Configuring Authentication, Authorization, and Accounting

This chapter contains the following sections:

Information About AAA

AAA Security Services

The authentication, authorization, and accounting (AAA) features allows you to verify the identity of, grant access to, and track the actions of users who manage Cisco Nexus devices. The Cisco Nexus device supports Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control device Plus (TACACS+) protocols.

Based on the user ID and password that you provide, the switches perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the switch and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

- Authentication—Identifies users, including login and password dialog, challenge and response, messaging support, and, encryption depending on the security protocol that you select.
- Authorization—Provides access control.

  Authorization to access a Cisco Nexus device is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

- Accounting—Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

Note The Cisco NX-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- User password lists for each switch in the fabric are easier to manage.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- The accounting log for all switches in the fabric can be centrally managed.
- User attributes for each switch in the fabric are easier to manage than using the local databases on the switches.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. A server group provides for failover servers if a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, that server group option is considered a failure. If required, you can specify multiple server groups. If a switch encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

On Cisco Nexus devices, you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication

- User management session accounting

The following table lists the CLI commands for each AAA service configuration option.

Table 1: AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
User session accounting	**aaa accounting default**

You can specify the following authentication methods for the AAA services:

- RADIUS server groups—Uses the global pool of RADIUS servers for authentication.
- Specified server groups—Uses specified RADIUS or TACACS+ server groups for authentication.
- Local—Uses the local username or password database for authentication.
- None—Uses only the username.

Note If the method is for all RADIUS servers, instead of a specific server group, the Cisco Nexus devices choose the RADIUS server from the global pool of configured RADIUS servers in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the Cisco Nexus devices.

The following table describes the AAA authentication methods that you can configure for the AAA services.

Table 2: AAA Authentication Methods for AAA Services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
User management session accounting	Server groups and local

Note For console login authentication, user login authentication, and user management session accounting, the Cisco Nexus devices try each option in the order specified. The local option is the default method when other configured options fail.

Authentication and Authorization Process for User Logins

The authentication and authorization process for user login is as occurs:

- When you log in to the required Cisco Nexus device, you can use the Telnet, SSH, Fabric Manager or Device Manager, or console login options.
- When you have configured the AAA server groups using the server group authentication method, the Cisco Nexus device sends an authentication request to the first AAA server in the group as follows:

  If the AAA server fails to respond, then the next AAA server is tried and so on until the remote server responds to the authentication request.

  If all AAA servers in the server group fail to respond, the servers in the next server group are tried.

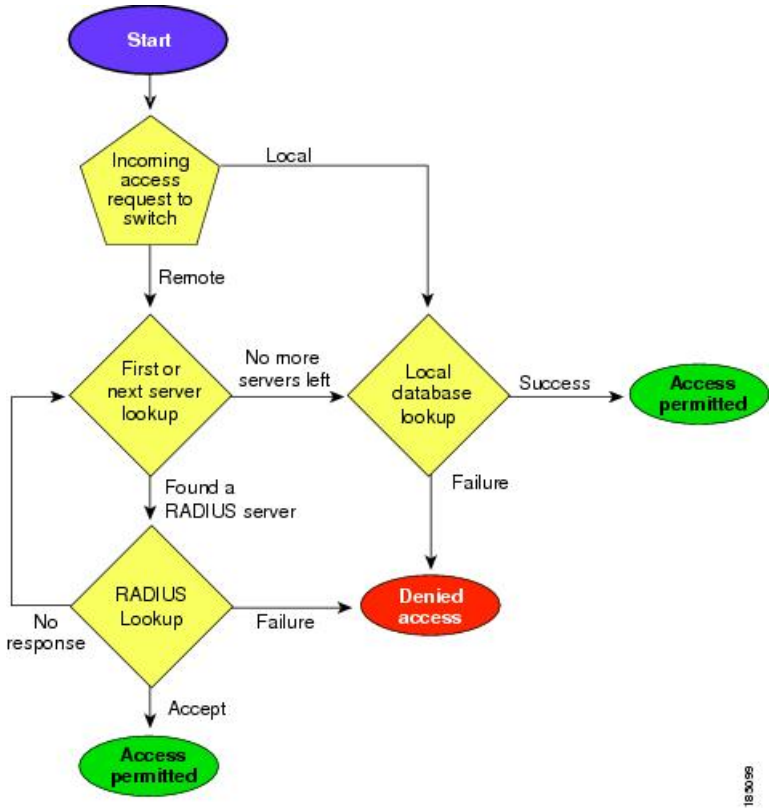
  If all configured methods fail, the local database is used for authentication.
- If a Cisco Nexus device successfully authenticates you through a remote AAA server, the following conditions apply:

  If the AAA server protocol is RADIUS, user roles specified in the cisco-av-pair attribute are downloaded with an authentication response.

