# Configure Catalyst 9800 Wireless Controller AP Authorization List

## Contents



## Introduction

This document describes how to configure Catalyst 9800 Wireless LAN Controller Access Point (AP) authentication policy.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

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

### Components Used

Cisco recommends these hardware and software versions:

- 9800 WLC v17.3
- AP 1810W
- AP 1700
- Identity Service Engine (ISE) v2.2

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.

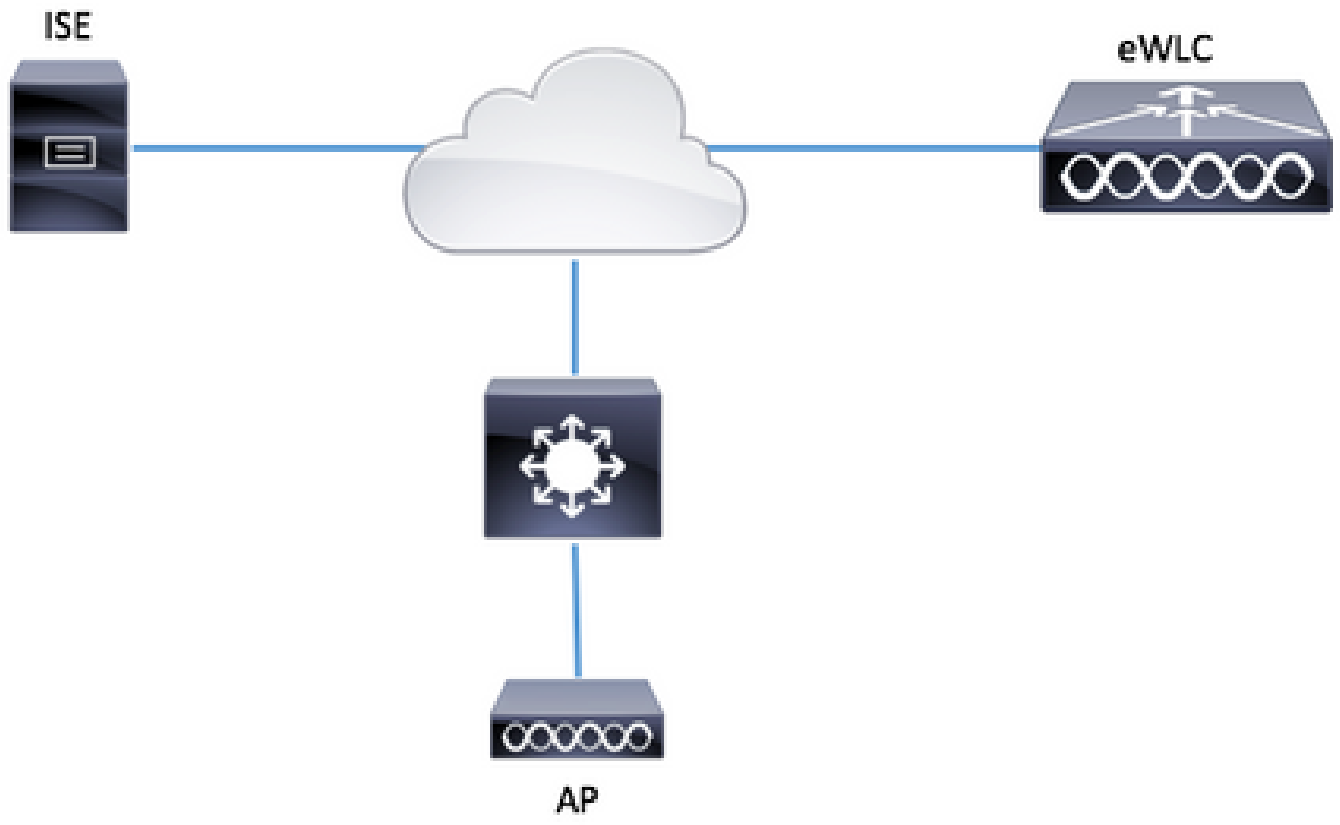
# Background Information

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 externalRemote Authentication Dial-In User Service (RADIUS) server.

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 trace the typical AP authorization flow (see references).

# Configure

## Network Diagram



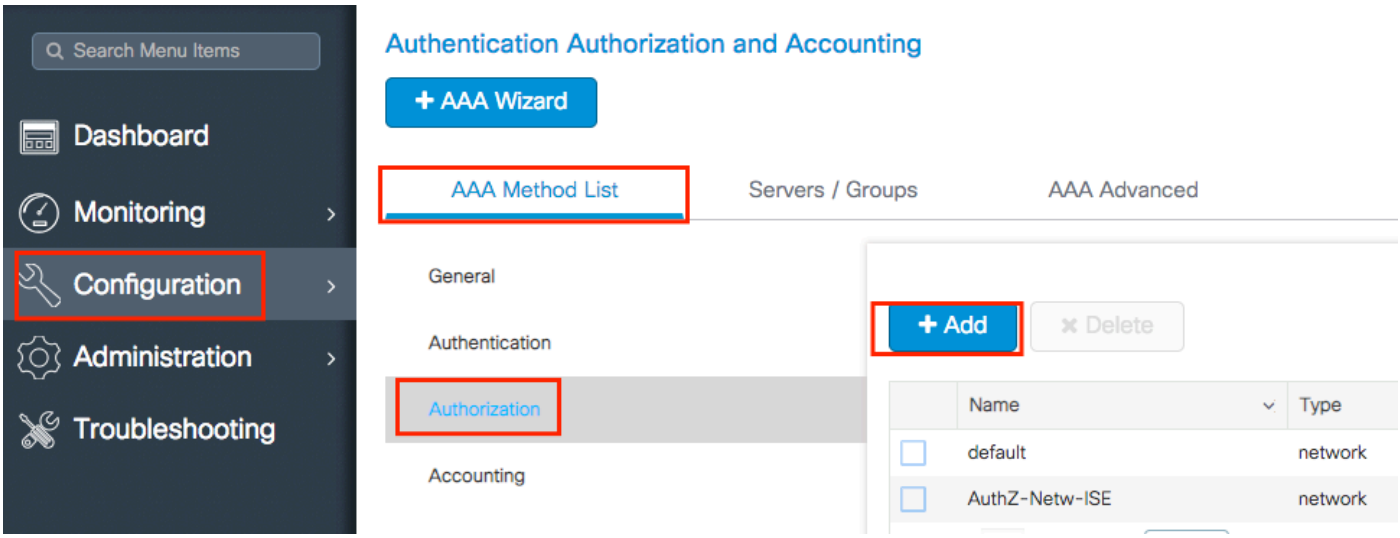

## Configurations

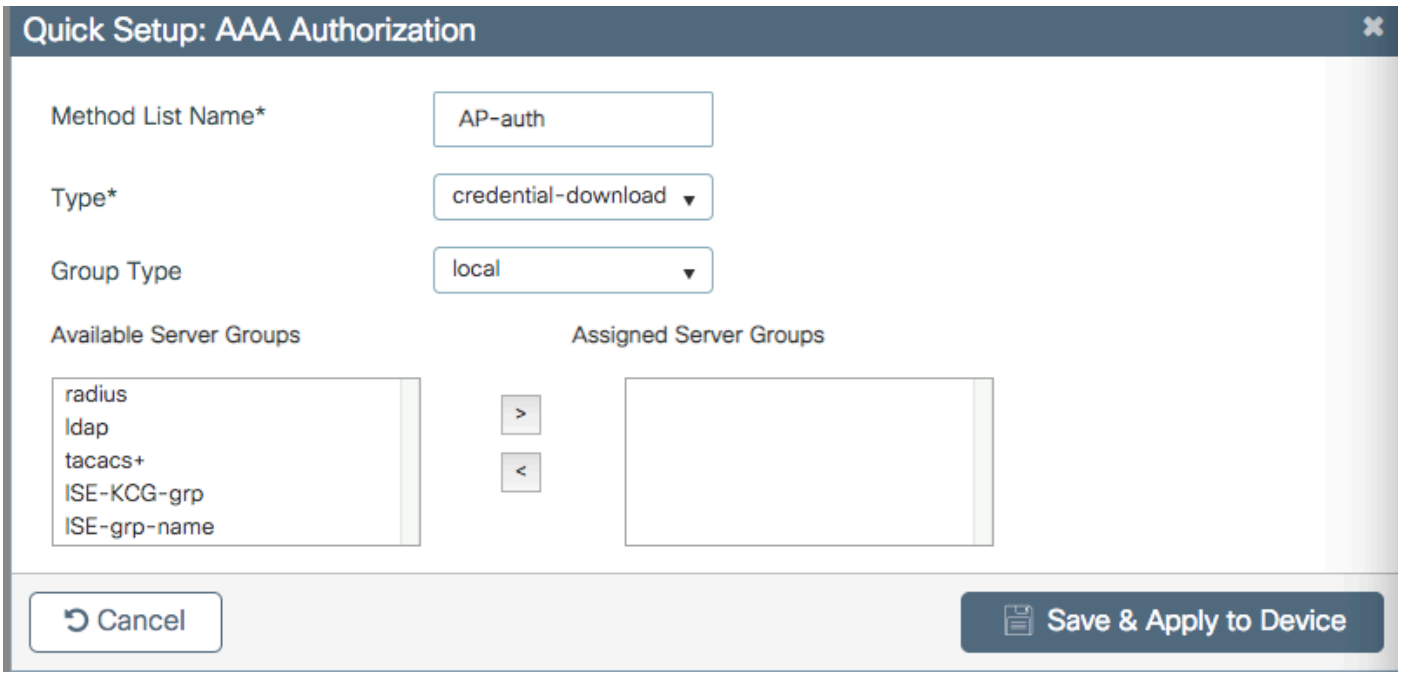
### MAC AP authorization List - Local

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

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

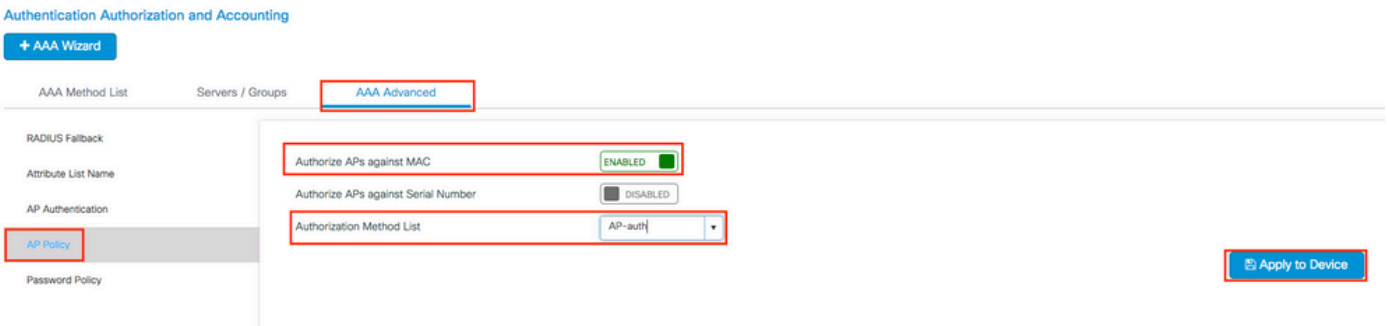
Navigate to **Configuration > Security > AAA > AAA Method List > Authorization > + Add**.



