RADIUS Authentication and Accounting

RADIUS realms

A RADIUS realm is an AAA configuration element that

- enables assigning authentication and accounting requests to specific RADIUS servers based on user domain information
- uses the domain portion of a user's Network Access Identifier (NAI) to select the appropriate RADIUS server, and
- provides realm-based filtering and control for authentication and accounting requests within a WLAN.

Feature history

Table 1: Feature history for RADIUS realms

Feature name	Release information	Feature description
RADIUS realms	Cisco IOS XE 16.9.1	RADIUS realms are configuration elements in AAA (Authentication, Authorization, and Accounting) systems. They use the domain portion of a user's Network Access Identifier (NAI)—such as the part after "@" in an email address—to direct authentication and accounting requests to specific RADIUS servers. Realms help organizations manage user access and resource usage across different groups or domains.

When mobile clients connect to a WLAN, the RADIUS realm is included in the Extensible Authentication Protocol Method for UMTS Authentication and Key Agreement (EAP-AKA) identity response of the authentication request packet. For WLAN, the NAI format for EAP-AKA can be specified as *username@domain.com*. In the NAI format, the realm follows the @ symbol and is specified as *example.com*. If the vendor-specific attribute "test" is added, the NAI format becomes *test@example.com*.

The RADIUS Realm feature can be enabled and disabled on a WLAN. If Realm is enabled on a WLAN, the corresponding user should send the username in the NAI format. The controller sends the authentication request to the AAA server only after it receives a realm in NAI format from the client and verifies that the realm complies with the required standards. Additionally, the controller sends accounting requests to the AAA server based on realm filtering.

Realm support on a WLAN

Each WLAN is configured to support NAI realms. Once the realm is enabled on a specific SSID, the system compares the received realms in the EAP identity response to those configured on the RADIUS server. If a username with the realm is not sent by the client, the WLAN uses the default RADIUS server for authentication. When the client's realm does not match the realms configured on the WLAN, the client is deauthenticated and dropped.

If the RADIUS Realm feature is not enabled on a WLAN, the controller uses the username received in the EAP identity request directly and authenticates the user using the configured RADIUS server. The RADIUS Realm feature is disabled on WLANs by default.

- **Realm match for authentication**: In dot1x with EAP methods (similar to EAP AKA), the username is received as part of an EAP identity response. A realm is derived from the username and is matched with the realms that are already configured in the corresponding RADIUS authentication server. If there is a match, the authentication requests are forwarded to the RADIUS server. If there is a mismatch, the client is deauthenticated.
- **Realm match for accounting**: After receiving a client's username through an access-accept message, the system derives the realm from the username when accounting messages are triggered. The realm is compared with the configured accounting realms on the RADIUS accounting server. When there is a match, the server receives the accounting requests. If there is no match, the system drops the accounting requests.

Enable RADIUS realm

Enable the RADIUS realm to allow the WLAN to use the RADIUS realm selection for AAA authentication.

Enabling the RADIUS realm is required when configuring wireless AAA policies that use RADIUS for authentication on your Cisco device.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a new AAA policy.**wireless aaa policy**

Example:

```
Device(config)# wireless aaa policy aaa-policy
```

Step 3 Enable AAA RADIUS realm selection.

Example:

```
Device(config-aaa-policy)# aaa-realm enable
```

Note
Use the **no aaa-realm enable** or the **default aaa-realm enable** command to disable the RADIUS realm.

The AAA RADIUS realm is now enabled and available for authentication purposes in the specified wireless AAA policy.

```
Device# configure terminal
Device(config)# wireless aaa policy policy-1
Device(config-aaa-policy)# aaa-realm enable
```

Configure realm to match the RADIUS server for authentication and accounting

Configure the realm to use your chosen RADIUS server group for authentication and accounting network access requests.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a AAA authentication model.

Example:

```
Device(config)# aaa new-model
```

Step 3 Set the authorization method.

Example:

```
Device(config)# aaa authorization network default group radius-server-group
```

Step 4 Indicate that 802.1x must use the realm group RADIUS server.

Example:

```
Device(config)# aaa authentication dot1x realm group radius-server-group
```

Step 5 Define the authentication method at login.

Example:

```
Device(config)# aaa authentication login realm group radius-server-group
```

Step 6 Enable accounting so the system sends a start record notice when a client is authorized and a stop record notice when the session ends.

Example:

```
Device(config)# aaa accounting identity realm start-stop group radius-server-group
```

Your device now uses the realm and RADIUS server group for authentication, authorization, and accounting.

