Introduction

This document explains how to configure Access Point (AP) authentication policy. This feature ensures that only authorized Access Points (APs) are able to join a Catalyst 9800 Wireless LAN Controller. This document does not cover the case of mesh (1500 series) APs which require a mac filter entry to join the controller but do not follow the typical AP authorization flow (see references).

Background Information

To authorize an Access Point (AP), Ethernet MAC address of the AP needs to be authorized against local database with 9800 Wireless LAN Controller or against an external Remote Authentication Dial-In User Service (RADIUS) server.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- 9800 WLC
- Command line Interface (CLI) access to the wireless controllers

Components Used
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Configure

Network Diagram

![Network Diagram]

Configurations

MAC AP authorization List - Local

The MAC address of the authorized APs are stored locally in the 9800 WLC.

Step 1. Create a local authorization credential-download method list.

Navigate to Configuration > Security > AAA > AAA Method List > Authorization > + Add
Step 2. Enable AP MAC authorization.

Navigate to Configuration > Security > AAA > AAA Advanced > AP Policy. Enable Authorize APs against MAC and select the Authorization Method List created in Step 1.

Step 3. Add the AP's ethernet mac address.

Navigate to Configuration > Security > AAA > AAA Advanced > Device Authentication >
**Note:** AP’s ethernet mac address must be in one of these formats when entered in the web UI (xx:xx:xx:xx:xx:xx (or) xxxx.xxxx.xxxx (or) xx-xx-xx-xx-xx-xx) in version 16.12. In version 17.3, they have to be in format xxxxxxxxxxx without any separator. The CLI format is always xxxxxxxxxxxx in any version (in 16.12, the web UI removes the separators in the config). CSCvv43870 will allow to use any format in CLI or web UI in later releases.

**CLI:**

```
# config t
# aaa new-model # aaa authorization credential-download <AP-auth> local
# ap auth-list authorize-mac # ap auth-list method-list <AP-auth> # username <aaaabbbbcccc> mac
```

**MAC AP Authorization List - External RADIUS server**

**9800 WLC Config**

The MAC address of the authorized APs are stored on an external RADIUS server, in this example ISE.
On ISE, you can register the MAC address of the APs either as usernames/password or as Endpoints. Along the steps you are instructed which setting to select to use one way or the other.

GUI:

Step 1. Declare RADIUS server

Navigate to **Configuration > Security > AAA > Servers / Groups > RADIUS > Servers > + Add** and enter the RADIUS server’s information.

Ensure **Support for CoA** is enabled if you plan to use Central Web Authentication (or any kind of security that requires CoA) in the future.

![Add RADIUS server](image)

Step 2. Add the RADIUS server to a RADIUS group

Navigate to **Configuration > Security > AAA > Servers / Groups > RADIUS > Server Groups > + Add**

To have ISE authenticate the AP’s MAC address as usernames leave MAC-Filtering as none.
To have ISE authenticate the AP’s MAC address as Endpoints change MAC-Filtering to mac.

Step 3. Create an authorization credential-download method list.

Navigate to **Configuration > Security > AAA > AAA Method List > Authorization > + Add**
Step 4. Enable AP MAC authorization.