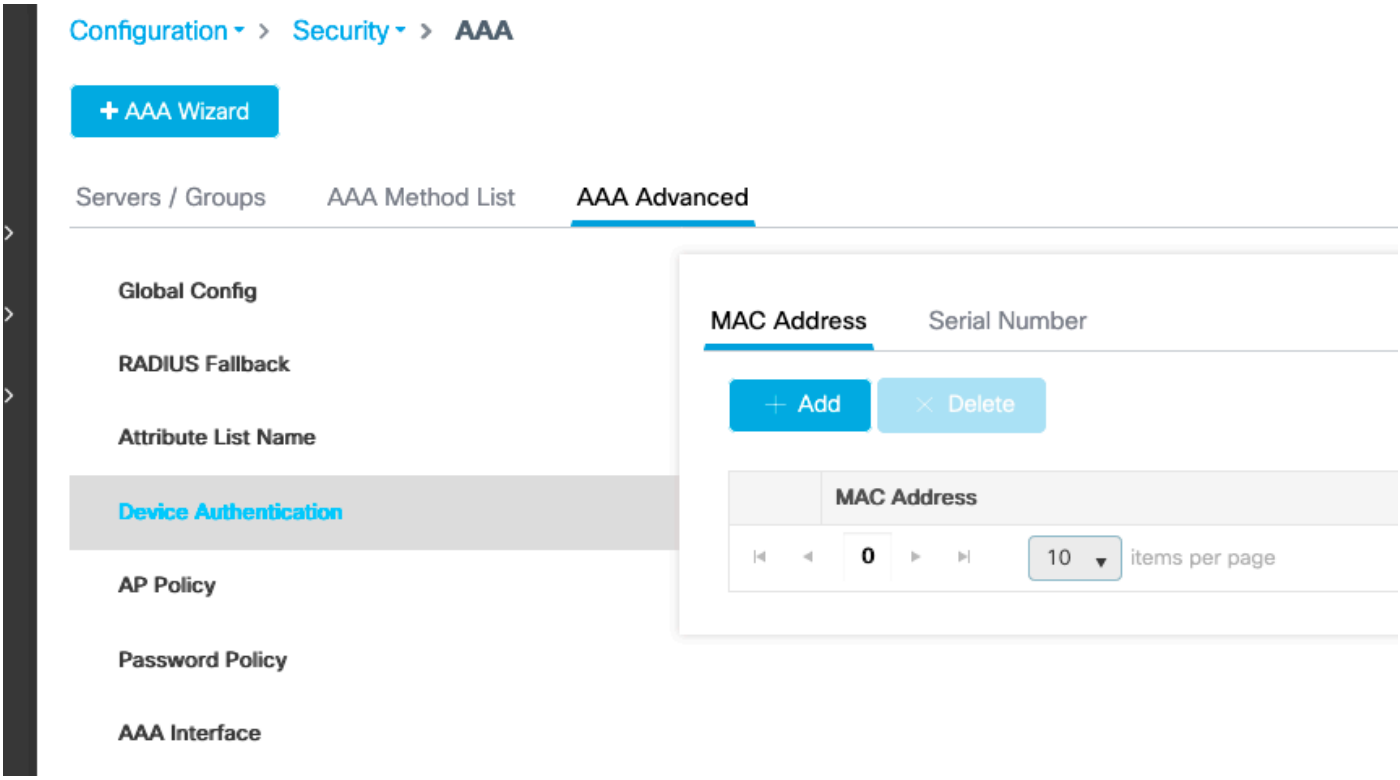

Step 2. Enable AP MAC authorization.

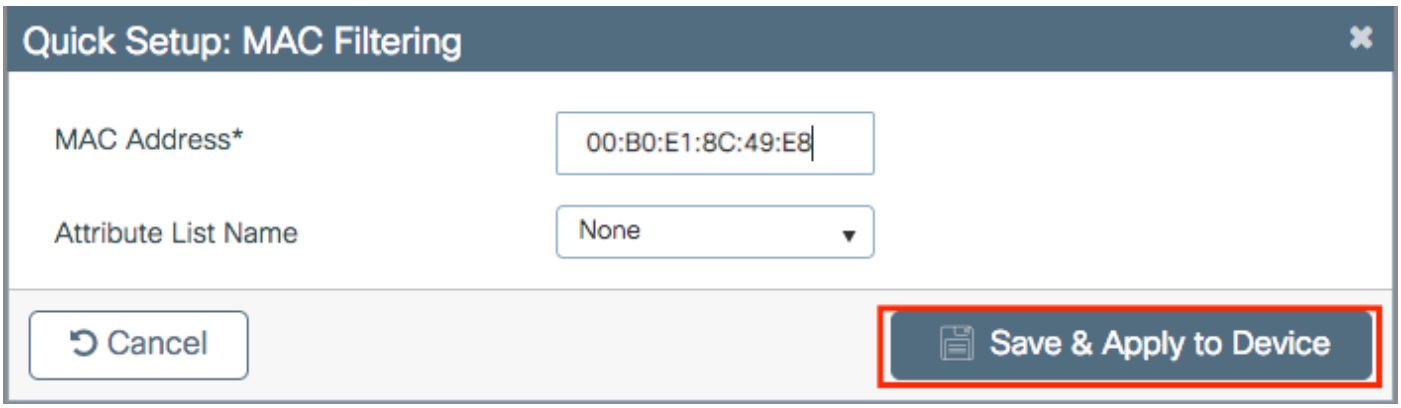
Navigate to **Configuration > Security > AAA > AAA Advanced > AP Policy**. Enable **Authorize APs against MAC** and select the **Authorization Method List** created in Step 1.




Step 3. Add the AP ethernet MAC Address.

Navigate to **Configuration** > **Security** > **AAA** > **AAA Advanced** > **Device Authentication** > **MAC Address** > **+ Add**.




**Note**: AP 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 xxxxxxxxxxxx without any separator. The CLI format is always xxxxxxxxxxxx in any version (in 16.12, the web UI removes the separators in the config). Cisco bug ID CSCvv43870 allows the use of 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**

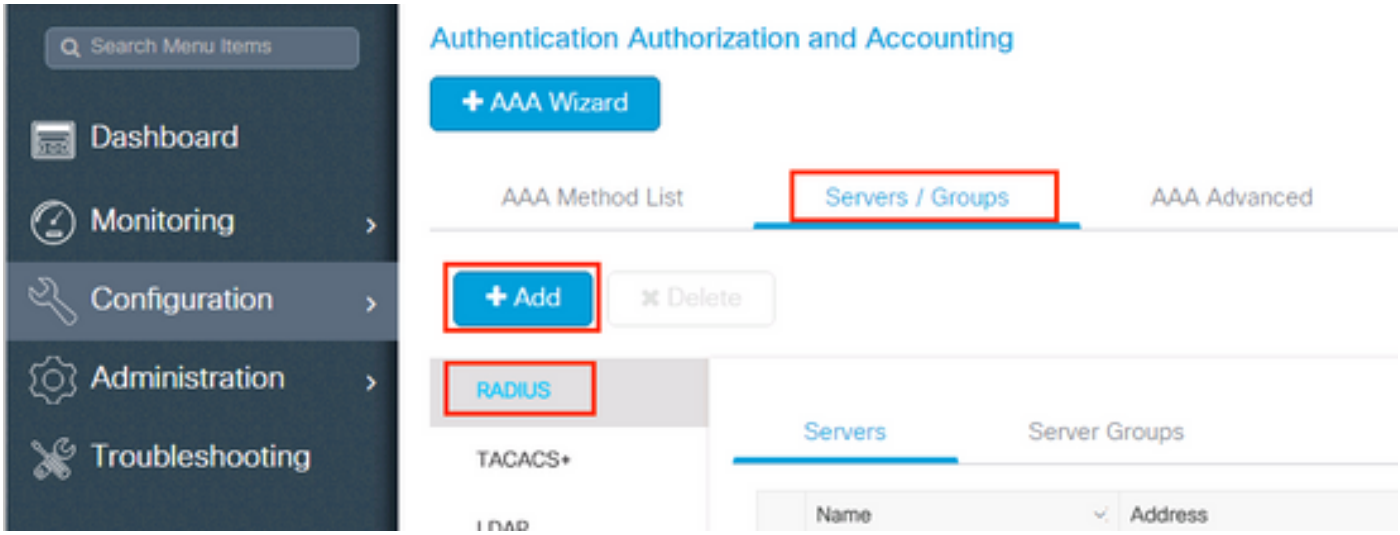
The MAC address of the authorized APs are stored on an external RADIUS server, in this example ISE.

On ISE, you can register the MAC address of the APs either as usernames/password or as Endpoints. Along the steps you are instructed how to select to use one way or the other.

