access

This section includes the following topics:

Authentication Overview

The security appliance lets you configure network access authentication using AAA servers.

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 *Cisco Security Appliance Command Reference* for timeout values.) For example, if you configure the security appliance 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.

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

If you do not want to allow HTTP(S), Telnet, or FTP through the security appliance but want to authenticate other types of traffic, you can configure virtual Telnet. With virtual Telnet, the user Telnets to a given IP address configured on the security appliance and the security appliance provides a Telnet prompt. For more information about the **virtual telnet** command, see the *Cisco Security Appliance Command Reference*.

For Telnet, HTTP(S), and FTP, the security appliance generates an authentication prompt. If the destination server also has its own authentication, the user enters another username and password.

For HTTP authentication, the security appliance checks local ports when static NAT is configured. If it detects traffic destined for local port 80, regardless of the global port, the security appliance intercepts the HTTP connection and enforces authentication.

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

```
static (inside,outside) tcp 10.48.66.155 889 192.168.123.10 www netmask 255.255.255.255
```

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

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

```
static (inside,outside) tcp 10.48.66.155 889 192.168.123.10 111 netmask 255.255.255.255
```

then users do not see the authentication page. Instead, the security appliance sends to the web browser an error message indicating that the user must be authenticated prior using the requested service.

Note If you use HTTP authentication without using the **aaa authentication secure-http-client** command, the username and password are sent in clear text to the destination web server, and not just to the AAA server. For example, if you authenticate inside users when they access outside web servers, anyone on the outside can learn valid usernames and passwords. We recommend that you use the **aaa authentication secure-http-client** command whenever you enable HTTP authentication. For more information about the **aaa authentication secure-http-client** command, see the "Enabling Secure Authentication of Web Clients" section on page 16-4.

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

```
name> jamiec@jchrichton
```

```
password> letmein@he110
```

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

Enabling Network Access Authentication

To enable network access authentication, perform the following steps:

Step 1 Using the **aaa-server** command, identify your AAA servers. If you have already identified your AAA servers, continue to the next step.

For more information about identifying AAA servers, see the "Identifying AAA Server Groups and Servers" section on page 10-14.

Step 2 Using the **access-list** command, create an ACL that identifies the source addresses and destination addresses of traffic you want to authenticate. For steps, see the "Adding an Extended Access List" section on page 13-5.

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, Telnet, or FTP in the ACL because the user must authenticate with one of these services before other services are allowed through the security appliance.

Step 3 To configure authentication, enter the following command:

```
hostname/contexta(config)# aaa authentication match acl_name interface_name server_group
```

where *acl_name* is the name of the ACL you created in Step 2, *interface_name* is the name of the interface as specified with the **nameif** command, and *server_group* is the AAA server group 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 *Cisco Security Appliance Command Reference* for more information.

Step 4 (Optional) If you are using the local database for network access authentication and you want to limit the number of consecutive failed login attempts that the security appliance allows any given user account, use the **aaa local authentication attempts max-fail** command. For example:

```
hostname/contexta(config)# aaa local authentication attempts max-fail 7
```

Tip To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.

For example, the following commands authenticate all inside HTTP traffic and SMTP traffic:

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

```
hostname/contexta(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
```