Navigate to Configuration > Security > AAA > AAA Advanced > AP Policy. Enable Authorize APs against MAC and select the Authorization Method List created in Step 3.
CLI:

```bash
# config t
# aaa new-model

# radius server <radius-server-name>
# address ipv4 <radius-server-ip> auth-port 1812 acct-port 1813
# timeout 300
# retransmit 3
# key <shared-key>
# exit

# aaa group server radius <radius-grp-name>
# server name <radius-server-name>
# exit

# aaa server radius dynamic-author
# client <radius-server-ip> server-key <shared-key>

# aaa authorization credential-download <AP-auth> group <radius-grp-name> # ap auth-list authorize-mac
# ap auth-list method-list <AP-ISE-auth>

ISE Config

Step 1. Follow these instructions to add 9800 WLC to ISE

**Declare 9800 WLC on ISE**

Based on how you want to authenticate the AP's MAC address follow the required steps:

**Configure USE to authenticate MAC address as endpoints**

**Configure ISE to authenticate MAC address as username/password**

Configure ISE to authenticate MAC address as endpoints

Step 2. (Optional) Create an identity group for Access Points

Navigate to Administration > Identity Management > Groups > Endpoint Identity Groups > + Add.

Choose a name and click Submit.
Step 3. Add the AP's ethernet mac address to its endpoint identity group.

Navigate to **Work Centers > Network Access > Identities > Endpoints > +**

Enter the needed information.
Step 4. Verify the identity store used on your default authentication rule contains the internal endpoints.

A. Navigate to **Policy > Authentication** and take note of the Identity store.

B. Navigate to **Administration > Identity Management > Identity Source Sequences > Identity Name**.
C. Ensure Internal Endpoints belongs to it, if not, add it.
Configure ISE to authenticate MAC address as username/password

Step 2. (Optional) Create an identity group for Access Points

Navigate to Administration > Identity Management > Groups > User Identity Groups > + Add.
Choose a name and click **Submit**.

Step 3. Verify that your current password policy allows you to add a mac address as username and password.

Navigate to **Administration > Identity Management > Settings > User Authentication Settings > Password Policy** and ensure that at least these options are disabled:
Note: You can also want to disable the option **Disable user account after XX days if password was not changed.** As this is a mac address, the password never changes.

Step 4. Add the AP's ethernet mac address.

Navigate to **Administration > Identity Management > Identities > Users > + Add**
Enter the needed information.

**Network Access User**

- **Name**: aabbbccc
- **Status**: Enabled
- **Email**: 

**Passwords**

- **Password Type**: Internal Users
- **Login Password**: ************
- **Re-Enter Password**: ************
- **Enable Password**: 

**User Information**

- **First Name**: 
- **Last Name**: 

**Account Options**

- **Description**: 
- **Change password on next login**: 

**Account Disable Policy**

- **Disable account if date exceeds**: 2018-11-09 (yyyy-mm-dd)

**User Groups**

- **AccessPoints**

[Submit] [Cancel]
Note: Name and Login Password field must be the ethernet MAC address of the AP, all lower case and no separators.

Authorization Policy to Authenticate APs

Navigate to Policy > Authorization as shown in the image.

Insert a new rule as shown in the image.

First, select a name for the rule and the Identity group where the Access Point is stored (AccessPoints). Select User Identity Groups if you decided to authenticate the mac address as username password or Endpoint Identity Groups if you choose to authenticate the AP’s MAC address as endpoints.
After that, select other conditions that do the authorization process to fall into this rule. In this example, the authorization process hits this rule if it uses service-type Call Check and the authentication request comes from the IP address 10.88.173.52.

Finally, select the Authorization profile that is assigned to the clients that hit that rule, click Done and save it as shown in the image.

**Note:** APs that already joined in the controller does not lose their association. If, however, after enabling authorization list, they lose communication with the controller and attempt to join back, they go through the authentication process. If their mac addresses are not listed locally or in the RADIUS server they are not be able to join back to the controller.

**Verify**

Verify if 9800 WLC has enabled ap authentication list

# show ap auth-list

Authorize APs against MAC : Disabled
Authorize APs against Serial Num : Enabled
Authorization Method List : <auth-list-name>
Verify radius configuration:
WLC 9800 provides ALWAYS-ON tracing capabilities. This ensures all AP join related errors, warning and notice level messages are constantly logged and you can view logs for an incident or failure condition after it has occurred.

**Note:** Depending on volume of logs being generated, you can go back few hours to several days.

In order to view the traces that 9800 WLC collected by default, you can connect via SSH/Telnet to the 9800 WLC and follow these steps (Ensure you are logging the session to a text file).

**Step 1.** Check controller’s current time so you can track the logs in the time back to when the issue happened.

```
# show clock
```

**Step 2.** Collect syslogs from the controller’s buffer or the external syslog as dictated by the system configuration. This provides a quick view into the system health and errors, if any.

