Configure AAA

AAA administrative models

Authentication, authorization, and accounting (AAA) is an administrative model that

- controls user access on Cisco NCS 1014 by validating user identity against a local database or a remote server,
- determines which tasks individual users can perform through task-based authorization, and
- records the network services users access and the resources they consume, enabling integration with TACACS+ and RADIUS servers and server groups.

Note From Release 24.4.1, the AAA local database supports configuring up to 3000 usernames. You can configure more than 3000 users, but the system scale and performance are not assured beyond this limit.

Deprecation of Type 7 password and Type 5 secret

Password configuration options before Release 24.4.1

Until Release 24.4.1, there were two options for configuring a password:

- Password: Uses Type 7 encryption to store the password.
- Secret: Supports Type 5, 8, 9, or 10 hashing algorithms to store the password securely.

Deprecation notice

Starting from Release 24.4.1, the use of Type 7 password and Type 5 secret is deprecated due to security concerns. The deprecation process begins from Release 24.4.1, and the full deprecation is expected in a future release. Cisco recommends using the default option, which is Type 10 secret.

This topic covers the following CLI options and use cases:

password

The **password** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#password ?
LINE The type 7 password followed by '7 ' OR SHA512-based password (deprecated, use 'secret')
```

Changes:

- All the options that were present until Release 24.4.1 are removed except LINE (to accept cleartext).
- During upgrade: Any configuration using the Type 7 password is automatically converted to Type 10 secret.

  Post-upgrade: You can still use the Type 7 password configuration option after new commits, but the password is stored as Type 10 secret.
- A new syslog message is added to indicate the deprecation process:

  ```
  %SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is deprecated.

   Converting it to a Type 10 secret for user <user name>.
  ```
- **show running configuration** command output before upgrade:

  ```
  username example
  password 7 106D000A0618
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  secret 10
  $6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iClOryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqq8st.
  !
  ```

masked-password

The **masked-password** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#masked-password ?
0 Specifies a cleartext password will follow
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
<cr> The cleartext user password
```

Changes:

- The options 7 and encrypted that were present until Release 24.4.1 are removed.
- During upgrade: Any configuration using the Type 7 password is automatically converted to Type 10 secret.

- Post-upgrade: Masked-password is an alternate method of configuring the password. You can still use the masked-password keyword with a clear string after new commits, but the password is stored as Type 10 secret.
- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is deprecated.

 Converting it to a Type 10 secret for user <user name>.
```

- **show running configuration** command output before upgrade:

```
username example
password 7 106D000A0618
!
```

show running configuration command output post-upgrade:

```
username example
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iClOryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqq8st.
!
```

password-policy

The **password-policy** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#password-policy ?
WORD Specify the password policy name

RP/0/RP0/CPU0:ios(config-un)#password-policy abcd password ?
0 Specifies an UNENCRYPTED password will follow
7 Specifies that an encrypted password will follow
LINE The UNENCRYPTED (cleartext) user password
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
encrypted Config deprecated. Will be removed in 7.7.1. Specify '7' instead.
```

Changes:

- All the options that were present until Release 24.4.1 are removed except LINE (to accept cleartext).
- During upgrade: Any configuration using the Type 7 password is automatically converted to Type 10 secret.

  Post-upgrade: You can still use the password-policy configuration option after new commits, but it is stored as Type 10 secret.
- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is deprecated.

 Converting it to a Type 10 secret for user <username>.
```

- **show running configuration** command output before upgrade:

```
username example
password-policy abcd password 7 106D000A0618
!
```

show running configuration command output post-upgrade:

```
username example
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iClOryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqq8st.
```

```
!
!
```

aaa password-policy

The **aaa password-policy** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config)#aaa password-policy abcd
RP/0/RP0/CPU0:ios(config-pp)#?
min-char-change Number of characters change required between old and new passwords
(deprecated, will be removed in 25.3.1)
restrict-password-advanced Advanced restrictions on new password (deprecated, will be removed
 in 25.3.1)
restrict-password-reverse Restricts the password to be same as reversed old password
(deprecated, will be removed in 25.3.1)
```

Changes:

- The options min-char-change, restrict-password-advanced, and restrict-password-reverse that were present until Release 24.4.1 are deprecated.
- During upgrade: These deprecated configurations do not change during upgrade.