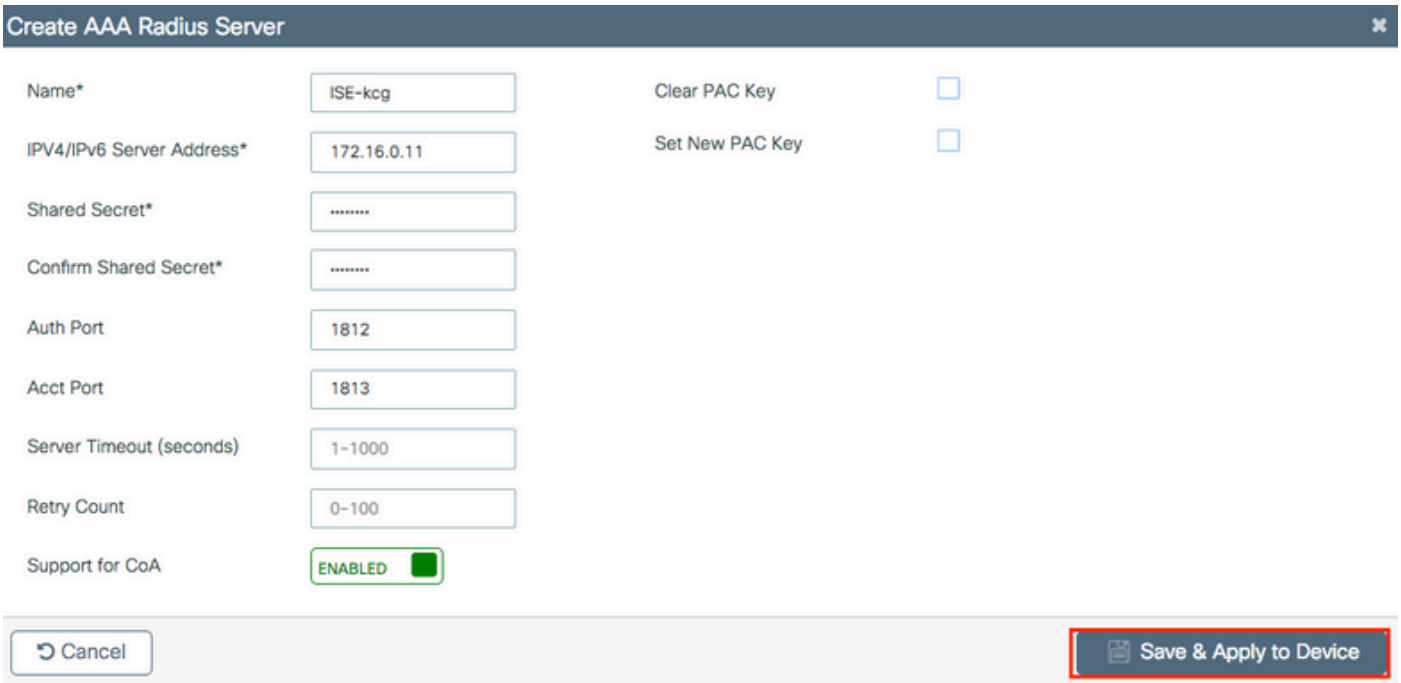
GUI:

Step 1. Declare RADIUS server.

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

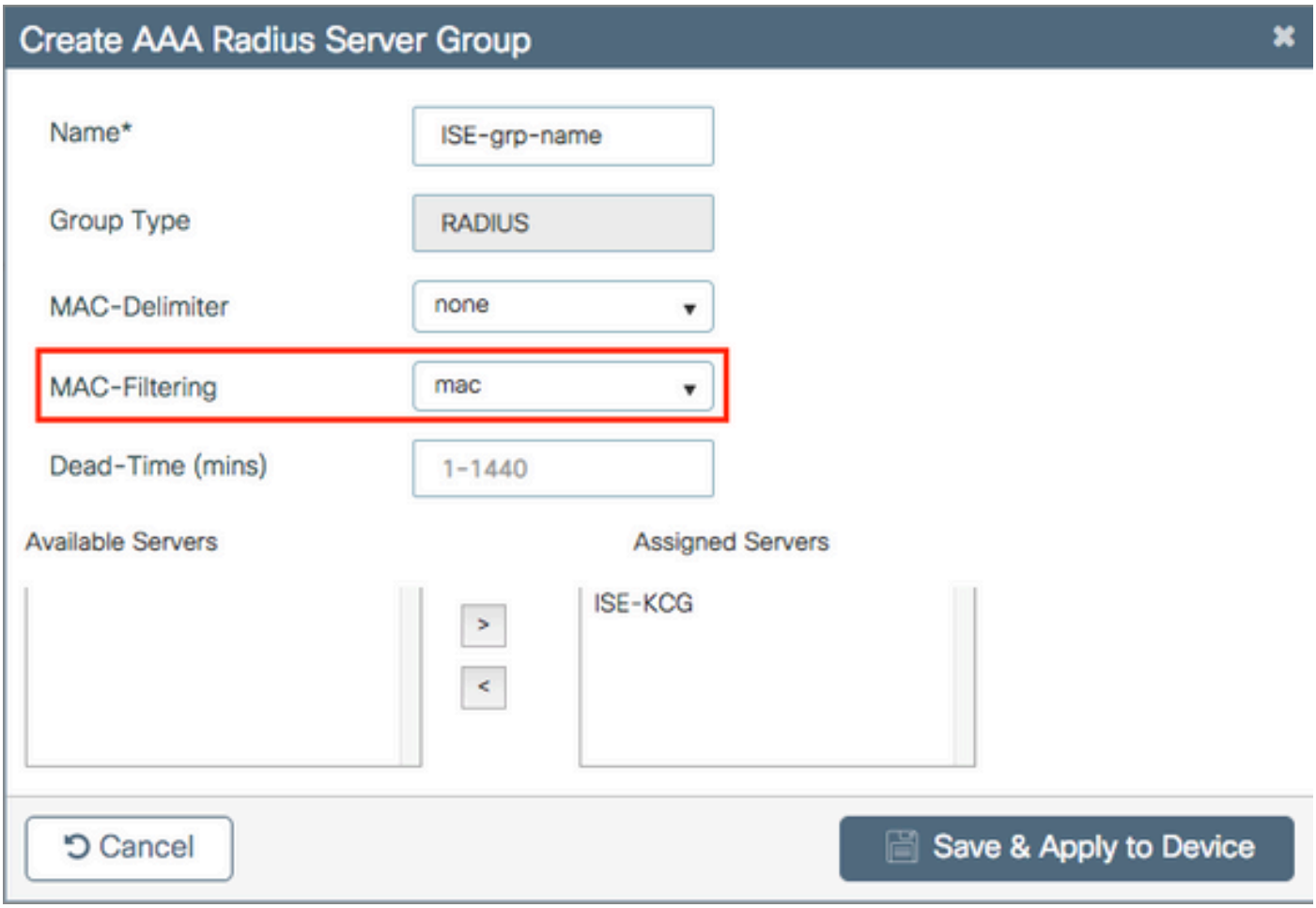


Step 2. Add the RADIUS server to a RADIUS group.

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

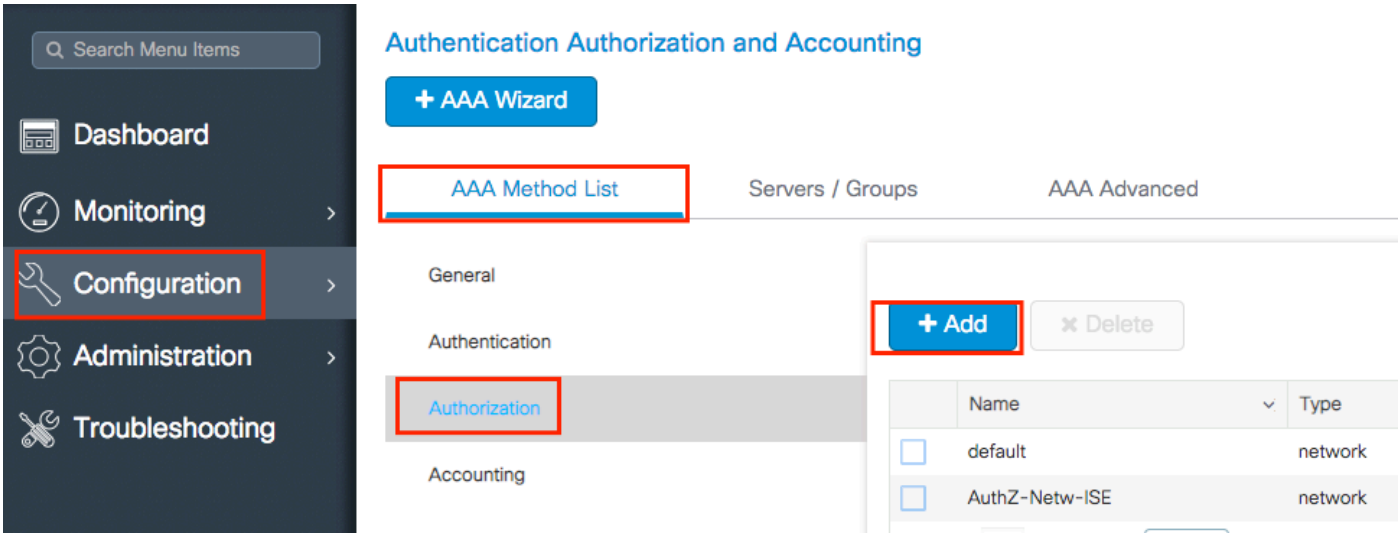
To have ISE authenticate the AP MAC address as usernames, leave MAC-Filtering as none.

