Configuring AAA Authorization and Authentication Cache

The AAA Authorization and Authentication Cache feature allows you to cache authorization and authentication responses for a configured set of users or service profiles, providing performance improvements and an additional level of network reliability. Users and service profiles that are returned from authorization and authentication responses can be queried from multiple sources and need not depend solely on an offload server. This feature also provides a failover mechanism so that if a network RADIUS or TACACS+ server is unable to provide authorization and authentication responses network users and administrators can still access the network.

Prerequisites for Implementing Authorization and Authentication Profile Caching

The following prerequisites apply to implementing authorization and authentication profile caching:

- Understand how you want to implement profile caching, that is, are profiles being cached to improve network performance or as a failover mechanism if your network authentication and authorization servers (RADIUS and TACACS+) become unavailable.
- RADIUS and TACACS+ server groups must already be configured.

Information About Implementing Authorization and Authentication Profile Caching

The following sections provide information about implementing authorization and authentication profile caching.

Network Performance Optimization Using Authorization and Authentication Profile Caching

RADIUS and TACACS+ clients run on Cisco devices and send authentication requests to a central RADIUS or TACACS+ server that contains all user authentication and network service access information. The device is required to communicate with an offload RADIUS or TACACS+ server to authenticate a given call and then apply a policy or service to that call. Unlike authentication, authorization, and accounting (AAA) accounting, AAA authentication and authorization is a blocking procedure, which means the call setup may not proceed while the call is being authenticated and authorized. Thus, the time required to process the call setup is directly impacted by the time required to process such an authentication or authorization request from the device to the offload RADIUS or TACACS+ server, and back again. Any communication problems in the transmission, offload server utilization, and numerous other factors cause significant degradation in a device's call setup performance because of the AAA authentication and authorization step. The problem is further highlighted when multiple AAA authentications and authorizations are needed for a single call or session.

A solution to this problem is to minimize the impact of such authentication requests by caching the authentication and authorization responses for specific users on the device, thereby removing the need to send the requests to an offload server again and again. This profile caching adds significant performance improvements to the call setup times. Profile caching also provides an additional level of network reliability because user and service profiles that are returned from authentication and authorization responses can be queried from multiple sources and need not depend solely on an offload server.

To take advantage of this performance optimization, you need to configure the authentication method list so that the AAA cache profile is queried first when a user attempts to authenticate to the device. See Method Lists in Authorization and Authentication Profile Caching section for more information.

Authorization and Authentication Profile Caching as a Failover Mechanism

If, for whatever reason, RADIUS or TACACS+ servers are unable to provide authentication and authorization responses, network users and administrators can be locked out of the network. The profile caching feature allows usernames to be authorized without having to complete the authentication phase. For example, a user by the name *user100@example.com* with the password *secretpassword1* can be stored in a profile cache using the regular expression .*@example.com. Another user by the name *user101@example.com* with the password *secretpassword2* can also be stored using the same regular expression, and so on. Because the number of users in the .*@example.com profile could run into thousands, it is not feasible to authenticate each user with their personal password. Therefore, authentication is disabled, and each user simply accesses authorization profiles from a common Access Response stored in the cache.

The same reasoning applies in cases where higher-end security mechanisms such as Challenge Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), or Extensible Authentication Protocol (EAP), which use an encrypted password between a client and AAA offload server. To allow these unique secure username and password profiles to retrieve their authorization profiles, authentication is bypassed.

To take advantage of this failover capability, you need to configure the authentication and authorization method list so that the cache server group is queried last when a user attempts to authenticate to the device. See Method Lists in Authorization and Authentication Profile Caching section for more information.

Method Lists in Authorization and Authentication Profile Caching

A method list is a sequential list describing the authentication methods to be queried in order to authenticate a user. Cisco support methods such as local (use the local Cisco IOS XE database), none (do nothing), RADIUS server group, or TACACS+ server group. Typically, more than one method can be configured into a method list. Cisco IOS XE software uses the first listed method to authenticate users. If that method fails to respond, the Cisco IOS XE software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or until all the methods defined in the method list are exhausted.

To optimize network performance or provide failover capability using the profile caching feature, change the order of the authentication and authorization methods in the method list. To optimize network performance, make sure the cache server group appears first in the method list. For failover capability, the cache server group should appear last in the method list.

Authorization and Authentication Profile Caching Guidelines

Because the number of usernames and profiles that can request to be authenticated or authorized at a given device on a given point of presence (POP) can be quite extensive, it is not feasible to cache all of them. Therefore, only usernames and profiles that are commonly used or that share a common authentication and authorization response should be configured to use caching. Commonly used usernames such as aolip and aolnet, which are used for America Online (AOL) calls, or preauthentication dialed number identification service (DNIS) numbers used to connect Public Switched Telephone Network (PSTN) calls to a network-attached storage device, along with domain-based service profiles, are all examples of usernames and profiles that can benefit from authentication and authorization caching.