```
# show logging
```

**Step 3.** Verify if any debug conditions are enabled.

```
# show debugging
```

IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop

IOSXE Packet Tracing Configs:

Packet Infra debugs:

<table>
<thead>
<tr>
<th>Ip Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------</td>
<td>------</td>
</tr>
</tbody>
</table>

**Note:** If you see any condition listed, it means the traces are being logged up to debug level for all the processes that encounter the enabled conditions (mac address, ip address etc). This would increase the volume of logs. Therefore, it is recommended to clear all conditions when not actively debugging

**Step 4.** Assuming mac address under test was not listed as a condition in Step 3, collect the always-on notice level traces for the specific radio mac address.

```
# show logging profile wireless filter { mac | ip } { <aaaa.bbbbb.ccccc> | <a.b.c.d> } to-file always-on=<FILENAME.txt>
```
You can either display the content on the session or you can copy the file to an external TFTP server.

```
# more bootflash:always-on-<FILENAME.txt>
```

or

```
# copy bootflash:always-on-<FILENAME.txt> tftp://a.b.c.d/path/always-on-<FILENAME.txt>
```

**Conditional Debugging and Radio Active Tracing**

If the always-on traces do not give you enough information to determine the trigger for the problem under investigation, you can enable conditional debugging and capture Radio Active (RA) trace, which will provide debug level traces for all processes that interact with the specified condition (client mac address in this case). In order to enable conditional debugging, follow these steps.

**Step 5.** Ensure there are no debug conditions are enabled.

```
# clear platform condition all
```

**Step 6.** Enable the debug condition for the wireless client mac address that you want to monitor.

This command starts to monitor the provided mac address for 30 minutes (1800 seconds). You can optionally increase this time to up to 2085978494 seconds.

```
# debug wireless mac <aaaa.bbbb.cccc> {monitor-time <seconds>}
```

**Note:** In order to monitor more than one client at a time, run `debug wireless mac <aaaa.bbbb.cccc>` command per mac address.

**Note:** You do not see the output of the client activity on terminal session, as everything is buffered internally to be viewed later.

**Step 7.** Reproduce the issue or behavior that you want to monitor.

**Step 8.** Stop the debugs if the issue is reproduced before the default or configured monitor time is up.

```
# no debug wireless mac <aaaa.bbbb.cccc>
```

Once the monitor-time has elapsed or the debug wireless has been stopped, the 9800 WLC generates a local file with the name:

```
ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log
```

**Step 9.** Collect the file of the mac address activity. You can either copy the ra trace .log to an external server or display the output directly on the screen.

Check the name of the RA traces file.
Copy the file to an external server:

```plaintext
```

Display the content:

```plaintext
copy bootflash:ra_trace_MAC_aaaaabbbcccc_HHMMSS.XXX_timezone_DayWeek.Month_Day_year.log
```

Step 10. If the root cause is still not obvious, collect the internal logs which are a more verbose view of debug level logs. You do not need to debug the client again as we are only taking a further detailed look at debug logs that have been already collected and internally stored.

```plaintext
show logging profile wireless internal filter { mac | ip } { <aaaa.bbbb.cccc> | <a.b.c.d> }
to-file ra-internal-<FILENAME>.txt
```

**Note:** This command output returns traces for all logging levels for all processes and is quite voluminous. Please engage Cisco TAC to help parse through these traces.

You can either copy the ra-internal-FILENAME.txt to an external server or display the output directly on the screen.

Copy the file to an external server:

```plaintext
copy bootflash:ra-internal-<FILENAME>.txt tftp://a.b.c.d/ra-internal-<FILENAME>.txt
```

Display the content:

```plaintext
more bootflash:ra-internal-<FILENAME>.txt
```

Step 11. Remove the debug conditions.

```plaintext
clear platform condition all
```

**Note:** Ensure that you always remove the debug conditions after a troubleshooting session.

**References**

Join mesh APs to 9800 WLC