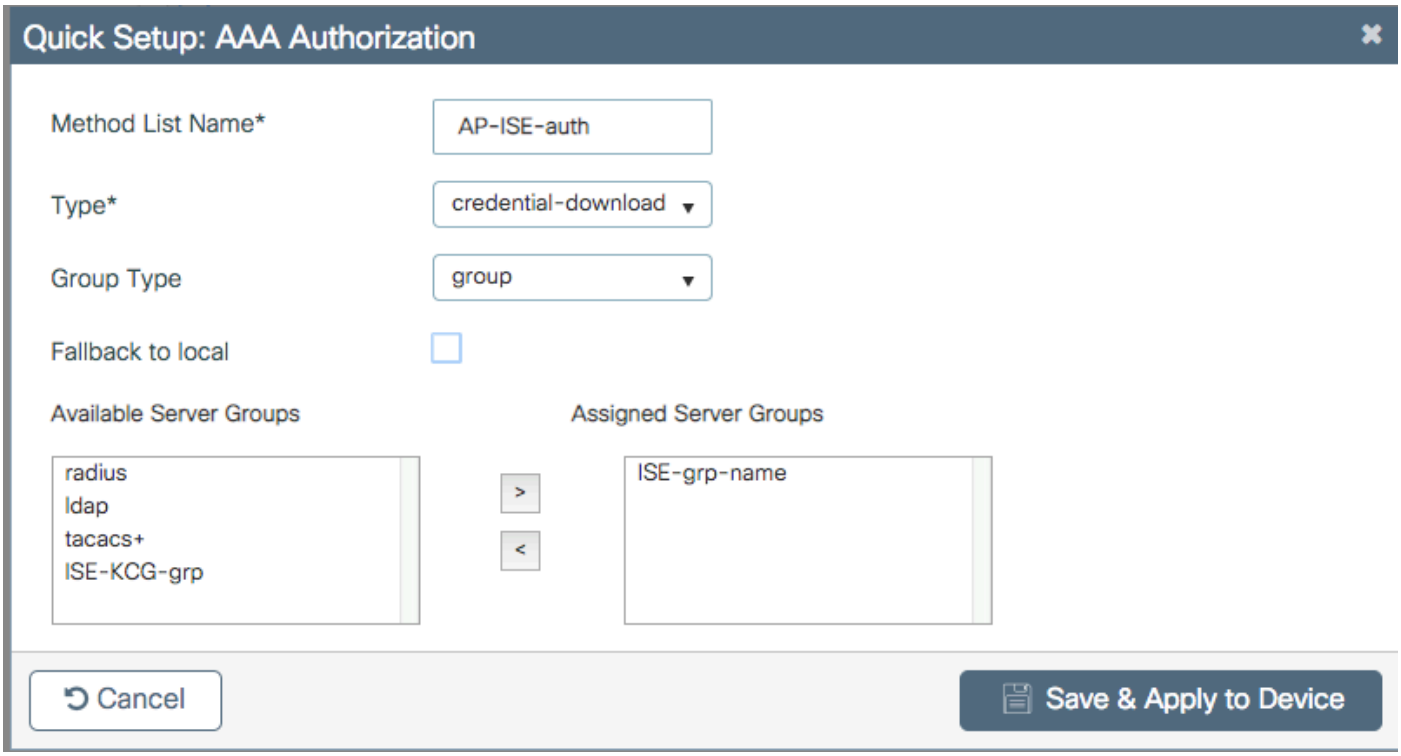

To have ISE authenticate the AP MAC address as Endpoints, change MAC-Filtering to MAC.




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

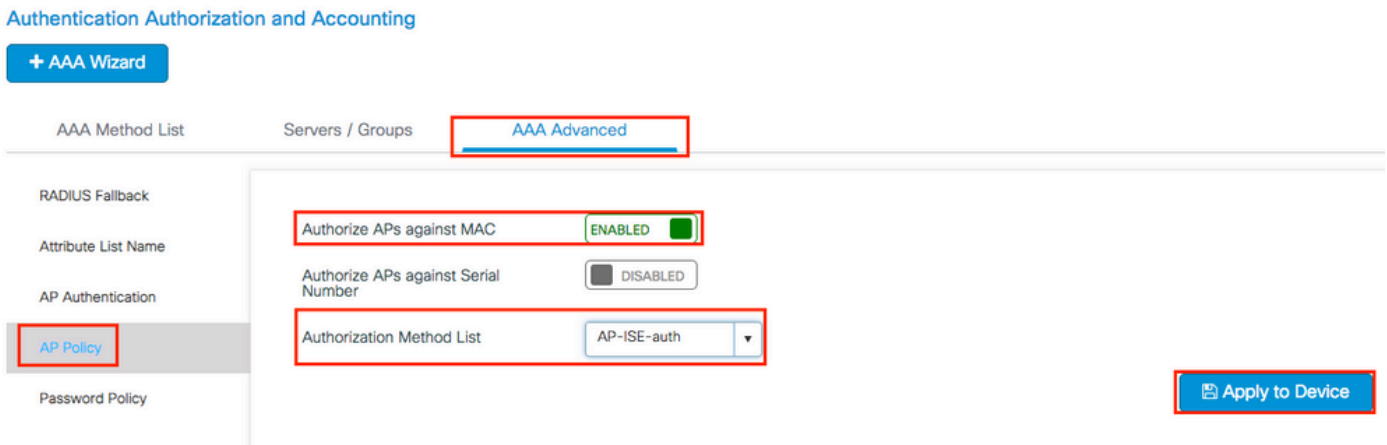
Navigate to **Configuration** > **Security** > **AAA** > **AAA Method List** > **Authorization** > **+ Add**.