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization network default group aaa_group_name
Device(config)# aaa authentication dot1x cisco.com group cisco1
Device(config)# aaa authentication login cisco.com group cisco1
Device(config)# aaa accounting identity cisco.com start-stop group v
```

Configure AAA policy for WLANs

Configure authentication, authorization, and accounting (AAA) policies to control and secure access for WLAN users.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a new AAA policy for wireless.

Example:

```
Device(config)# wireless aaa policy aaa-policy-name
```

Step 3 Enable AAA RADIUS server selection by realm.

Example:

```
Device(config-aaa-policy)# aaa-realm enable
```

Step 4 Return to global configuration mode.

Example:

```
Device(config-aaa-policy)# exit
```

Step 5 Configure a WLAN policy profile.

Example:

```
Device(config)# wireless profile policy wlan-policy-profile
```

Step 6 Map the AAA policy.**aaa-policy***aaa-policy*

Example:

```
Device(config-wireless-policy)# aaa-policy aaa-policy
```

Step 7 Set the accounting list.**accounting-list***acct-config-realm*

Example:

```
Device(config-wireless-policy)# accounting-list acct-config-realm
```

Step 8 Return to global configuration mode.

Example:

```
Device(config-wireless-policy)# exit
```

Step 9 Configure a WLAN.**wlan***wlan-namewlan-idssid*

Example:

```
Device(config)# wlan wlan-name wlan-id ssid
```

Step 10 Enable the security authentication list for IEEE 802.1x.

Example:

```
Device(config-wlan)# security dot1x authentication-list auth-list-realm
```

Step 11 Return to global configuration mode.

Example:

```
Device(config-wlan)# exit
```

Step 12 Configuresa policy tag.

Example:

```
Device(config)# wireless tag policy policy-name
```

Step 13 Map a policy profile to the WLAN.

Example:

```
Device(config-policy-tag)# wlan wlan-name policy policy-profile
```

Step 14 Return to global configuration mode.

Example:

```
Device(config-policy-tag)# exit
```

The AAA policy is successfully applied to the target WLAN, enabling centralized authentication and accounting for users connecting to that WLAN.

```
Device# configure terminal
Device(config)# wireless aaa policy aaa-policy-1
Device(config-aaa-policy)# aaa-realm enable
Device(config-aaa-policy)# exit
Device(config)# wireless profile policy wlan-policy-1
Device(config-wireless-policy)# aaa-policy aaa-policy-1
Device(config-wireless-policy)# accounting-list cisco.com
Device(config-wireless-policy)# exit
Device(config)# wlan wlan2 14 wlan-aaa
Device(config-wlan)# security dot1x authentication-list cisco.com
Device(config-wlan)# exit
Device(config)# wireless tag policy tag-policy-1
Device(config-policy-tag)# wlan Abc-wlan policy wlan-policy-a
Device(config-policy-tag)# exit
```

Verify RADIUS-realm configuration

Run this command to verify the RADIUS-realm configuration.

```
Device# show wireless client mac-address 14bd.61f3.6a24 detail
Client MAC Address : 14bd.61f3.6a24
```

```
Client IPv4 Address : 192.0.2.1
Client IPv6 Addresses : fe80::286e:9fe0:7fa6:8f4
Client Username : cisco-mac@cisco.com
AP MAC Address : 4c77.6d79.5a00
AP Name: AP4c77.6d53.20ec
AP slot : 1
Client State : Associated
Policy Profile : name-policy-profile
Flex Profile : N/A
Wireless LAN Id : 3
Wireless LAN Name: ha_realm_WLAN_WPA2_AES_DOT1X
BSSID : 4c77.6d79.5a0f
Connected For : 26 seconds
Protocol : 802.11ac
Channel : 44
Client IIF-ID : 0xa0000001
Association Id : 1
Authentication Algorithm : Open System
Client CCX version : No CCX support
Re-Authentication Timeout : 1800 sec (Remaining time: 1775 sec)
Input Policy Name  : None
Input Policy State : None
Input Policy Source : None
Output Policy Name  : None
Output Policy State : None
Output Policy Source : None
WMM Support : Enabled
U-APSD Support : Enabled
  U-APSD value : 0
  APSD ACs    : BK, BE, VI, VO
Fastlane Support : Disabled
Power Save : OFF
Supported Rates : 9.0,18.0,36.0,48.0,54.0
Mobility:
  Move Count                  : 0
  Mobility Role               : Local
  Mobility Roam Type          : None
  Mobility Complete Timestamp : 06/12/2018 19:52:35 IST
Policy Manager State: Run
NPU Fast Fast Notified : No
Last Policy Manager State : IP Learn Complete
Client Entry Create Time : 25 seconds
Policy Type : WPA2
Encryption Cipher : CCMP (AES)
Authentication Key Management : 802.1x
Encrypted Traffic Analytics : No
Management Frame Protection : No
Protected Management Frame - 802.11w : No
EAP Type : PEAP
VLAN : 113
Multicast VLAN : 0
Access VLAN : 113
Anchor VLAN : 0
WFD capable : No
Managed WFD capable : No
Cross Connection capable : No
Support Concurrent Operation : No
Session Manager:
  Interface        : capwap_9040000f
  IIF ID           : 0x9040000F
  Authorized       : TRUE
  Session timeout  : 1800
  Common Session ID: 097704090000000DF4607B3B
  Acct Session ID  : 0x00000fa2
```