  If the AAA server protocol is TACACS+, another request is sent to the same server to get the user roles specified as custom attributes for the shell.
- If your username and password are successfully authenticated locally, the Cisco Nexus device logs you in and assigns you the roles configured in the local database.

The following figure shows a flowchart of the authentication and authorization process.

Figure 1: Authentication and Authorization Flow for User Login



Note This figure is applicable only to username password SSH authentication. It does not apply to public key SSH authentication. All username password SSH authentication goes through AAA.

In the figure, "No more servers left" means that there is no response from any server within this server group.

Prerequisites for Remote AAA

Remote AAA servers have the following prerequisites:

- At least one RADIUS or TACACS+ server must be IP reachable.
- The Cisco Nexus device is configured as a client of the AAA servers.
- The preshared secret key is configured on the Cisco Nexus device and on the remote AAA servers.
- The remote server responds to AAA requests from the Cisco Nexus device.

Configuring AAA

Configuring Console Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS or TACACS+ servers
- Local database on the Cisco Nexus device.
- Username only **none**

The default method is local.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command are used for a set of previously defined RADIUS servers. Use the **radius server-host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Before you configure console login authentication methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default console login method is **local**, which is used when no methods are configured or when all of the configured methods fail to respond.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configuration of the console login authentication methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure authentication methods for the console login:

```
switch# configure terminal
switch(config)# aaa authentication login console group radius
switch(config)# exit
switch# show aaa authentication
switch# copy running-config startup-config
```

Configuring Default Login Authentication Methods

The default method is local.

Before you configure default login authentication methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**}	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default login method is **local** , which is used when no methods are configured or when all of the configured methods do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configuration of the default login authentication methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

When you log in, the login is processed by the local user database if the remote AAA servers do not respond. If you have enabled the displaying of login failure messages, the following message is displayed:

```
Remote AAA servers unreachable; local authentication done.
Remote AAA servers unreachable; local authentication failed.
```

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login error-enable**	Enables login authentication failure messages. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) switch# **show aaa authentication**	Displays the login failure message configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring AAA Command Authorization

When a TACACS+ server authorization method is configured, you can authorize every command that a user executes with the TACACS+ server which includes all EXEC mode commands and all configuration mode commands.

The authorization methods include the following:

- Group—TACACS+ server group
- Local—Local role-based authorization
- None—No authorization is performed

The default method is Local.

Note Authorization on the console session is not supported on the Cisco Nexus 5000 platform. It is supported on the Cisco Nexus 5500 platform, release 6.x onwards.

Before you begin

You must enable TACACS+ before configuring AAA command authorization.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization** {**commands** | **config-commands**} {**default**} {{[**group** *group-name*] | [**local**]} | {[**group** *group-name*] | [**none**]}} **Example:** `switch(config)# aaa authorization config-commands default group tac1` **Example:**	Configures authorization parameters. Use the **commands** keyword to authorize EXEC mode commandes. Use the **config-commands** keyword to authorize configuration mode commands. Use the **group**, **local**, or **none** keywords to identify the authorization method.

	Command or Action	Purpose
	`switch# aaa authorization commands default group tac1`	

Example

The following example shows how to authorize EXEC mode commands with TACACS+ server group *tac1*:

```
switch# aaa authorization commands default group tac1
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

```
switch(config)# aaa authorization config-commands default group tac1
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, the command is authorized based on the user's *local* role.

```
switch(config)# aaa authorization config-commands default group tac1 local
```

The followng example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, allow the command regardless of the local role.

```
switch# aaa authorization commands default group tac1 none
```

The following example shows how to authorize EXEC mode commands regardless of the local role:

```
switch# aaa authorization commands default none
```

The following example shows how to authorize EXEC mode commands using the local role for authorization:

```
switch# aaa authorization commands default local
```

Configuring Console Authorization Commands

The authorization methods include the following:

- Named subset of TACACS+ servers
- Local database on the Cisco Nexus device.