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

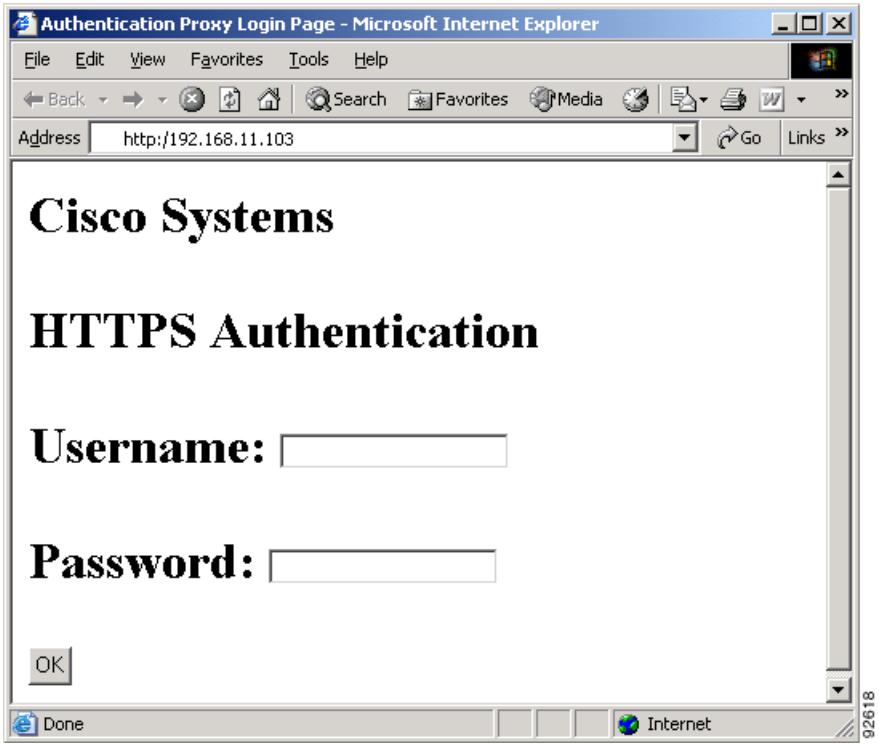
```
hostname/contexta(config)# aaa-server AuthInbound protocol tacacs+
hostname/contexta(config-aaa-server-group)# exit
hostname/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.1
hostname/contexta(config-aaa-server-host)# key TACPlusUauthKey
hostname/contexta(config-aaa-server-host)# exit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

Enabling Secure Authentication of Web Clients

The security appliance provides a method of securing HTTP authentication. Without securing HTTP authentication, usernames and passwords provided to the security appliance would be passed to the destination web server. By using the **aaa authentication secure-http-client** command, you enable the exchange of usernames and passwords between a web client and the security appliance with HTTPS. HTTPS encrypts the transmission, preventing the username and password from being passed to the external web server by HTTP.

After enabling this feature, when a user accesses a web page requiring authentication, the security appliance displays the Authentication Proxy Login Page shown in Figure 16-1.

Figure 16-1 Authentication Proxy Login Page



Note The Cisco Systems text field shown in this example was customized using the **auth-prompt** command. For the detailed syntax of this command refer to the *Cisco Security Appliance Command Reference*. If you do not enter a string using the **auth-prompt** command, this field will be blank.

After the user enters a valid username and password, an "Authentication Successful" page appears and closes automatically. If the user fails to enter a valid username and password, an "Authentication Failed" page appears.

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 non-authenticated 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 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 line configures static PAT for web traffic and the second line must be added to support the HTTPS authentication configuration.

```
static (inside,outside) tcp 10.132.16.200 www 10.130.16.10 www
static (inside,outside) tcp 10.132.16.200 443 10.130.16.10 443
```

- HTTP users see a pop-up window generated by the browser itself if **aaa authentication secure-http-client** is not configured. If **aaa authentication secure-http-client** is configured, a form loads in the browser to collect username and password. In either case, if a user enters an incorrect password, the user is prompted again. When the web server and the authentication server are on different hosts, use the **virtual** command to get the correct authentication behavior.

To enable secure authentication of web clients, perform the following steps:

Step 1 Enable HTTP authentication. For more information about enabling authentication, see the "Enabling Network Access Authentication" section on page 16-3.

Step 2 To enable secure authentication of web clients, enter this command:

```
aaa authentication secure-http-client
```

Note Use of the **aaa authentication secure-http-client** command is not dependent upon enabling HTTP authentication. If you prefer, you can enter this command before you enable HTTP authentication so that if you later enable HTTP authentication, usernames and passwords are already protected by secured web-client authentication.

Configuring Authorization for Network Access

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

This section includes the following topics:

Configuring TACACS+ Authorization

You can configure the security appliance to perform network access authorization with TACACS+. 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.

Tip Using ACLs to identify traffic to be authorized can greatly reduced the number of authorization commands you must enter. This is because each authorization rule 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 statement will be denied. For authorization to succeed, a user must first authenticate with the security appliance. Because a user at a given IP address only needs to authenticate one time for all rules and types, if the authentication session hasn't expired, authorization can occur even if the traffic is matched by an authentication statement.

After a user authenticates, the security appliance checks the authorization rules for matching traffic. If the traffic matches the authorization statement, the security appliance sends the username to the TACACS+ server. The TACACS+ server responds to the security appliance with a permit or a deny for that traffic, based on the user profile. The security appliance 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:

Step 1 Enable authentication. For more information, see the "Enabling Network Access Authentication" section on page 16-3. If you have already enabled authentication, continue to the next step.

Step 2 Using the **access-list** command, create an ACL that identifies the source addresses and destination addresses of traffic you want to authorize. For steps, see the "Adding an Extended Access List" section on page 13-5.

The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization. The ACL you use for authorization matching should contain 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 you created for use with the **aaa authentication match** command.

Step 3 To enable authorization, enter the following command:

```
hostname/contexta(config)# aaa authorization match acl_name interface_name server_group
```

where *acl_name* is the name of the ACL you created in Step 2, *interface_name* is the name of the interface as specified with the **nameif** command or by default, and *server_group* is the AAA server group 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 *Cisco Security Appliance Command Reference* for more information.

The following commands authenticate and authorize 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/contexta(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
hostname/contexta(config)# access-list SERVER_AUTH extended permit tcp any host
209.165.201.5 eq telnet
hostname/contexta(config)# aaa-server AuthOutbound protocol tacacs+
hostname/contexta(config-aaa-server-group)# exit
hostname/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname/contexta(config-aaa-server-host)# key TACPlusUauthKey
hostname/contexta(config-aaa-server-host)# exit
hostname/contexta(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname/contexta(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 Authentication for Network Access" section on page 16-1.

When you configure the security appliance 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 security appliance. It does provide information about how the security appliance handles ACL information received from RADIUS servers.

You can configure a RADIUS server to download an ACL to the security appliance 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 used the **access-group** command to apply ACLs to interfaces, be aware of the following effects of the **per-user-override** keyword on authorization by user-specific ACLs:

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

For more information, see the **access-group** command entry in the *Cisco Security Appliance Command Reference*.

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 security appliance.
- 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 security appliances.

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 security appliance receives downloadable ACLs from Cisco Secure ACS using the following process:

1. The security appliance 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 contains the internal name of the applicable downloadable ACL. The Cisco IOS cisco-av-pair RADIUS VSA (vendor 9, attribute 1) contains 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 security appliance examines the name of the downloadable ACL and determines if it has previously received the named downloadable ACL.
   - If the security appliance has previously received the named downloadable ACL, communication with Cisco Secure ACS is complete and the security appliance applies the ACL to the user session. Because the name of the downloadable ACL includes the date and time it was last modified, matching the name sent by Cisco Secure ACS to the name of an ACL previous downloaded means that the security appliance has the most recent version of the downloadable ACL.

- If the security appliance 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 security appliance 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 security appliance signs the request with the Message-Authenticator attribute (IETF RADIUS attribute 80).

4. Upon receipt of a RADIUS authentication request that has a username attribute containing 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 containing the ACL. The largest ACL that can fit in a single access-accept message is slightly less than 4 KB because some 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 contain an ACE and are numbered serially:

   ```
   ip:inacl#1=ACE-1
   ip:inacl#2=ACE-2
   .
   .
   .
   ip:inacl#n=ACE-n
   ```

   An example of an attribute-value pair follows:

   ```
   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 contains a portion of the ACL, formatted as described above, and an State attribute (IETF RADIUS attribute 24), which contains 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 security appliance stores the portion of the ACL received and responds with another access-request message containing 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 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 security appliance commands that are similar to the extended **access-list** command (see the "Adding an Extended Access List" section on page 13-5), 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 security appliance, 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 security appliance 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 security appliance 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 the "Adding an Extended Access List" section on page 13-5), 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 security appliance. 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 security appliance, 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 security appliance 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 security appliance 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 security appliance, you may need the security appliance to convert wildcard netmask expressions to standard netmask expressions. This is because Cisco VPN 3000 Series Concentrators support wildcard netmask expressions but the security appliance only supports standard netmask expressions. Configuring the security appliance to convert wildcard netmask expressions helps minimize the effects of these differences upon 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 security appliance without altering the configuration of the downloadable ACLs on the RADIUS server.