```
  Aaa Server Details
  Server IP        : 192.0.2.2
  Auth Method Status List
       Method : Dot1x
              SM State         : AUTHENTICATED
              SM Bend State    : IDLE
  Local Policies:
       Service Template : wlan_svc_name-policy-profile_local (priority 254)
              Absolute-Timer   : 1800
              VLAN             : 113
  Server Policies:
  Resultant Policies:
              VLAN             : 113
              Absolute-Timer   : 1800
DNS Snooped IPv4 Addresses : None
DNS Snooped IPv6 Addresses : None
Client Capabilities
  CF Pollable : Not implemented
  CF Poll Request : Not implemented
  Short Preamble : Not implemented
  PBCC : Not implemented
  Channel Agility : Not implemented
  Listen Interval : 0
Fast BSS Transition Details :
  Reassociation Timeout : 0
11v BSS Transition : Not implemented
FlexConnect Data Switching : Central
FlexConnect Dhcp Status : Central
FlexConnect Authentication : Central
FlexConnect Central Association : No
.
.
.
Fabric status : Disabled
Client Scan Reports
Assisted Roaming Neighbor List
```

RADIUS accounting of AP events

RADIUS accounting of AP events is a network monitoring mechanism that

- Tracks the status transitions of APs within a wireless controller environment
- Records AP join and disjoin events
- Provides historical visibility into AP downtime and uptime through accounting messages sent to a RADIUS server.

Feature History

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

This feature is also available in all the releases subsequent to the one in which they are introduced in, unless noted otherwise.

Table 2: Feature history table

Feature Name	Release	Description
Device Ecosystem Data	Cisco IOS XE 17.10.1	This feature sends device analytics data that is present in the RADIUS accounting request to Cisco ISE to profile endpoints The command is introduced: • dot11-tlv-accounting
Chargeable User Identity in RADIUS Accounting	Cisco IOS XE 17.9.1	Chargeable User Identity (CUI) is a unique identifier for a client visiting a network. This attribute can be used as an alternative for the client's username as part of the authentication process. The command is introduced: • dot11-tlv-accounting
Improved Logging in RADIUS Accounting	Cisco IOS XE 17.1.1	Prior to Cisco IOS XE Amsterdam 17.1.1 release, the controller did not send accounting messages for AP join and disjoin events during network issues. From Cisco IOS XE Amsterdam 17.1.1 Release and later, the RADIUS server keeps a record of all APs that were down and have come up.

Configure accounting method-list for an AP profile

Define an accounting method list within an access point (AP) profile to enable or disable accounting for AP operations.

Use this task to specify how accounting is managed for an AP profile on your device. This allows tracking of AP events and assists with auditing and troubleshooting

Before you begin

- Identify the AP profile name you want to configure. The default AP profile name is `default-ap-profile`.
- Determine the accounting method list name you wish to apply.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the AP profile. The default AP join profile name is default-ap-profile.

Example:

```
Device(config)# ap profile ap-profile-name
```

Step 3 Configure the accounting method list for the AP profile.

Example:

```
Device(config-ap-profile)# [no] accounting method-list method-list-name
```

Use the **no** form of this command to disable the accounting method list.

The system associates the specified accounting method list with the AP profile, enabling or disabling accounting

Example

```
Device# configure terminal
Device(config)# ap profile ap-profile-name
Device(config-ap-profile)# [no] accounting method-list method-list-name
```

Verify AP accounting information

Verify the AP accounting information, including the MAC address, packets sent, packets received, and the method list.