- Username only **none**

The default method is local.

Before you configure console authorization commands, configure TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authorization commands console** {**group** *group-list* [**none**] | **local** | **none**}	Configures authorization for the console. The *group-list* argument consists of a space-delimited list of group name. The group name is: • *named-group* —Uses a named subset of TACACS+ servers for authorization. The **local** method uses the local database for authorization. The **none** method uses the username only. The default console authorization is **local**, which is used when no methods are configured or when all of the configured methods fail to respond.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show aaa authorization**	Displays the configuration of the console authorization commands.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the console authorization commands:

```
switch# configure terminal
switch(config)# aaa authorization commands console group tacacs+
switch(config)# exit
switch# show aaa authorization
switch# copy running-config startup-config
```

Enabling MSCHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. You can use MSCHAP for user logins to a Cisco Nexus device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco Nexus device uses Password Authentication Protocol (PAP) authentication between the switch and the remote server. If you enable MSCHAP, you must configure your RADIUS server to recognize the MSCHAP vendor-specific attributes (VSAs).

The following table describes the RADIUS VSAs required for MSCHAP.

Table 3: MSCHAP RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by an AAA server to an MSCHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP user in response to the challenge. It is only used in Access-Request packets.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login mschap enable**	Enables MS-CHAP authentication. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication login mschap**	Displays the MS-CHAP configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Related Topics

VSAs, on page 12

Configuring AAA Accounting Default Methods

The Cisco Nexus device supports TACACS+ and RADIUS methods for accounting. The switches report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Cisco Nexus device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:.

- RADIUS server group—Uses the global pool of RADIUS servers for accounting.
- Specified server group—Uses a specified RADIUS or TACACS+ server group for accounting.
- Local—Uses the local username or password database for accounting.

Note If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before you begin

Before you configure AAA accounting default methods, configure RADIUS or TACACS+ server groups as needed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa accounting default** {**group** *group-list* | **local**}	Configures the default accounting method. One or more server group names can be specified in a space-separated list. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for accounting. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for accounting. The **local** method uses the local database for accounting. The default method is **local**, which is used when no server groups are configured or when all the configured server group do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa accounting**	Displays the configuration AAA accounting default methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Using AAA Server VSAs

VSAs

You can use vendor-specific attributes (VSAs) to specify the Cisco Nexus device user roles and SNMPv3 parameters on AAA servers.

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute seperator value *
```

The protocol is a Cisco attribute for a particular type of authorization, separator is an equal sign (=) for mandatory attributes, and an asterisk (*) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco Nexus device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

VSA Format

The following VSA protocol options are supported by the Cisco Nexus device:

- Shell— Used in access-accept packets to provide user profile information.
- Accounting—Used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the Cisco Nexus device:

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space.
- accountinginfo—Stores additional accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Switch User Roles and SNMPv3 Parameters on AAA Servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the Cisco Nexus device using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the cisco-av-pair attribute, the default user role is network-operator.

Note For information on Cisco Unified Wireless Network TACACS+ configurations and to change the user roles, see Cisco Unified Wireless Network TACACS+ Configuration.

You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the cisco-av-pair attribute, MD5 and DES are the default authentication protocols.

For additional information, see the Configuring User Accounts and RBAC chapter in the System Management Configuration Guide for your Cisco Nexus device.

Secure Login Enhancements

The following secure login enhancements are supported in Cisco NX-OS:

Configuring Login Parameters

Use this task to configure your Cisco NX-OS device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

All login parameters are disabled by default. You must enter the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following default is enforced:

- All login attempts made through Telnet or SSH are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is entered.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `Switch(config)# login block-for 100 attempts 2 within 100`	Configures your Cisco NX-OS device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.
Step 3	**[no] login quiet-mode access-class** {*acl-name* | *acl-number*} **Example:** `Switch(config)# login quiet-mode access-class myacl`	(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console.
Step 4	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show login failures** **Example:**	Displays login parameters. • **failures** --Displays information related only to failed login attempts.

	Command or Action	Purpose
	`Switch# show login`	

Configuration Examples for Login Parameters

Setting Login Parameters Example

The following example shows how to configure your switch to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests are denied during the quiet period except hosts from the ACL "myacl."