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:

```
# 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. To add 9800 WLC to ISE:

Declare 9800 WLC on ISE

Choose to configure the AP´s MAC address based on authentication with the required steps:

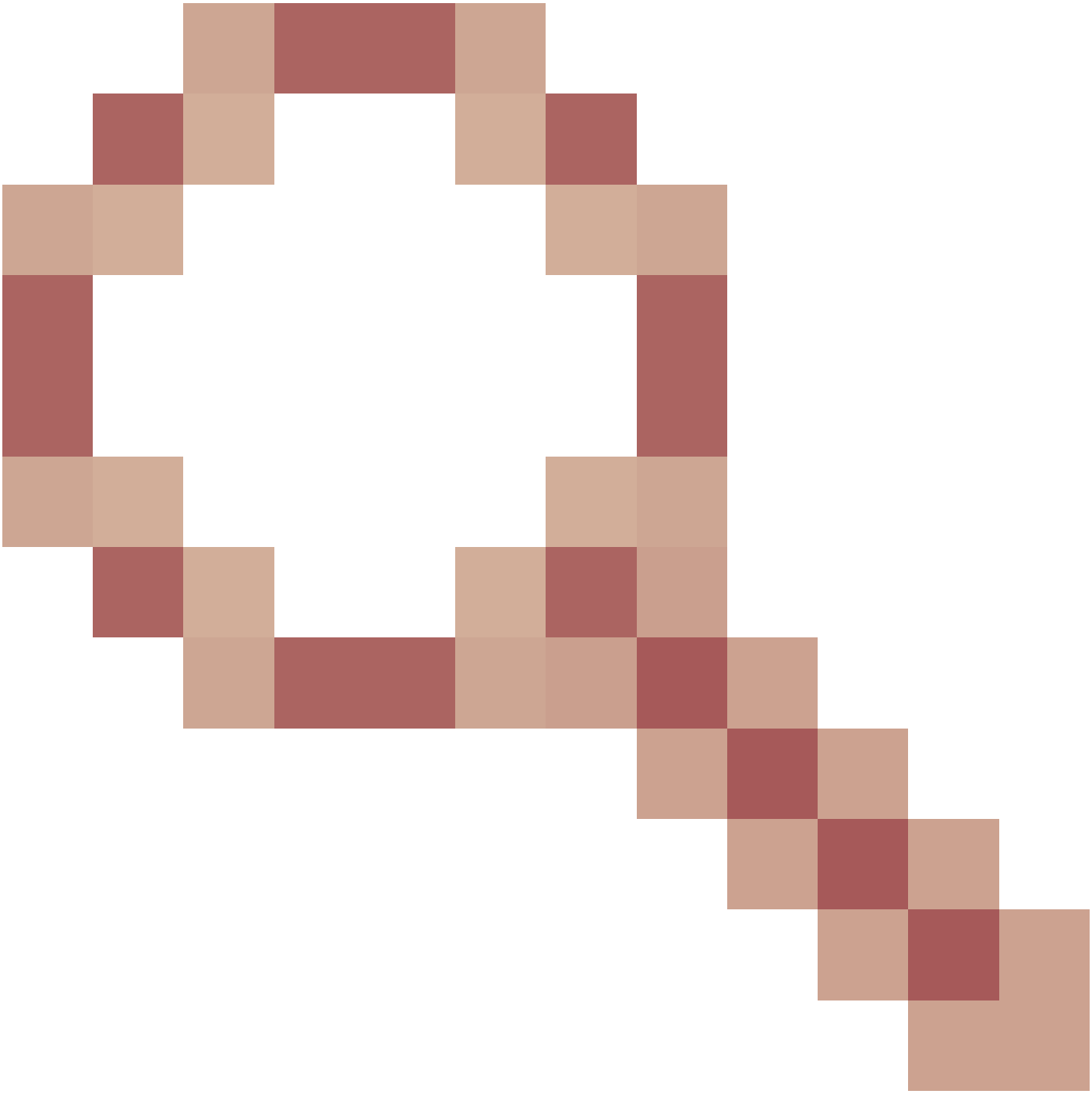
Configure USE to authenticate MAC address as endpoints

Configure ISE to authenticate MAC address as username/password

**Configure ISE to Authenticate MAC Address as Endpoints**

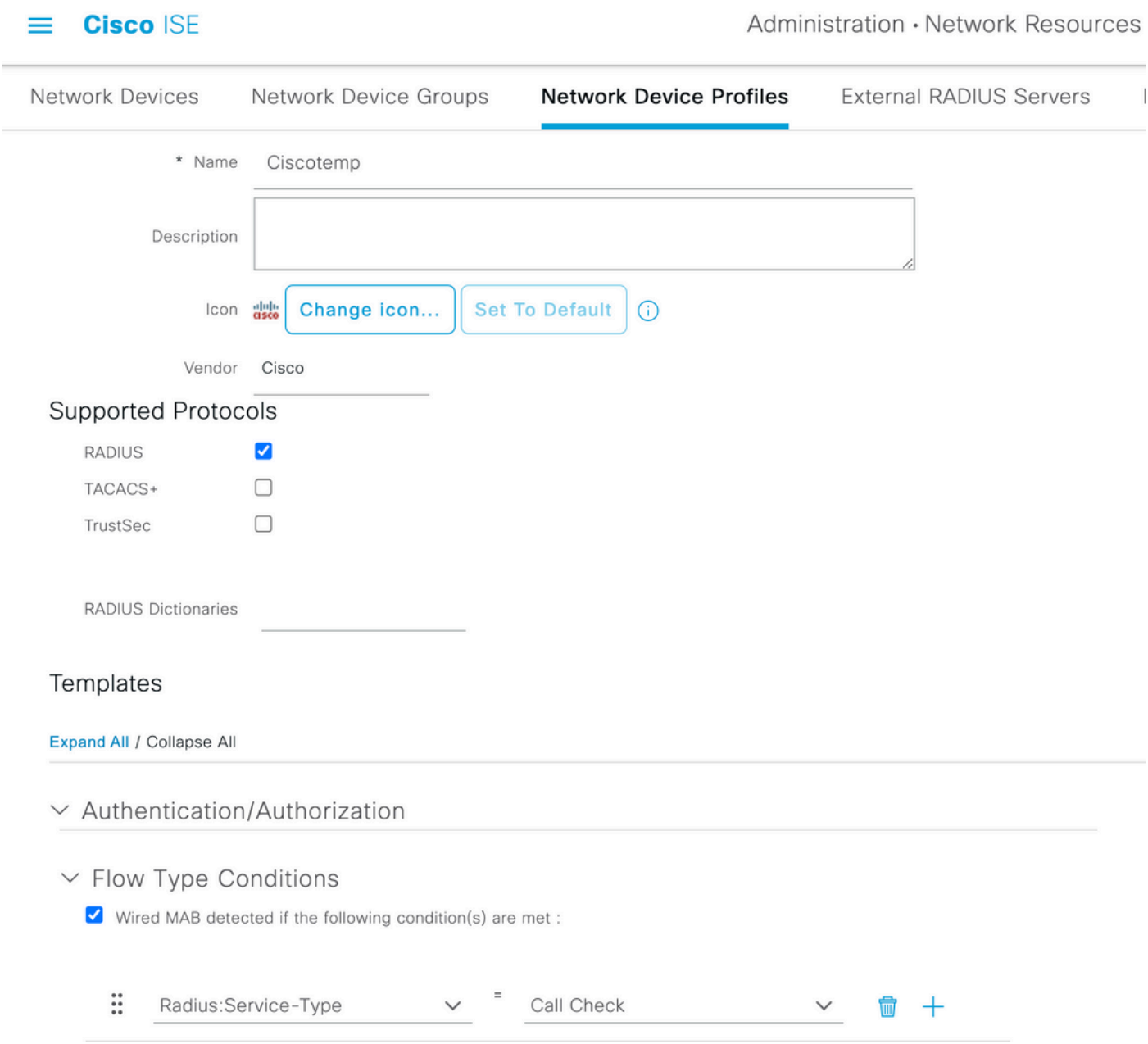
Step 2. (Optional) Create an identity group for Access Points.

Because the 9800 does not send the NAS-port-Type attribute with AP authorization (Cisco bug IDCSCvy74904),



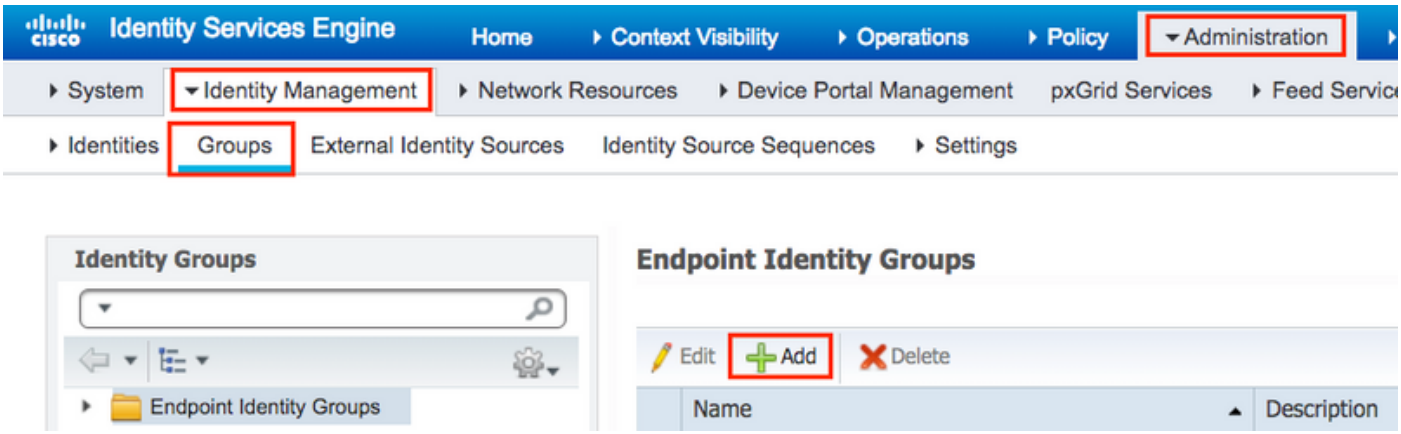
ISE does not recognize an AP authorization as a MAB workflow. Therefore, it is not possible to authenticate an AP if the MAC address of the AP is placed in the endpoint list unless you modify the MAB workflows to not require the NAS-PORT-type attribute on ISE.