General Configuration Procedure for Implementing Authorization and Authentication Profile Caching

To implement authorization and authentication profile caching, complete the following procedure:

1. Create cache profile groups and define the rules for what information is cached in each group.

   Entries that match based on exact username and regular expressions, or specify all authentication and authorization requests, can be cached.

2. Update existing server groups to reference newly defined cache groups.
3. Update authentication or authorization method lists to use the cached information to optimize network performance or provide a failover mechanism.

How to Implement Authorization and Authentication Profile Caching

The following sections provide information about the various tasks that comprise authorization and authentication profile-caching configuration.

Creating Cache Profile Groups and Defining Caching Rules

Perform this task to create a cache profile group, define the rules for what information is cached in that group, and verify and manage cache profile entries.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa cache profile** *group-name* **Example:** `Device(config)# aaa cache profile networkusers@companyname`	Defines an authentication and authorization cache profile server group and enters profile map configuration mode.
Step 5	**profile** *name* [**no-auth**] **Example:** `Device(config-profile-map)# profile networkuser1 no-auth`	Creates an individual authentication and authorization cache profile based on a username match. • The *name* argument must be an exact match to a username being queried by an authentication or authorization service request. • Use the **no-auth** keyword to bypass authentication for this user.

	Command or Action	Purpose
		Note • For EAP-PEAP with MSCHAPv2 and EAP-PEAP with GTC methods, AAA authentication caching for 802.1x is supported with or without bypass authentication. However, for EAP methods such as EAP-TLS and EAP-MD5, AAA authentication caching for 802.1x is only supported with bypass authentication. • For EAP-MSCHAPV2 use cases that do not use **no-auth** (bypass authentication), the administrator must configure the Cisco AV-pairs AS-username and AS-passwordHash on the Cisco Identity Services Engine (ISE), such that Cisco ISE sends these RADIUS attributes through the RADIUS `ACCESS-Accept` message to the network access server (NAS) device. Also, AS-passwordHash must be configured with nt-hash of the user password. Repeat this step for each username that you want to add to the profile group in Step 4.
Step 6	**regexp** *matchexpression* {**any** | **only**} [**no-auth**] **Example:**	(Optional) Creates an entry in a cache profile group that matches a regular expression. • If you use the **any** keyword, all the unique usernames matching the regular expression are saved.

	Command or Action	Purpose
	`Device(config-profile-map)# regexp .*@example.com any no-auth`	• If you use the **only** keyword, only one profile entry is cached for all the usernames matching the regular expression. • Use the **no-auth** keyword to bypass authentication for a user or set of users. • Because the number of entries in a regular expression cache profile group could run into thousands, and validating each request against a regular expression can be time consuming, we recommend that you do not use regular expression entries in cache profile groups. Repeat this step for each regular expression that you want to add to the cache profile group defined in Step 4.
Step 7	**all** [**no-auth**] **Example:** `Device(config-profile-map)# all no-auth`	(Optional) Specifies that all authentication and authorization requests are cached. • Use the **all** keyword for specific service authorization requests, but avoid it when dealing with authentication requests.
Step 8	**end** **Example:** `Device(config-profile-map)# end`	Exits profile map configuration mode and returns to privileged EXEC mode.
Step 9	**show aaa cache group** *name* {**profile** *name* | **all**} **Example:** `Device# show aaa cache group networkusers@companyname all`	(Optional) Displays an individual server group profile details or all the server group profile details.
Step 10	**clear aaa cache group** *name* {**profile** *name* | **all**} **Example:** `Device# clear aaa cache group networkusers@companyname profile networkuser1`	(Optional) Clears an individual entry or all the entries in the cache.
Step 11	**debug aaa cache group** **Example:**	(Optional) Displays debug information about cached entries.

	Command or Action	Purpose
	`Device# debug aaa cache group`	

Defining RADIUS and TACACS Server Groups that Use Cache Profile Group Information

Perform this task to define how RADIUS and TACACS+ server groups use the information stored in each cache profile group.

Before you begin

RADIUS and TACACS+ server groups must be created.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa group server radius** *group-name* or **aaa group server tacacs+** *group-name* **Example:** `Device(config)# aaa group server radius networkusers@companyname`	Enters RADIUS server group configuration mode. To enter TACACS+ server group configuration mode, use the **aaa group server tacacs+** *group-name* command.
Step 5	**cache authorization profile** *name* **Example:** `Device(config-sg-radius)# cache authorization profile networkusers@companyname`	Activates the authorization caching rules in the networkusers profile for this RADIUS or TACACS+ server group. The *name* argument is a AAA cache profile group name.