```
Switch(config)# login block-for 100 attempts 15 within 100
Switch(config)# login quiet-mode access-class myacl
```

Showing Login Parameters Example

The following sample output from the **show login** command verifies that no login parameters have been specified:

```
Switch# show login

No Quiet-Mode access list has been configured, default ACL will be applied.

Switch is enabled to watch for login Attacks.
If more than 2 login failures occur in 45 seconds or less,  logins will be disabled for 70
 seconds.

Switch presently in Normal-Mode.
Current Watch Window remaining time 10 seconds.
Present login failure count 0.
```

The following sample output from the **show login failures** command shows all failed login attempts on the switch:

```
Switch# show login failures

Information about last 20 login failures with the device.
-------------------------------------------------------------------------------
Username                                Line    Source                Appname
TimeStamp
-------------------------------------------------------------------------------
admin                                    pts/0   bgl-ads-728.cisco.com   login
        Wed Jun 10 04:56:16 2015
admin                                    pts/0   bgl-ads-728.cisco.com   login
        Wed Jun 10 04:56:19 2015
-------------------------------------------------------------------------------
```

The following sample output from the **show login failures** command verifies that no information is presently logged:

```
Switch# show login failures
*** No logged failed login attempts with the device.***
```

Configuring Login Block Per User

The Login Block Per User feature helps detect suspected Denial of Service (DoS) attacks and to slow down dictionary attacks. This feature is applicable only for local users. Use this task to configure login parameters to block an user after failed login attempts.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**aaa authentication rejected** *attempts* **in** *seconds* **ban** *seconds* **Example:** `switch(config)# aaa authentication rejected 3 in 20 ban 300`	Configures login parameters to block an user. **Note** Use the **no aaa authentication rejected** command to revert to the default login parameters.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	**show running config** **Example:** `switch# show running config`	(Optional) Displays the login parameters.
Step 5	**show aaa local user blocked** **Example:** `switch# show aaa local user blocked`	(Optional) Displays the blocked local users.
Step 6	**clear aaa local user blocked** {**username** *user* | **all**} **Example:** `switch# clear aaa local user blocked username testuser`	(Optional) Clears the blocked local users. • **all**–Clears all the blocked local users.

Configuration Examples for Login Block Per User

Setting Parameters for Login Block Per User

The following example shows how to configure the login parameters to block a user for 300 seconds when five login attempts fail within a period of 60 seconds:

```
switch(config)# aaa authentication rejected 5 in 60 ban 300
```

Showing Login Parameters

The following example shows the login parameters configured for a switch:

```
switch# show run | i rejected
aaa authentication rejected 5 in 60 ban 300
```

Showing Blocked Local Users

The following example shows the blocked local users:

```
switch# show aaa local user blocked
Local-user                  State

testuser                    Watched (till 11:34:42 IST Feb 5 2015)
```

Clearing Blocked Local Users

The following example shows how to clear the blocked local user testuser:

```
switch# clear aaa local user blocked username testuser
```

Restricting Sessions Per User—Per User Per Login

Use this task to restrict the maximum sessions per user.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] user max-logins** *max-logins* **Example:** `Switch(config)# user max-logins 1`	Restricts the maximum sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, then only one session (telnet/SSH) is allowed per user.
Step 3	**exit** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Switch(config)# exit`	

Configuring Passphrase Length

Use this task to configure the maximum and minimum passphrase length.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**userpassphrase** {{**min-length** *value* | **max-length** *value*} | **min-length** *value* **max-length** *value*} **Example:** `switch(config)# userpassphrase max-length 127`	Configures the user passphrase length. The range of minimum passphrase length values are from 8 to 127. The range of maximum passphrase length values are from 80 to 127. The default minimum passphrase length is 8 and the default maximum passphrase length is 127.
Step 3	**no userpassphrase** {**min-length** | **max-length** | **length**} **Example:** `switch(config)# no userpassphrase max-length`	Resets the passphrase length configuration to the default configuration.
Step 4	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show userpassphrase** {**min-length** | **max-length** | **length**} **Example:** `switch# show userpassphrase length`	Displays the maximum and minimum user passphrase length.

Configuring Passphrase Time Values

You can configure the following passphrase time values for a user:

- Lifetime – Life time of a passphrase in days. After the passphrase expires, the user is prompted to change the passphrase upon first login.

- Gracetime – Grace time of a passphrase in days. Gracetime is the number of days of inactivity after a passphrase has expired before an account is locked.
- Warntime – Warning time of the expiry of a passphrase in days. Warntime is the number of days prior to a passphrase expiring, when a user is warned that the user's passphrase is about to expire.

  The default time values are 99999 days for lifetime, 14 days for warntime, and 3 days for gracetime. The value 99999 indicates that a user's passphrase never expires by default.

Note By default, an extra configuration is added to the running configuration for every user except 'admin'. This indicates a user's passphrase time values. By default, the extra configuration displays the default passphrase time values for users.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**username** *username* **passphrase** {{**lifetime** | **warntime** | **gracetime**} *time-value* | {**lifetime** *time-value* **warntime** *time-value* **gracetime** *time-value*}} **Example:** `switch(config)# username test-user passphrase lifetime 990`	Configures passphrase time values for a user. Note that this step can be performed only by a network-admin.
Step 3	(Optional) **no username** *username* **passphrase** {**lifetime** | **warntime** | **gracetime** | **timevalues**} **Example:** `switch(config)# no username test-user passphrase lifetime`	Resets passphrase time value to default values for a user. Note that this step can be performed only by a network-admin.
Step 4	(Optional) **userpassphrase** {**default-lifetime** | **default-warntime** | **default-gracetime**} *time-value* **Example:** `switch(config)# userpassphrase default-lifetime 990`	Updates default passphrase time values. Note that this step can be performed only by a network-admin.

	Command or Action	Purpose
Step 5	(Optional) **no userpassphrase** {**default-lifetime** | **default-warntime** | **default-gracetime timevalue**} **Example:** `switch(config)# no userpassphrase default-lifetime`	Resets the configured default values to the initial default values. Note that this step can be performed only by a network-admin.
Step 6	(Optional) **username** *username* **expire-userpassphrase** **Example:** `switch(config)# username john expire-userpassphrase`	Sets any userpassphrase to expire immediately. When you try to log in after a passphrase expires, you are prompted to enter and create a new password after entering the old password correctly. Note that this step can be performed only by an admin.
Step 7	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 8	**show userpassphrase** {**default-lifetime** | **default-warntime** | **default-gracetime** | **timevalues**} **Example:** `switch# show userpassphrase default-lifetime`	Displays the passphrase time values.
Step 9	**show username** *username* **passphrase timevalues** **Example:** `switch# show username john passphrase timevalues`	Displays the passphrase lifetime, warning time, and grace time for a specific user.
Step 10	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the configured values.

Configuring Passphrase Time Values

The following example shows how to configure passphrase time values for test-user.

```
switch(config)# username test-user passphrase lifetime 365 warntime 10 gracetime 5
switch(config)# show username test-user passphrase timevalues
Last passphrase change(Y-M-D):  2016-01-28
Passphrase lifetime:            365 days after last passphrase change
Passphrase warning time starts: 10 days before passphrase lifetime
Passphrase Gracetime ends:      5 days after passphrase lifetime
```

```
switch# show running-config

!Command: show running-config
!Time: Mon Nov 30 02:32:51 2015

version 7.3(0)N1(1)
hostname switch

role name test
username admin password 5 5$0sCUUZQm$fXdGj90e9yXv1XeuY9qResKmLGKQtn8Tj6ab4s4IcVA  role
network-admin username test-user password 5
5$c9Gmvm8E$aoSQ1X7vfphlJ6WeRQl3C0Py6TlpiDjhWcF6kYi4hg6 expire 1970-01-01 role network-operator