Navigate to **Administrator > Network device profile** and create a new device profile. Enable **RADIUS**, and add service-type=call-check for Wired MAB. You can copy the rest from the Cisco original profile. The idea is to have no nas-port-type condition for the Wired MAB.

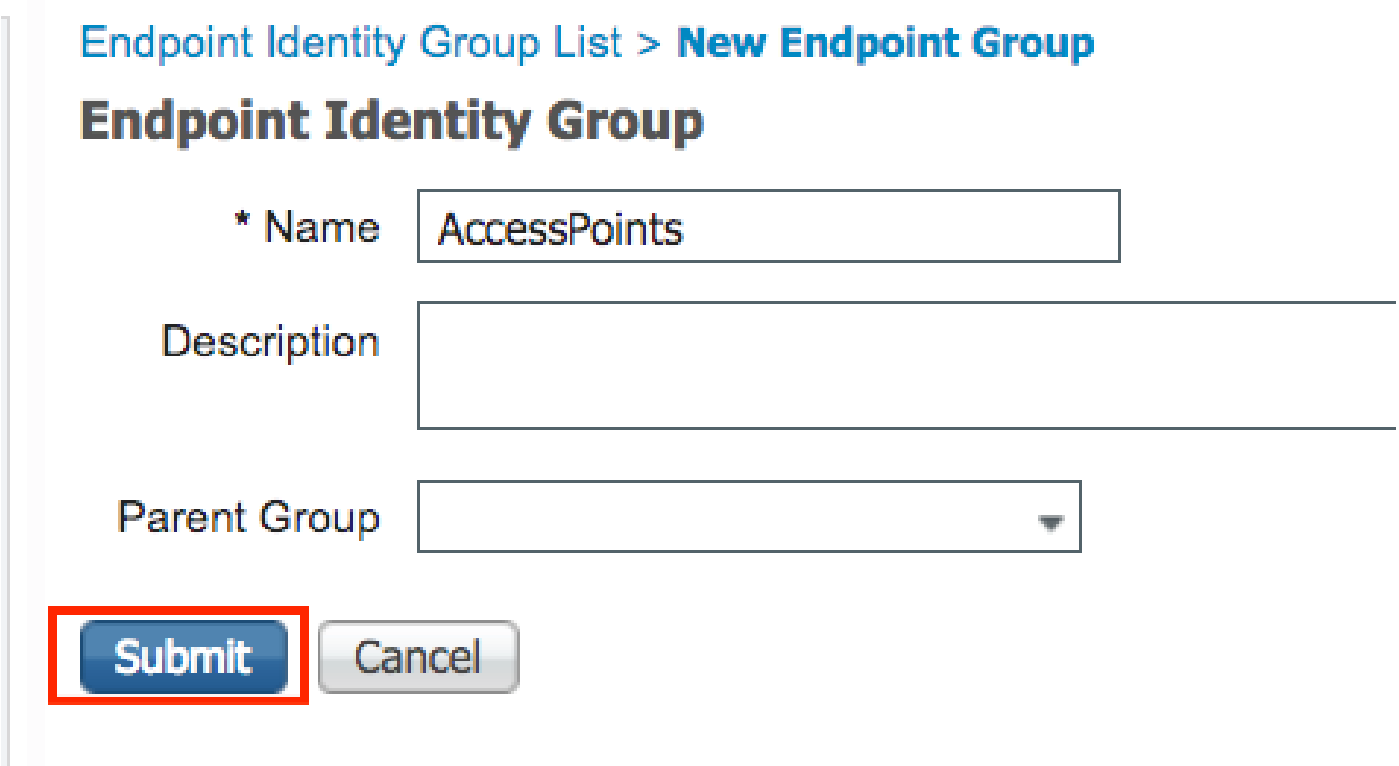



Go back to your network device entry for the 9800 and set its profile to the newly created device profile.

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

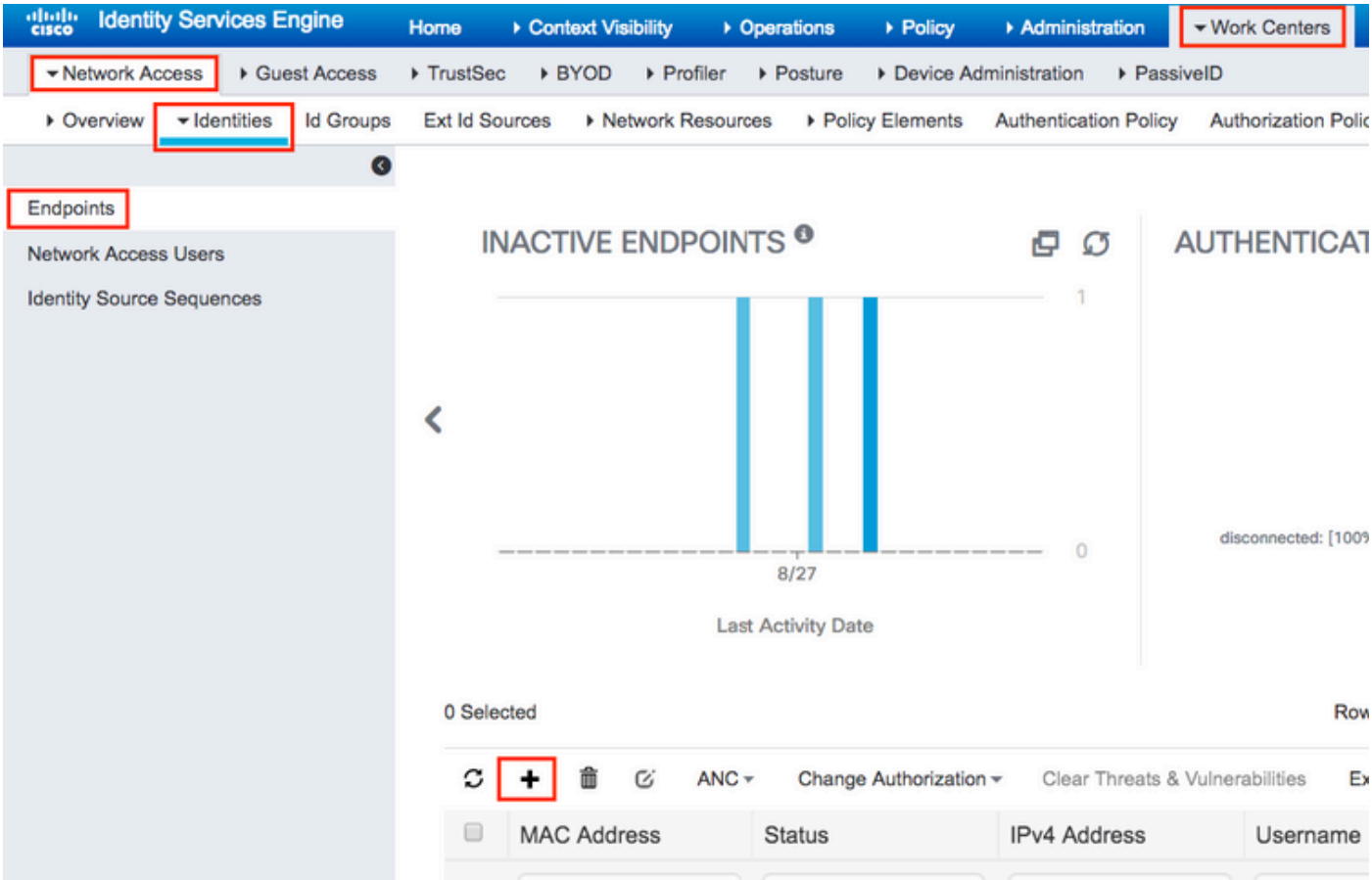



Choose a name and click **Submit**.

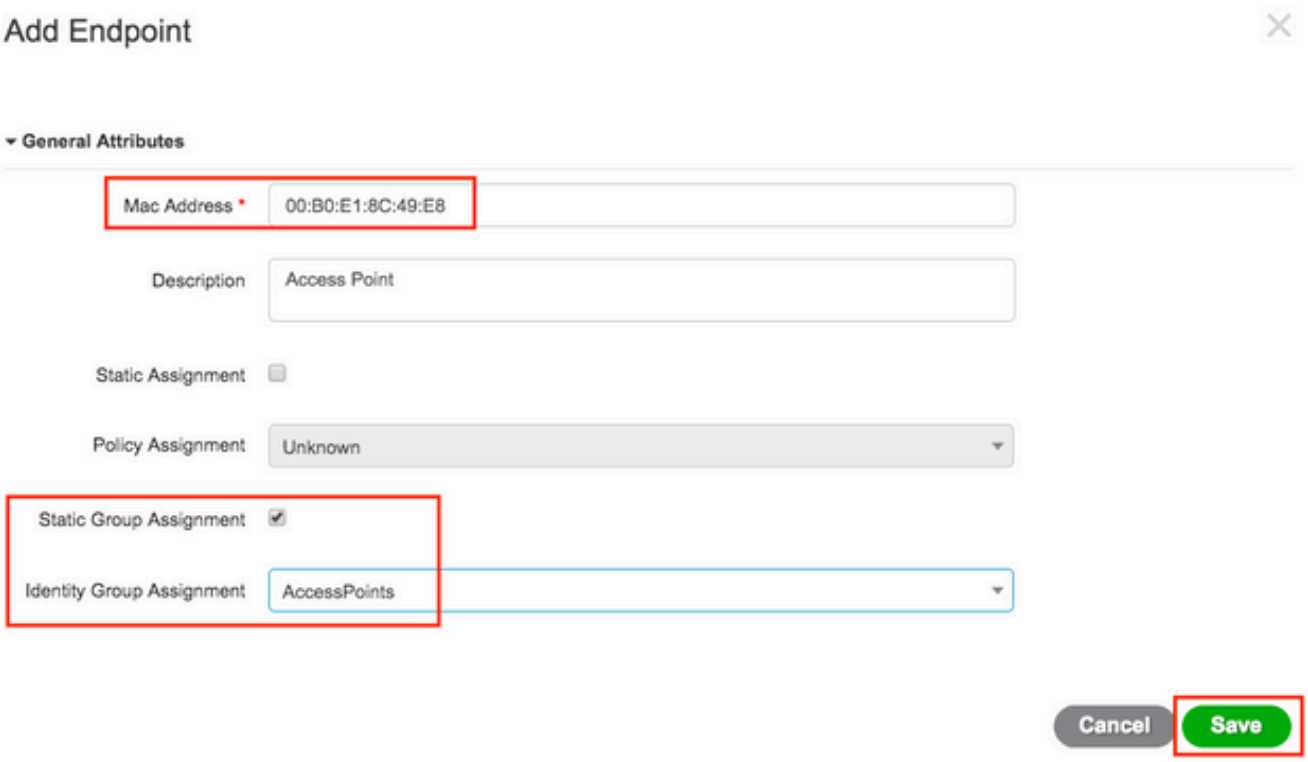



Step 3. Add the AP ethernet MAC address to its endpoint identity group.

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