  Post-upgrade: These deprecated keywords do not take effect when configured post-upgrade.
- New syslog messages are added to indicate the deprecation process:
  - `%SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy option 'min-char-change' is deprecated. Password/Secret will not be checked against this option now.`
  - `%SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy option 'restrict-password-reverse' is deprecated. Password/Secret will not be checked against this option now.`
  - `%SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy option 'restrict-password-advanced' is deprecated. Password/Secret will not be checked against this option now.`
- **show running configuration** command output before upgrade:

  ```
  aaa password-policy abcd
  lower-case 3
  min-char-change 1
  restrict-password-reverse
  restrict-password-advanced
  !
  ```

  show running configuration command output post-upgrade:

  ```
  aaa password-policy abcd
  lower-case 3
  min-char-change 1
  restrict-password-reverse
  restrict-password-advanced
  !
  ```

secret

The **secret** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#secret ?
0 Specifies a cleartext password will follow
```

```
10 Specifies that SHA512-based password will follow
8 Specifies that SHA256-based password will follow
9 Specifies that Scrypt-based password will follow
LINE The cleartext user password
```

```
RP/0/RP0/CPU0:ios(config-un)#secret 0 enc-type ?
<8-10> Specifies which algorithm to use. Only 8,9,10 supported [Note: Option '5' is not
available to use from 24.4]
```

Changes:

- The options 5 and encrypted are removed.
- During upgrade: Configurations using Type 5 secret remain unchanged.

  Post-upgrade: Although the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 secret.
- A new syslog message is added to indicate the deprecation process:

  ```
  %SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.
  Please use the 'secret' keyword with option type 10 for user.
  ```
- **show running configuration** command output before upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

masked-secret

The **masked-secret** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#masked-secret ?
0 Specifies a cleartext password will follow
10 Specifies that SHA512-based password will follow
8 Specifies that SHA256-based password will follow
9 Specifies that Scrypt-based password will follow
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
<cr> The cleartext user password
```

Changes:

- The options 5 and encrypted are removed.
- During upgrade: Configurations using masked-secret with Type 5 remain unchanged.

  Post-upgrade: Although the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 masked secret.
- A new syslog message is added to indicate the deprecation process:

  ```
  %SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.
  Please use the 'secret' keyword with option type 10 for user.
  ```
- **show running configuration** command output before upgrade:

```
username example
secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
!
!
```

show running configuration command output post-upgrade:

```
username example
secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
!
!
```

Special use cases

Use case 1: Configurations using both Type 7 password and secret with 8, 9, or 10 hashing, for the same user

- During upgrade:
  - For the first 3000 username configurations, the password configuration is rejected, and the secret configuration remains unchanged.
  - For the rest of the username configurations, the original secret configuration is rejected, and the password is converted to Type 10 secret.
- Post-upgrade:
  - For a new username, or a username that already existed before the upgrade, the password configuration is rejected.
  - A new syslog message is added to indicate the deprecation process:

    ```
    %SECURITY-PSLIB-4-SECRET_CONFIG_PRESENT : The password configuration is deprecated.
    Once secret is configured, cannot use password config for user <user name> at index
     <x> now.
    ```

    where 'x' is a number representing the index.

Use case 2: Configurations using both Type 7 password and Type 5 secret, for the same user

- During upgrade:
  - For any username configuration, the original Type 5 secret configuration is rejected, and the password is converted to Type 10 secret.
- Post-upgrade:
  - For a new username, or a username that already existed before the upgrade, the password configuration is converted to Type 10 secret.
  - A new syslog message is added to indicate the deprecation process:

    ```
    %SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is
    deprecated.
     Converting it to a Type 10 secret for user <username>.
    ```

TACACS+ applications

A TACACS+ application is a security application that

- centralizes user validation for Cisco NCS 1014,
- uses AAA commands and can be enabled and configured on Cisco NCS 1014, and
- provides detailed accounting information and flexible administrative control over user access.

When a TACACS+ server is configured and the protocol is enabled on the node, Cisco NCS 1014 authenticates user credentials through the TACACS+ server:

- When a user attempts to log in to the node, Cisco NCS 1014 forwards the username and password to the configured TACACS+ servers and receives the authentication status.
- If authentication fails through the TACACS+ server, Cisco NCS 1014 sends the credentials to the node and authenticates them against the local node.
- If authentication fails against the node, Cisco NCS 1014 does not allow the user to log in.

Configure a TACACS+ server

Enable AAA accounting on Cisco NCS 1014 so that the switch sends start-record and stop-record accounting notices to the TACACS+ server for each privileged EXEC process.

When AAA accounting is enabled, Cisco NCS 1014 tracks the network services that users access and the amount of network resources that they consume. The switch sends this user activity data to the TACACS+ security server as accounting records. Each record contains attribute-value pairs and is saved on the security server for later analysis, network management, client billing, or auditing.

Follow these steps to configure the TACACS+ server.

Procedure

Step 1 Enter the Cisco IOS XR configuration mode.

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Step 2 Enable TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.