username test-user passphrase  lifetime 365 warntime 10 gracetime 5
```

Locking User Accounts

As an admin, you can lock or unlock any user account.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] username** *username* **lock-user-account** **Example:** `switch(config)# username john lock-user-account`	Locks the specified user account. Use the **no** form of this command to unlock a user account.
Step 3	(Optional) **unlock locked-users** **Example:** `switch(config)# unlock locked-users`	Unlocks all the locked user accounts.
Step 4	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show locked-users** **Example:** `switch# show locked-users`	Displays all the locked users.

Logging Invalid Usernames

As an admin, you can ensure non-logging or logging of invalid usernames in logs during an authentication failure. By default, invalid usernames during authentication failures are not logged. Any username that does

not pass authentication is considered as an invalid username and it is not logged, because when a password is entered in the username field by mistake, it can get logged. This feature can be used to mitigate the risk of logging passwords.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] aaa authentication login invalid-username-log** **Example:** `switch(config)# aaa authentication login invalid-username-log`	Enables the logging of invalid usernames during an authentication failure. Use the **no** form of this command to disable the logging of invalid usernames.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	**show aaa authentication login invalid-username-log** **Example:** `switch# show aaa authentication login invalid-username-log`	Displays whether logging invalid names is enabled.

Changing Password

Use this task to change the password.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 To change the password, perform one of the following:

- Authenticate with the old password and then enter the new password:

  switch(config)# **change-password**

  Note By default, **password secure-mode** is enabled. So, users must use the old password for authentication before changing the password. An admin user can disable password secure-mode by using the **no password secure-mode** command. This enables users to change password without authenticating with the old password by using the **username** *username* **password** *new_password* command.

- If password secure-mode is enabled, an admin user can still use the **username** command to change password:

  switch(config)# **username admin password** *new-password* **role** *role-name*

  Note If password secure-mode is disabled, any user can use the **username** command to change the password.

Step 3 Exit to the privileged mode:

switch(config)# **exit**

Step 4 Display the status of password secure-mode:

switch# **show password secure-mode**

Changing Password

This example shows a running configuration to change the password. Replace the placeholders with relevant values for your setup.

```
config t
 change-password
 Enter old password:
 Enter new password:
 Confirm new password:
 exit
```

Enabling the Password Prompt for User Name

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] password prompt username** **Example:** `Switch(config)# password prompt username`	Enables the login knob. If this command is enabled and the user enters the **username** command without the password option, then the password is prompted. The password accepts hidden characters. Use the **no** form of this command to disable the login knob.
Step 3	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.

Support over SHA-256 Algorithm for Verifying OS Integrity

Use the **show file bootflash:/ sha256sum** command to display the sha256sum of the file. The sample output for this command is shown below:

```
Switch# show file bootflash:/ sha256sum

abd9d40020538acc363df3d1bae7d1df16841e4903fca2c07c7898bf4f549ef5
```

Configuring Share Key Value for using RADIUS/TACACS+

The shared secret you configure for remote authentication and accounting must be hidden. For the **radius-server key** and **tacacs-server key** commands, a separate command to generate encrypted shared secret can be used.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**generate type7_encrypted_secret** **Example:** `Switch(config)# generate type7_encrypted_secret`	Configures RADIUS and TACACS shared secret with key type 7. While generating an encrypted shared secret, user input is hidden. **Note** You can generate encrypted equivalent of plain text separately and can configure the encrypted shared secret later.
Step 3	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.

Monitoring and Clearing the Local AAA Accounting Log

The Cisco Nexus device maintains a local log for the AAA accounting activity.

Procedure

	Command or Action	Purpose
Step 1	switch# **show accounting log** [*size*] [**start-time** *year month day hh* **:** *mm* **:** *ss*]	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the size argument to limit command output.

	Command or Action	Purpose
		The range is from 0 to 250000 bytes. You can also specify a start time for the log output.
Step 2	(Optional) switch# **clear accounting log**	Clears the accounting log contents.

Verifying the AAA Configuration

To display AAA configuration information, perform one of the following tasks:

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information.
show aaa authorization	Displays AAA authorization information.
show aaa groups	Displays the AAA server group configuration.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show startup-config aaa	Displays the AAA configuration in the startup configuration.

Configuration Examples for AAA

The following example shows how to configure AAA:

```
switch(config)# aaa authentication login default group radius
switch(config)# aaa authentication login console group radius
switch(config)# aaa accounting default group radius
```

Default AAA Settings

The following table lists the default settings for AAA parameters.

Table 4: Default AAA Parameters

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled

Parameters	Default
MSCHAP authentication	Disabled
Default accounting method	local
Accounting log display length	250 KB