```
Device#show wireless stats ap accounting
Base MAC       Total packet Send    Total packet Received Methodlist
-----------------------------------------------------------------------------------------
00b0.e192.0f20     4       3    abc
38ed.18cc.5788     8       8         ML_M
70ea.1ae0.af08     0       0         ML_A
```

View details for a method list configured for an AP profile.

```
Device# show ap profile name Method-list detailed
AP Profile Name               : test-profile
Description                   :
.
.
.
Method-list name              : Method-list
Packet Sequence Jump DELBA    : ENABLED
Lag status                    : DISABLED
.
Client RSSI Statistics
  Reporting                   : ENABLED
  Reporting Interval          : 30 seconds
```

Configure AAA accounting using default method list (CLI)

Use this task to monitor and record user command activity on devices through AAA accounting features.

Configure AAA accounting to track user commands executed on a controller, leveraging the default accounting method. This supports compliance and security needs.

Before you begin

- Confirm that AAA is enabled on the device.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create an accounting method list and enables accounting.

Example:

```
Device(config)# aaa accounting commands 15 default start-stop group group-name
```

- *privilege_level*: AAA accounting level. The valid range is from zero to 15.
- *group-name*: AAA accounting group that supports only TACACS+ group.

Step 3 Return to privileged EXEC mode.

Example:

```
Device(config)# end
```

The controller records user command activities according to the configured accounting method.

Configure HTTP command accounting using named method list (CLI)

Set up command accounting to track user actions on network devices via HTTP with a specified AAA method list using commands.

HTTP command accounting provides auditing and compliance by recording commands executed by users. Using a named method list offers flexibility for different accounting requirements.

Before you begin

- Ensure AAA accounting is enabled on your device.
- Have a predefined AAA accounting method list (if not, configure one).

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure HTTP command accounting using the named method list.

Example:

```
Device(config)# ip http accounting commands 1 oneacct
```

- *level*: The privilege value ranges from zero to 15. By default, the command privilege levels available on the controller are:
  - *0*: Includes the **disable**, **enable**, **exit**, **help**, and **logout** commands.
  - *1*: Includes all the user-level commands at the controller prompt (>).
  - *15*: Includes all the enable-level commands at the controller prompt (>).
- *named-accounting-method-list*: Name of the predefined command accounting method list.

Step 3 Return to privileged EXEC mode.

Example:

```
Device(config)# end
```

The device records user command activities in accordance with the configured accounting method.

RADIUS call station identifier

A RADIUS call station identifier is an attribute in the RADIUS protocol that

- allows a Network Access Server (NAS) to specify information about the endpoint (the called station) that a client is trying to access
- is included in Access-Request packets (as Called-Station-Id), and
- enables network policies by providing context for authorization decisions.

Use cases

- In dial-up scenarios, the attribute can contain the phone number dialed by the user. The NAS captures this attribute in the Access-Request packet using Dialed Number Identification (DNIS) or similar technology.
- For IEEE 802.1X authenticators (wired or wireless network access), the attribute can contain the MAC address of the bridge or AP, formatted as ASCII text.

Role in RADIUS authorization

- The RADIUS server can use this attribute to define which MAC addresses, network segments, or called stations a client is allowed to access.
- The RADIUS server can restrict or allow access based on the network resource or location to which the client is connecting. However, this is possible only in configurations that support preauthentication. In these configurations, a client attempts to authenticate before fully connecting.

The **Called-Station-Id** attribute enables the NAS to specify to the RADIUS server which endpoint or resource the client is requesting. The server uses this information to enforce connection policies.

Note The **Called-Station-Id** attribute applies only to Access-Request packets, not to Access-Accept or CoA-Request packets.

Feature history for RADIUS call station identifier

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

This feature is also available in all the releases subsequent to the one in which they are introduced in, unless noted otherwise.

Table 3: Feature history for RADIUS call station identifier

Release	Feature Information
Cisco IOS XE Bengaluru 17.4.1	The RADIUS called station identifier configuration is enhanced to include more attributes. The newly added options for authentication and accounting are: • policy-tag-name • flex-profile-name • ap-macaddress-ssid-flexprofilename • ap-macaddress-ssid-policytagname • ap-macaddress-ssid-sitetagname • ap-ethmac-ssid-flexprofilename • ap-ethmac-ssid-policytagname • ap-ethmac-ssid-sitetagname For more information on the attributes listed above, see the commands: • **radius-server attribute wireless accounting call-station-id** • **radius-server attribute wireless authentication call-station-id**

Configure a RADIUS call station identifier

Set a custom call station identifier (**Called-Station-ID**) for RADIUS authentication and accounting messages sent from the device.

Use this task to specify a policy tag name or identifier in RADIUS messages for better tracking or policy assignment.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure a call station identifier sent in the RADIUS authentication messages.

Example:

```
Device(config)# radius-server attribute wireless authentication call-station-id
policy-tag-name
```

Step 3 Configure a call station identifier sent in the RADIUS accounting messages.

Example:

```
Device(config)# radius-server attribute wireless accounting call-station-id policy-tag-name
```

The specified policy tag name is now included as the **Called-Station-ID** in all RADIUS authentication and accounting messages.

RADIUS VSA

A RADIUS VSA is a vendor-specific attribute used in RADIUS implementations that

- allows vendors to communicate specialized information between a network access server and a RADIUS server
- enables support for attributes not defined by standard RADIUS specifications, and
- is included in authentication or accounting packets, or both, according to implementation needs.

The main elements of the VSA are:

- Type: Identifies the attribute type.
- Length: Indicates attribute length.
- String (Data): Contains vendor-specific data, which includes:
  - Vendor ID
  - Vendor type
  - Vendor length
  - Vendor data

Supported modes and attributes

Authentication and accounting requests per WLAN support these VSAs in addition to the existing AAA attributes.

Table 4: Newly supported attributes

Attribute name	Well-known attribute	VSA sub-attribute	Vendor ID
SVR-Zip-Code	26	14	14369
SVR-Device-Type	26	17	14369
SVR-Device-Model-Number	26	18	14369
SVR-Lat-Long	26	19	14369
SVR-Venue-Category	26	20	14369
SVR-Network-Type	26	21	14369
Aggregation-AAA	26	22	14369
BW-Venue-Id	26	7	22472
BW-Venue-TZ	26	8	22472
BW-Class	26	10	22472
BW-Venue-Description	26	11	22472
BW-ISO-Country-Code	26	14	22472
BW-E164-Country-Code	26	15	22472
BW-State-Name	26	16	22472
BW-City-Name	26	17	22472
BW-Area-Code	26	18	22472
BW-User-Group	26	27	22472
BW-Venue-Name	26	29	22472
BW-Operator-Name	26	37	22472

Note You can use this feature only in FlexConnect central authentication mode with local switching. FlexConnect local authentication mode is not supported. Use this feature for wireless sessions only.

Create an attribute list

Create a custom AAA attribute list for use in policy mapping.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a AAA attribute list.

Example:

```
Device(config)# aaa attribute list attribute-list
```

Step 3 Specify a AAA attribute type.

Example:

```
Device(config-attr-list)# attribute type attribute-type
Device(config-attr-list)# attribute type BW-City-Name "MUMBAI"
```

Step 4 (Optional) Specify a AAA attribute type such as the state name.

Example:

```
Device(config-attr-list)# attribute type BW-State-Name "MAHARASHTRA"
```

Step 5 (Optional) Specify a AAA attribute type such as the venue name.

Example:

```
Device(config-attr-list)#attribute type BW-Venue-Name "WANKHEDE"
```

Step 6 Return to Privileged EXEC mode.

Example:

```
Device(config-attr-list)# end
```

After you create the AAA attribute list, you can use it in policy mapping.

What to do next

Create a AAA policy and map the attribute list.

Map a AAA policy to WLAN policy profile

Associate a AAA policy with a WLAN policy profile. This enforces authentication and authorization rules on wireless clients.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a new wireless policy profile.

Example:

```
Device(config)# wireless profile policy EAP-AKA profile-policy
```

Step 3 Create a new AAA policy.

Example:

```
Device(config-wireless-policy)# aaa-policy Verizon-aaa-policy aaa-policy
```

Step 4 Return to Privileged EXEC mode.

Example:

```
Device(config-wireless-policy)# end
```

You have mapped the AAA policy to the WLAN policy profile.

What to do next

Map the WLAN policy profile to a WLAN.

Map the WLAN policy profile to a WLAN

Associate a previously configured WLAN policy profile to a WLAN on a Cisco device.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a new policy tag.

Example:

```
Device(config)# wireless tag policy policy-name
```

Step 3 Map the policy profile to a WLAN.

Example:

```
Device(config-policy-tag)# wlan wlan-profile-name policy aaa-policy
```

Step 4 Return to Privileged EXEC mode.

Example:

```
Device(config-policy-tag)# end
```

The specified WLAN policy profile is successfully mapped to the WLAN.

```
Device# configure terminal
Device(config)# wireless tag policy EAP-AKA
Device(config-policy-tag)# wlan EAP-AKA policy EAP-AKA
Device(config-policy-tag)# end
```