FlexConnect Configuration with Central and Local Authentication on Catalyst 9800 Wireless Controllers

Contents

Introduction
Background Information
Prerequisites
Requirements
Components Used
Configure
Network Diagram
Configurations
AAA Configuration on C9800
WLAN Configuration
Policy Profile Configuration
Policy Tag Configuration
Flex Profile Configuration
Site Tag Configuration
Local Authentication with external RADIUS Server
Verify
Troubleshoot

Introduction

This document describes how to configure a 9800 WLC for FlexConnect central and local authentication.

Background Information

FlexConnect is a wireless solution for remote office deployment. It allows customer to configure Access Points (APs) in remote locations from the corporate office through a Wide Area Network (WAN) link without deploying a controller in each location. The FlexConnect APs can switch the client data traffic locally and perform client authentication locally when the connection to the controller is lost. In connected mode, the FlexConnect APs can also perform local authentication.

Prerequisites

Requirements
Cisco recommends that you have knowledge of these topics:

- Catalyst Wireless 9800 configuration model
- FlexConnect
- 802.1x

**Components Used**

The information in this document is based on these software and hardware versions:

- C9800-CL v16.10

**Configure**

**Network Diagram**

![Network Diagram](image)

**Configurations**

**AAA Configuration on C9800**

You can follow the instructions from this link:

[AAA Configuration on 9800 WLC](#)

**WLAN Configuration**

Step 1. Navigate to **Configuration > Wireless > WLANs** and click **+Add** to create a new WLAN.
Step 2. Enter the WLAN information

Navigate to the **Security** tab to configure the Layer2/Layer3 security mode as long as the encryption method, and Authentication List in case 802.1x is in use. Then click **Save & Apply to Device.**
Policy Profile Configuration

Step 1. Navigate to Configuration > Tags & Profiles > Policy and click +Add to create a Policy Profile. Add the name and uncheck the Central Switching box. With this setup, the controller handles client authentication, and the FlexConnect Access Point switches client data packets locally.
Step 2. Navigate to the **Access Policies** tab to assign the VLAN to which the wireless clients are assigned when they connect to this WLAN by default.

You can either select one VLAN name from the drop down or manually type a VLAN id.
Step 4. Navigate to the **Advanced** tab to configure the WLAN timeouts, DHCP, WLAN Flex Policy, and AAA policy in case they are in use. Then click **Save & Apply to Device**.
Policy Tag Configuration

Step 1. Navigate to Configuration > Tags & Profiles > Tags > Policy > +Add. Assign a name, and map the Policy Profile and WLAN Profile create before.
Flex Profile Configuration

Step 1. Navigate to Configuration > Tags & Profiles > Flex and click +Add to create a new one.

![Add Flex Profile](image)

**Note:** Native VLAN ID refers to the VLAN used by the APs that will get this Flex Profile assigned, and it should be the same VLAN ID configured as native on the switchport where the APs are connected.

Step 2. Under the VLAN tab, add the needed VLANs, those assigned by default to the WLAN through a Policy Profile, or the ones pushed by a RADIUS server. Then click Save & Apply to Device.
Note: At step 2 of section Policy Profile Configuration, you select the default VLAN assigned to the SSID. If you use a VLAN name on that step, ensure that you use the same VLAN name on the Flex Profile configuration, otherwise clients won't be able to connect to the WLAN.

Site Tag Configuration

Step 1. Navigate to Configuration > Tags & Profiles > Tags > Site and click +Add to create a new Site tag. Uncheck the Enable Local Site box to allow APs to switch the client data traffic locally, and add the Flex Profile created above.

Note: As Enable Local Site is disabled, the APs that get this Site tag assigned will be configured as FlexConnect mode.

Step 2. Navigate to Configuration > Wireless > Access Points > AP name to add the Site Tag and Policy Tag to an associated AP. This will cause the AP to restart its CAPWAP tunnel and join back to the 9800 WLC.
Once the AP joins back, notice the AP is now in FlexConnect mode.

Local Authentication with external RADIUS Server

Step 1. Add the AP as a network device into the RADIUS server.

Refer to the following link to see an example using the Identity Service Engine (ISE) as the RADIUS server:


Step 2. Create a WLAN.

The configuration can be the same as the one previously configured.

Step 3. Policy Profile Configuration.

It can be either create a new one or use the previously configured. This time, uncheck the Central
Step 4. **Policy tag** configuration.

Associate the **WLAN** configured in the step 1 and the **Policy Profile** created in the step 3 of this section.

Step 5. **Flex Profile** Configuration.

Create a Flex Profile, navigate to the **Local Authentication tab**, configure the **Radius Server Group** and check the **RADIUS** box.
Step 6. **Site tag** configuration.

Configure the Flex Profile configured in the step 5, and uncheck the **Enable Local Site** box.

**Verify**

Navigate to **Monitoring > Wireless > Clients** and confirm the **Policy Manager State** state and the FlexConnect parameters.

Central Authentication:
**Local Authentication:**

- **Policy Manager State**: Run
- **DNS Snooped IPv4 Addresses**: None
- **DNS Snooped IPv6 Addresses**: None
- **FlexConnect Data Switching**: Local
- **FlexConnect DHCP Status**: Local
- **FlexConnect Authentication**: Central
- **FlexConnect Central Association**: Yes
You can use these commands to verify current configuration:

```bash
# show wlan { summary | id | name | all }
# show run wlan
# show run aaa
# show aaa servers
# show ap config general
# show ap name <ap-name> config general
# show ap tag summary
# show ap name <AP-name> tag detail
# show wlan { summary | id | name | all }
# show wireless tag policy detailed <policy-tag-name>
# show wireless profile policy detailed <policy-profile-name>
```

**Troubleshoot**

WLC 9800 provides ALWAYS-ON tracing capabilities. This ensures all client connectivity 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
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 mac address.

```
# show logging profile wireless filter { mac | ip } { <aaaa.bbbb.cccc> | <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.

```bash
# clear platform condition all
```

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

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

```bash
# 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.

```bash
# 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_aaaabbbbccccc_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

```bash
# dir bootflash: | inc ra_trace
```

Copy the file to an external server:

```bash
# copy bootflash:ra_trace_MAC_aaaabbbbccccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log tftp://a.b.c.d/ra-FILENAME.txt
```

Display the content:
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.

```
# 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:

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

Display the content:

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

Step 11. Remove the debug conditions.

```
# clear platform condition all
```

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