You configure ACL 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 “Identifying AAA Server Groups and Servers” section on page 10-14. For more information about the **acl-netmask-convert** command, see the *Cisco Security Appliance Command Reference*.

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 security appliance 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 value for filter-id attributes are specified in boxes in the HTML interface, omitting **filter-id=** and entering only *acl_name*.

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

See the "Adding an Extended Access List" section on page 13-5 to create an ACL on the security appliance.

Configuring Accounting for Network Access

The security appliance can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the security appliance. 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 when sessions start and stop, username, the number of bytes that pass through the security appliance for the session, the service used, and the duration of each session.

To configure accounting, perform the following steps:

Step 1 If you want the security appliance to provide accounting data per user, you must enable authentication. For more information, see the "Enabling Network Access Authentication" section on page 16-3. If you want the security appliance to provide accounting data per IP address, enabling authentication is not necessary and you can continue to the next step.

Step 2 Using the **access-list** command, create an ACL that identifies the source addresses and destination addresses of traffic you want accounted. For steps, see the "Adding an Extended Access List" section on page 13-5.

The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization.

Note If you have configured authentication and want accounting data for all the traffic being authenticated, you can use the same ACL you created for use with the **aaa authentication match** command.

Step 3 To enable accounting, enter the following command:

```
hostname/contexta(config)# aaa accounting match acl_name interface_name server_group
```

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 *Cisco Security Appliance Command Reference* for more information.

The following commands authenticate, authorize, and account 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/contexta(config)# aaa-server AuthOutbound protocol tacacs+
hostname/contexta(config-aaa-server-group)# exit
hostname/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname/contexta(config-aaa-server-host)# key TACPlusUauthKey
hostname/contexta(config-aaa-server-host)# exit
hostname/contexta(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
hostname/contexta(config)# access-list SERVER_AUTH extended permit tcp any host
209.165.201.5 eq telnet
hostname/contexta(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname/contexta(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
hostname/contexta(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```

Using MAC Addresses to Exempt Traffic from Authentication and Authorization

The security appliance can exempt from authentication and authorization any traffic from specific MAC addresses.

For example, if the security appliance authenticates TCP traffic originating on a particular network but you want to allow unauthenticated TCP connections from a specific server, you would use the **mac-list** command to create a rule permitting traffic from the MAC address of the server and then use the **aaa mac-exempt** command to exempt from authentication and authorization any traffic from the server specified by the MAC list.

Conversely, if traffic from a particular computer should never be permitted regardless of authentication, you can use the MAC address of the computer in a **mac-list** command that denies traffic from the MAC address. The use of the **aaa mac-exempt** command in this scenario would disallow traffic from the computer even though authentication rules would otherwise permit the traffic.

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

Step 1 To configure a MAC list, enter the following command:

```
hostname/contexta(config)# mac-list id {deny | permit} mac macmask
```

where *id* is the hexadecimal number that you assign to the MAC list, *mac* is the MAC address of the computer whose traffic you want to permit or deny, and *macmask* is a MAC address mask. For more information about the **mac-list** command, see the *Cisco Security Appliance Command Reference*.

Step 2 To exempt traffic for the MAC addresses specified in a particular MAC list, enter the following command:

```
hostname/contexta(config)# aaa mac-exempt match id
```

where *id* is the string identifying the MAC list containing the MAC addresses whose traffic is to be exempt from authentication and authorization.

The following commands create two MAC lists, each consisting of a single MAC address. One permits traffic from its MAC address while the other denies traffic from its MAC address. The final two commands configure the security appliance to exempt from authentication and authorization any traffic originating from the MAC addresses in the two lists.

```
hostname/contexta(config)# mac-list adc permit 00a0.cp5d.0282 ffff.ffff.ffff
hostname/contexta(config)# mac-list ac deny 0061.54ff.b440 ffff.ffff.ffff
hostname/contexta(config)# aaa mac-exempt match adc
hostname/contexta(config)# aaa mac-exempt match ac
```