Example:

```
RP/0/RP0/CPU0:ios(config)#aaa accounting exec default start-stop group TACACS_ALL
```

Step 3 Create a default command accounting method list for accounting services provided by the TACACS+ security server. Configure the list for privilege-level commands with a stop-only restriction.

Example:

```
RP/0/RP0/CPU0:ios(config)#aaa accounting exec default start-stop group TACACS_ALL
```

Cisco NCS 1014 sends AAA accounting records for each privileged EXEC process to the configured TACACS+ server.

Configure TACACS+ server groups

Group TACACS+ server hosts on Cisco NCS 1014 into named server groups that AAA method lists can reference when configuring authentication, authorization, or accounting.

Configuring AAA server groups on Cisco NCS 1014 lets you create named groups of TACACS+ server hosts. You can select a subset of configured server hosts and use them for a particular service. Each server group refers to specific host IP addresses and works alongside the global server-host list. Once configured, these groups can be referenced in AAA method lists during authentication, authorization, or accounting.

Before you begin

- Ensure the external TACACS+ server is accessible during configuration.
- If you configure the same IP address for global configuration, provide server-private parameters

Procedure

Step 1 Enter the Cisco IOS XR configuration mode.

Example:

```
RP/0/RP0/CPU0:ios# configure
```

Step 2 Create an AAA server group and enter the server group sub-configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)# aaa group server tacacs+ tacgroup1
```

Step 3 Configure the IP address of the private TACACS+ server for the group server.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs+)# server-private 10.1.1.1 port 49 key a_secret
```

Note

- You can configure a maximum of 10 TACACS+ private servers in a server group.
- If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.

Step 4 Configure the authentication and encryption key used between Cisco NCS 1014 and the TACACS+ daemon running on the TACACS+ server. If no key string is specified, the global value is used.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs+)# key 7 08984B1A4D0C19157A5F57
```

Step 5 Configure the timeout value that sets the length of time the AAA server waits to receive a response from the TACACS+ server.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs-private)# timeout 4
```

Step 6 Repeat steps 3 through 5 for every private server that you want to add to the server group.

The TACACS+ server group is available for AAA method lists on Cisco NCS 1014.

About RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a distributed client-server security system that:

- Provides protection against unauthorized access in distributed client-server networks.
- Runs RADIUS clients on Cisco NCS 1014 that send authentication and accounting requests to a central RADIUS server.
- Stores all user authentication and network service access information on the central RADIUS server.

RADIUS in the Cisco AAA framework

The Cisco AAA security model supports RADIUS along with other security protocols, such as TACACS+, Kerberos, and local username lookup.

Configure RADIUS server groups

Group external RADIUS servers on Cisco NCS 1014 into named server groups that AAA method lists can reference for authentication, authorization, or accounting.

You can enter one or more server commands to specify the hostname or IP address of an external RADIUS server, along with port numbers. After the server group is configured, AAA method lists can reference the group during authentication, authorization, or accounting. You can configure a maximum of 30 servers and 30 private servers per RADIUS server group.

Before you begin

The external RADIUS server must be accessible at the time of configuration.

Procedure

Step 1 Enter the Cisco IOS XR configuration mode.

Example:

```
RP/0/RP0/CPU0:ios# configure
```

Step 2 Group different server hosts into a distinct list and enter the server group configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)# aaa group server radius radgroup1
```

Step 3 Specify the hostname or IP address of the RADIUS server host.

Example:

```
RP/0/RP0/CPU0:ios(config)# radius-server host 192.168.20.0
```

Step 4 Specify the User Datagram Protocol (UDP) destination port for authentication requests. If you set the port to 0, the host is not used for authentication. If you do not specify the port, it defaults to 1645.

Example:

```
RP/0/RP0/CPU0:ios(config)#auth-port 1812
```

Step 5 Specify the UDP destination port for accounting requests. If you set the port to 0, the host is not used for accounting. If you do not specify the port, it defaults to 1646.

Example:

```
RP/0/RP0/CPU0:ios(config)# acct-port 1813
```

Step 6 Specify the authentication and encryption key used between Cisco NCS 1014 and the RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used.

Example:

```
RP/0/RP0/CPU0:ios(config-radius-host)#key 7 08984B1A4D0C19157A5F57
```

The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

Step 7 Repeat steps 4 through 6 for every external RADIUS server that you want to add to the server group.

Step 8 Specify the default method list for authentication and enable authentication for the console in global configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config-radius-host)#aaa authentication login default group radius local
RP/0/RP0/CPU0:ios(config-radius-host)#commit
```

Step 9 (Optional) Display information about each RADIUS server group that is configured in the system.

Example:

```
RP/0/RP0/CPU0:ios# show radius server-groups
```

The RADIUS server group is available for AAA method lists on Cisco NCS 1014.