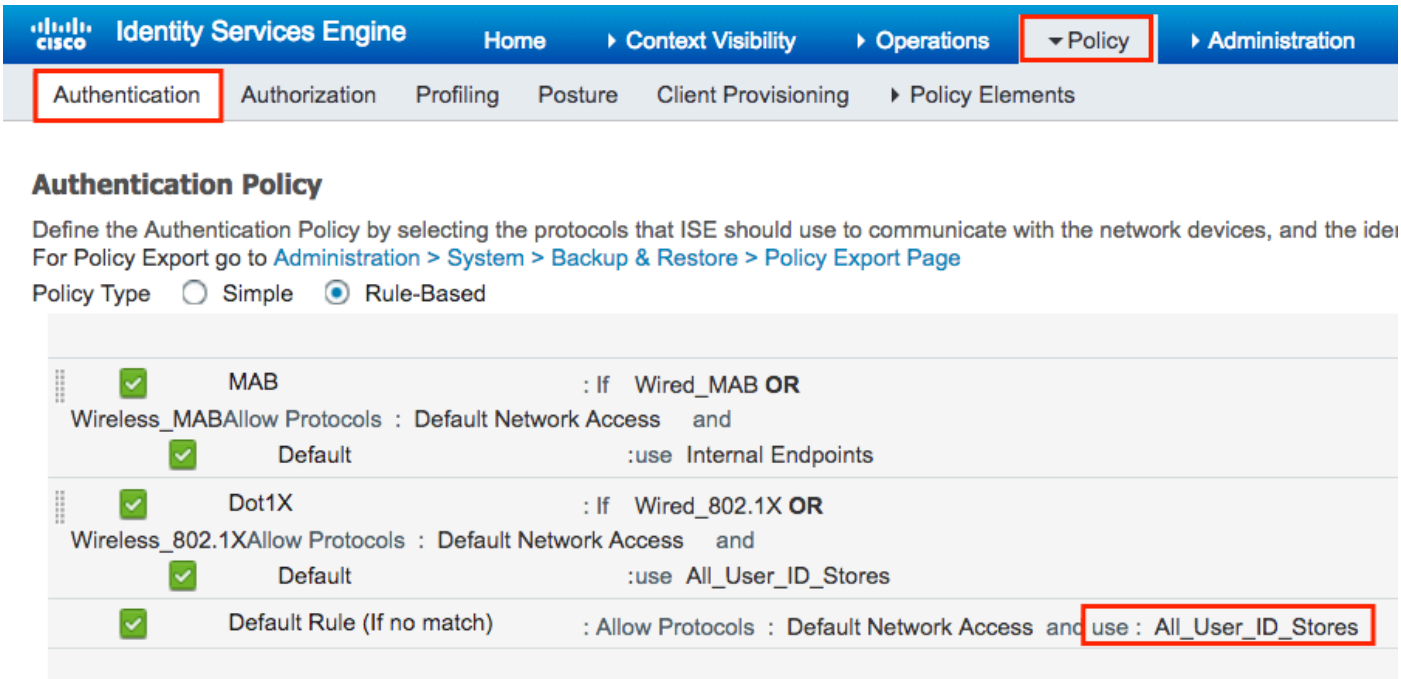



Enter the needed information.

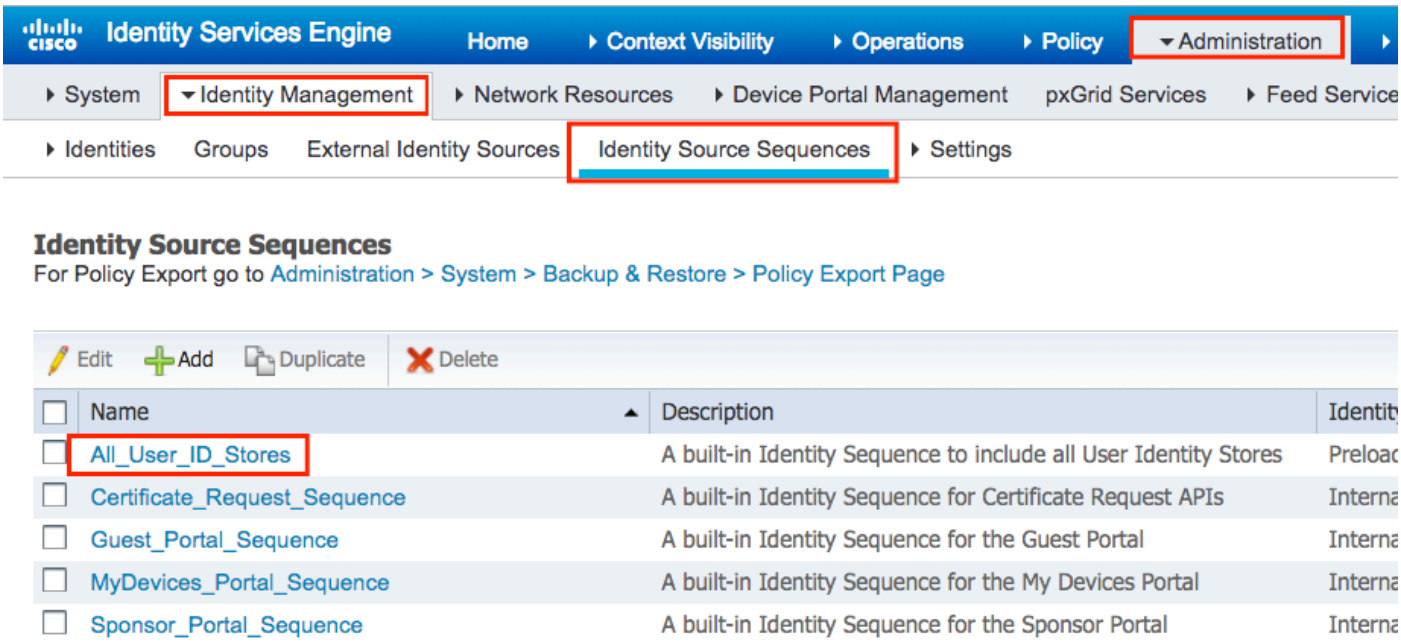



Step 4. Verify the identity store used on your default authentication rule that contains the internal endpoints.

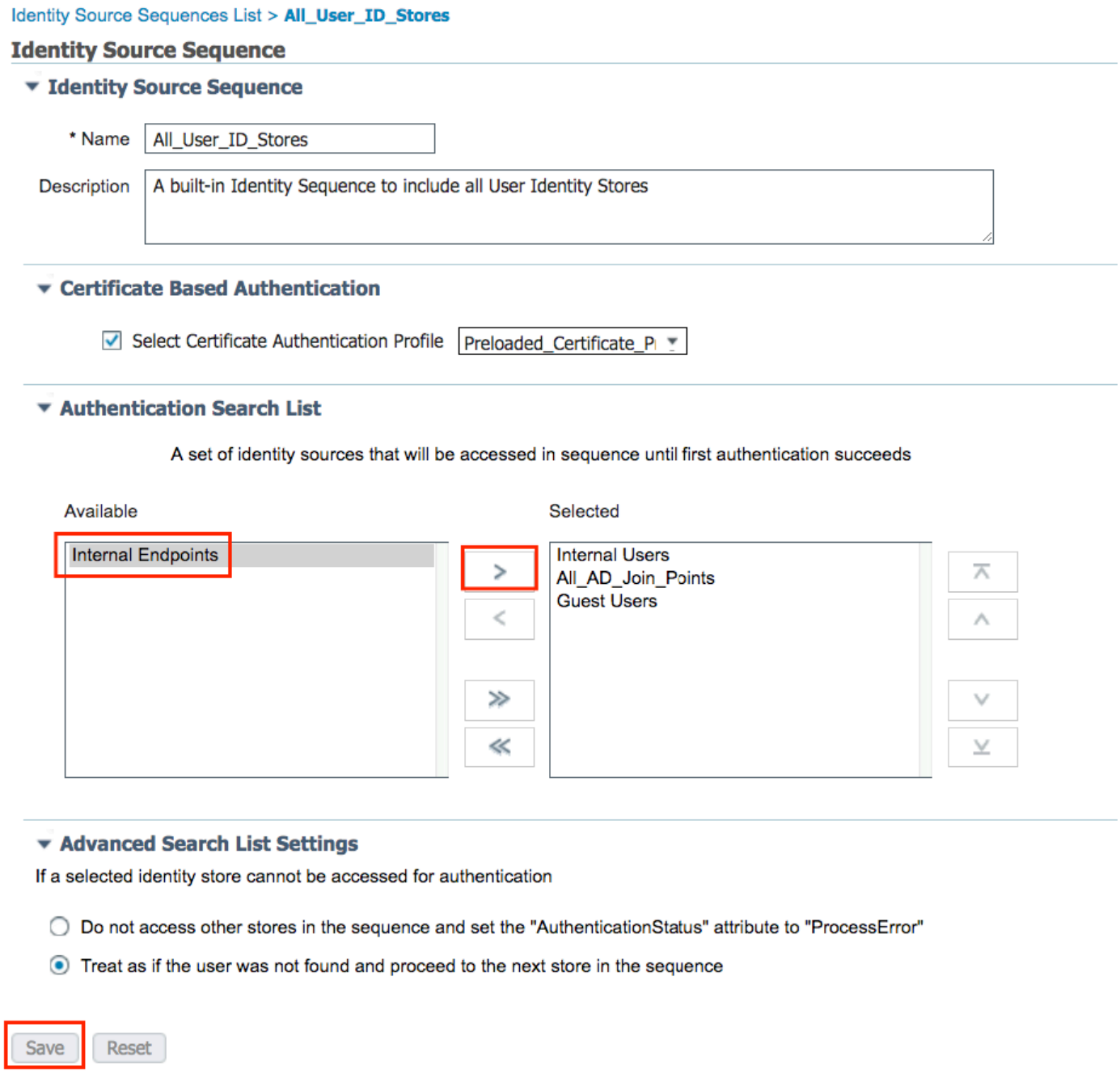
A. Navigate to **Policy** > **Authentication** and take note of the Identity store.




B. Navigate to **Administration** > **Identity Management** > **Identity Source Sequences** > **Identity Name**.




C. Ensure Internal Endpoints belong to it. Otherwise, add them.



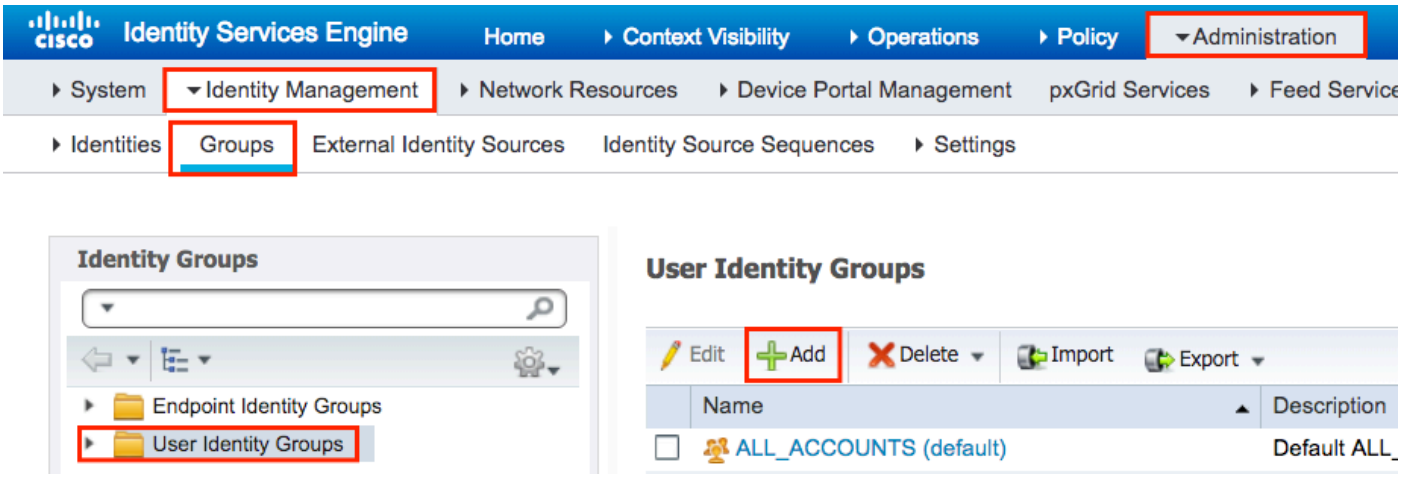

## Configure ISE to Authenticate MAC Address as Username/Password

This method is not advised as it requires lower password policies to allow the same password as the username.

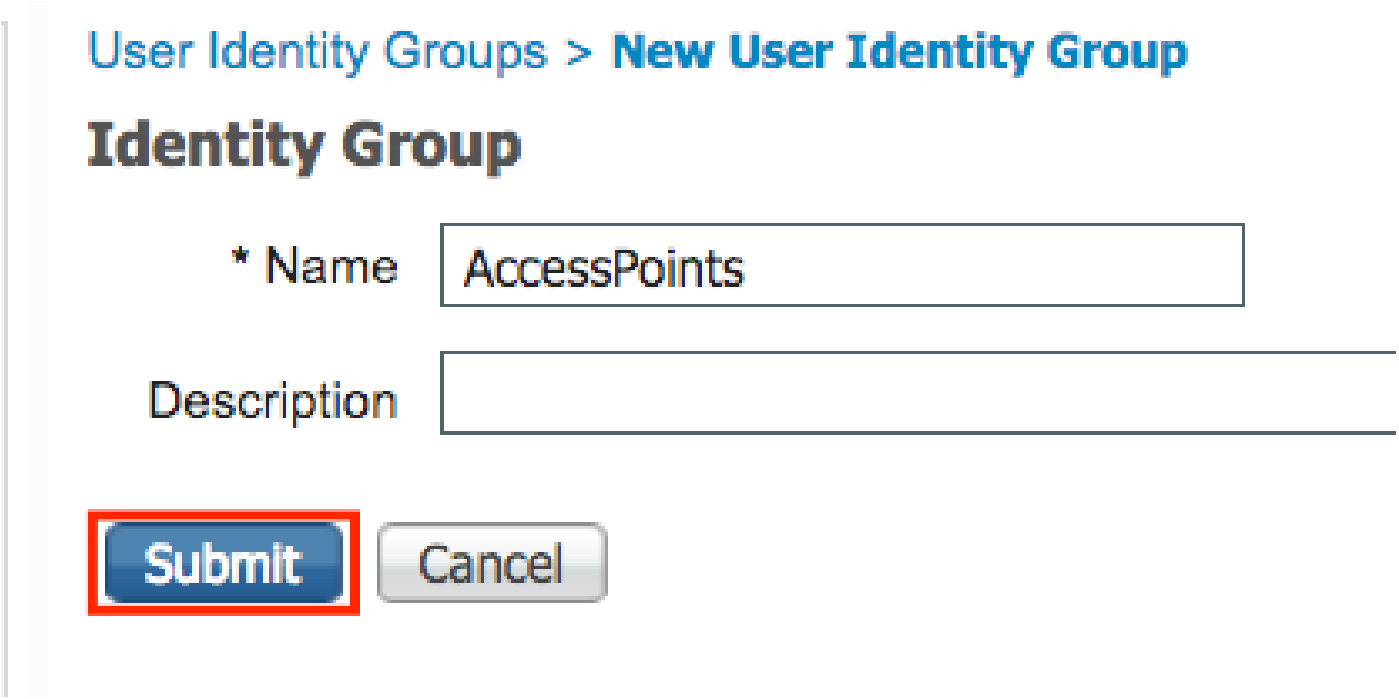
It can, however, be a workaround in case you cannot modify your Network device profile.