	Command or Action	Purpose
Step 6	**cache authentication profile** *name* **Example:** `Device(config-sg-radius)# cache authentication profile networkusers@companyname`	Activates the authentication-caching rules in the networkusers profile for this RADIUS or TACACS+ server group.
Step 7	**cache expiry** *hours* {**enforce** | **failover**} **Example:** `Device(config-sg-radius)# cache expiry 240 failover`	(Optional) Sets the amount of time before a cache profile entry expires (becomes stale). Use the **enforce** keyword to specify that after a cache profile entry expires, it is not used again. Use the **failover** keyword to specify that an expired cache profile entry can be used if all other methods to authenticate and authorize the user fails.
Step 8	**end** **Example:** `Device(config-sg-radius)# end`	Returns to privileged EXEC mode.

Updating Authorization and Authentication Method Lists to Specify How Cache Information is Used

Perform this task to update authorization and authentication method lists to use the authorization and authentication cache information.

Before you begin

Method lists must already be defined.

Procedure

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

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]] **Example:** `Device(config)# aaa authorization network default cache networkusers@companyname group networkusers@companyname`	Enables AAA authorization and creates method lists, which define the authorization methods used when a user accesses a specified function.
Step 5	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Device(config)# aaa authentication ppp default cache networkusers@companyname group networkusers@companyname`	Specifies one or more authentication methods for use on serial interfaces that are running PPP.
Step 6	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Device(config)# aaa authentication login default cache adminusers group adminusers`	Sets the authentication at login.
Step 7	**aaa authentication dot1x** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Device(config)# aaa authentication dot1x default cache RADGRP group RADGRP`	Sets the authentication for use on interfaces running IEEE 802.1x.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for Implementing Authorization and Authentication Profile Caching

This following sections display configuration examples for implementing authorization and authentication profile caching.

Example: Implementing Authorization and Authentication Profile Caching for Network Optimization

The following configuration example shows how to:

- Define a cache profile group admin_users that contains the names of all the administrators on the network and sets this list as the default list that is used for all login and privileged exec sessions.
- Activate the new caching rules for a RADIUS server group.
- Add the new cache profile group in the authentication and authorization method list and change the method order so that the cache profile group is queried first.

```
configure terminal
 aaa new-model
 ! Define aaa cache profile groups and the rules for what information is saved to cache.
 aaa cache profile admin_users
 profile adminuser1
 profile adminuser2
 profile adminuser3
 profile adminuser4
 profile adminuser5
 exit
 ! Define server groups that use the cache information in each profile group.
 aaa group server radius admins@companyname.com
 cache authorization profile admin_users
 cache authentication profile admin_users
 ! Update authentication and authorization method lists to specify how profile groups and
server groups are used.
 aaa authentication login default cache admins@companyname.com group admins@companyname.com

 aaa authorization exec default cache admins@companyname.com group admins@companyname.com
 end
```

Example: Implementing Authorization and Authentication Profile Caching as a Failover Mechanism

The following configuration example shows how to:

- Create a cache profile group admin_users that contains all the administrators on the network so that if the RADIUS or TACACS+ server should become unavailable the administrators can still access the network.

- Create a cache profile group abc_users that contains all the *ABC* company users on the network so that if the RADIUS or TACACS+ server should become unavailable, these users will be authorized to use the network.
- Activate the new caching rules for each profile group on a RADIUS server.
- Add the new cache profile group in the authentication and authorization method list and change the method order so that the cache profile group is queried last.

```
configure terminal
 aaa new-model
 ! Define aaa cache profile groups and the rules for what information is saved to cache.
 aaa cache profile admin_users
 profile admin1
 profile admin2
 profile admin3
 exit
 aaa cache profile abcusers
 profile .*@example.com only no-auth
 exit
 ! Define server groups that use the cache information in each cache profile group.
 aaa group server tacacs+ admins@companyname.com
 server 10.1.1.1
 server 10.20.1.1
 cache authentication profile admin_users
 cache authorization profile admin_users
 exit
 aaa group server radius abcusers@example.com
 server 172.16.1.1
 server 172.20.1.1
 cache authentication profile abcusers
 cache authorization profile abcusers
 exit
 ! Update authentication and authorization method lists to specify how cache is used.
 aaa authentication login default cache admins@companyname.com group admins@companyname.com

 aaa authorization exec default cache admins@companyname.com group admins@companyname.com
 aaa authentication ppp default group abcusers@example.com cache abcusers@example.com
 aaa authorization network default group abcusers@example.com cache abcusers@example.com
 end
```

Feature History for Implementing Authorization and Authentication Profile Caching

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	AAA Authorization and Authentication Cache	This feature optimizes network performance and provides a failover mechanism in the event a network RADIUS or TACACS+ server becomes unavailable for any reason.

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.7.1	AAA Cache for 802.1x	AAA authentication caching support for 802.1x has been introduced.
Cisco IOS XE Cupertino 17.7.1	AAA Authorization and Authentication Cache	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.