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

Navigate to **Administration** > **Identity Management** > **Groups** > **User Identity Groups** > **+ Add**.

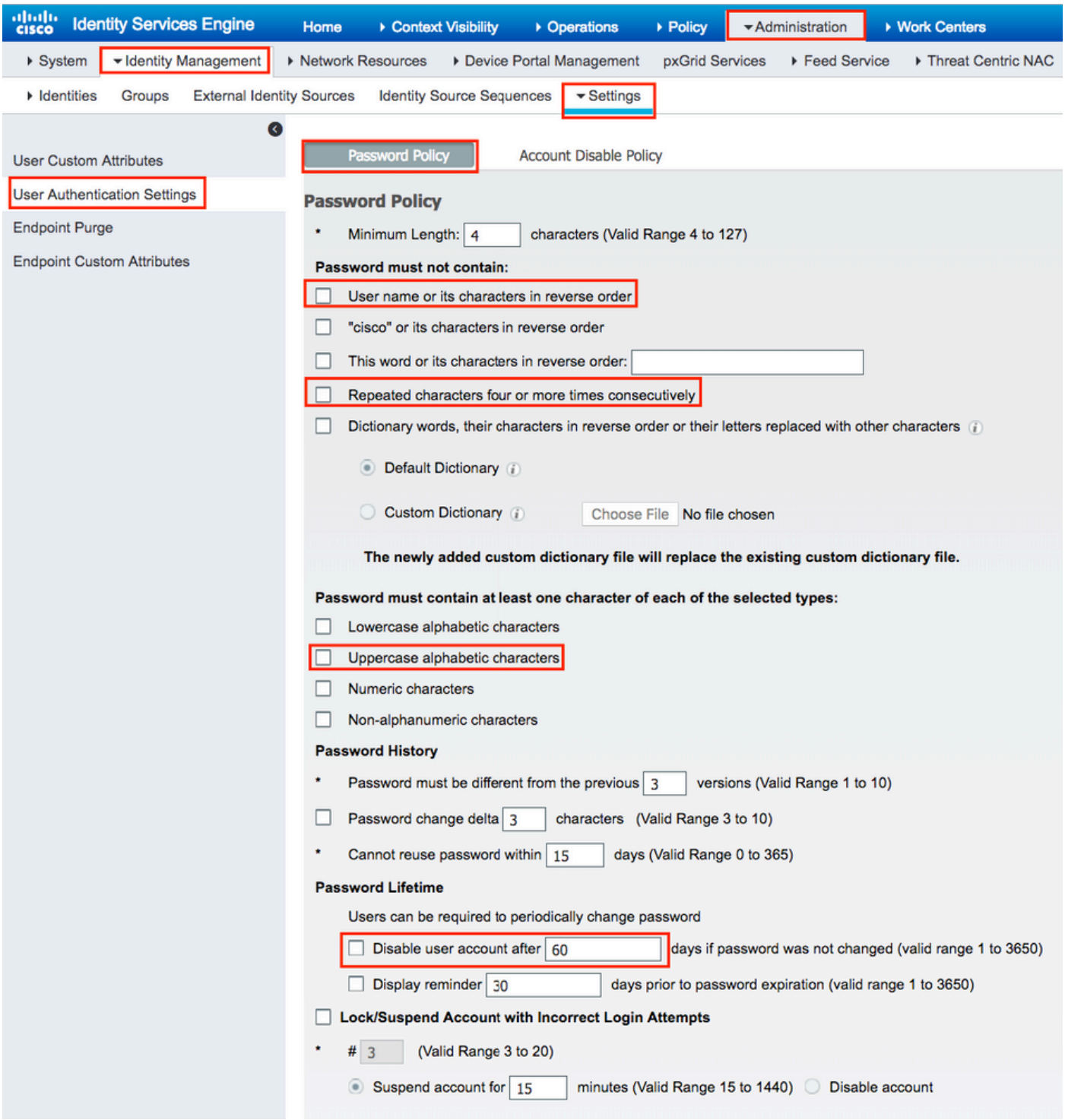



Choose a name and click **Submit**.




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

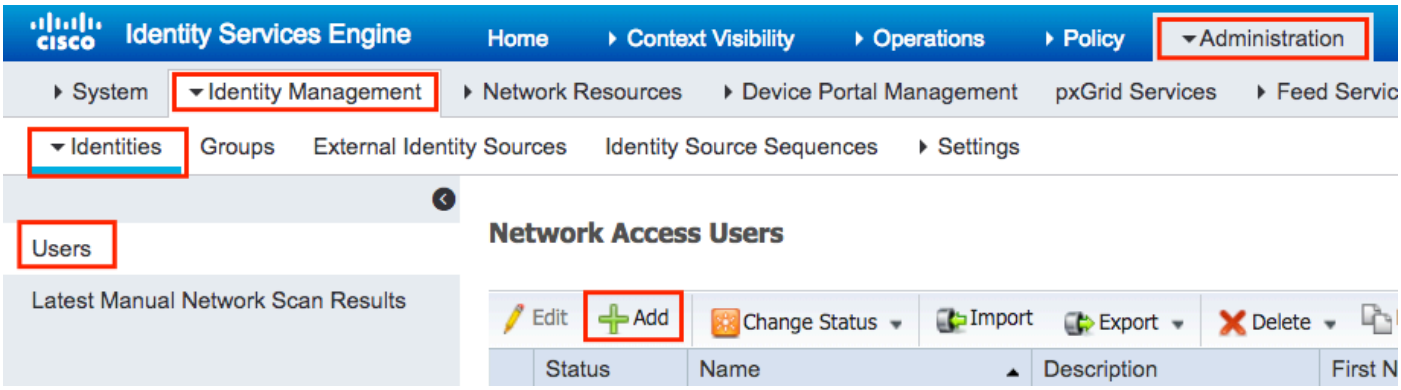
Navigate to **Administration > Identity Management > Settings > User Authentication Settings > Password Policy** and ensure that at least these options are disabled:



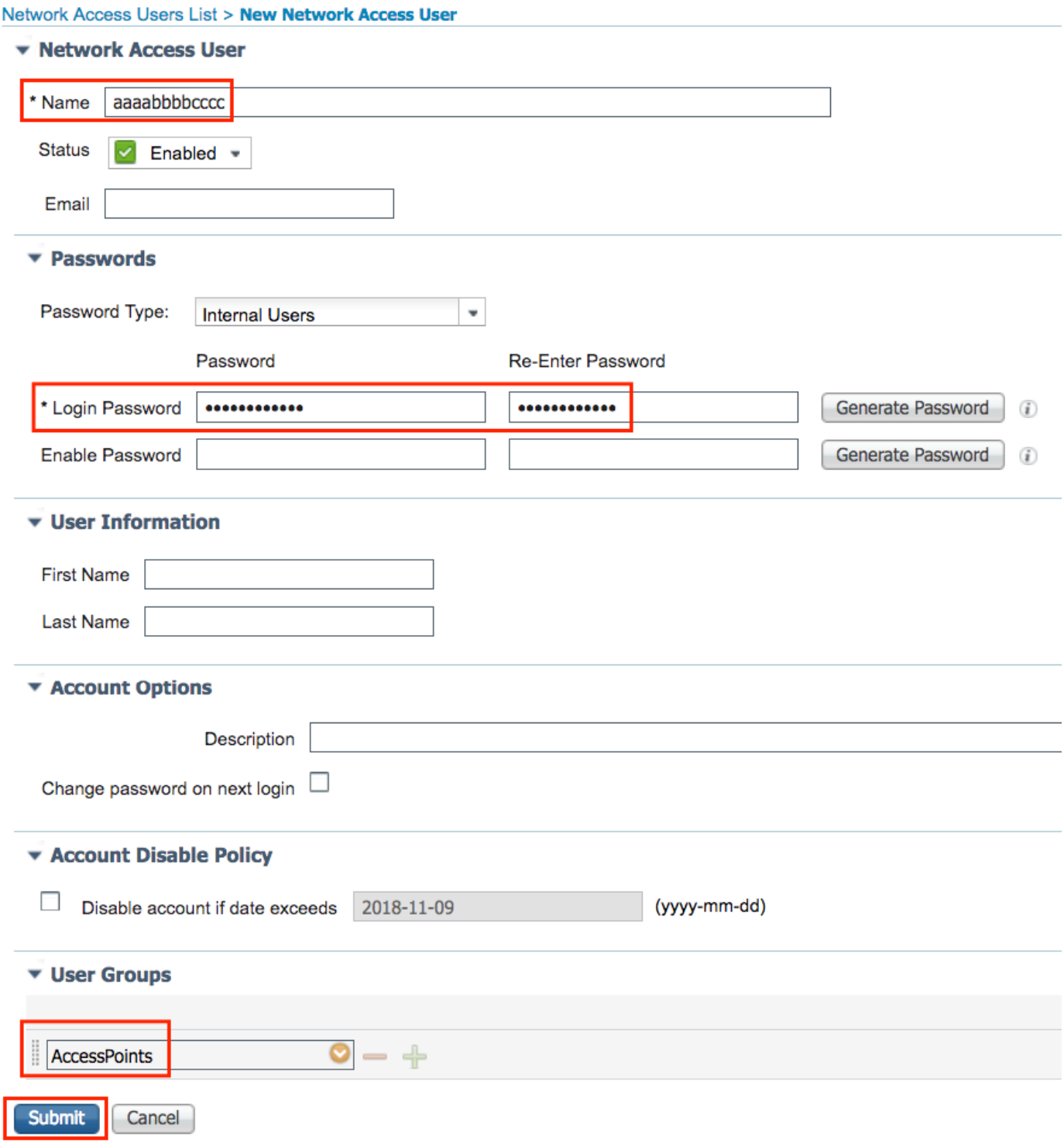

**Note**: You can also want to disable the option **Disable user account after** XX **days if password was not changed.**As this is a MAC address, the password never changes.

Step 4. Add the AP ethernet MAC address.

Navigate to **Administration** > **Identity Management** > **Identities** > **Users** > **+ Add**.



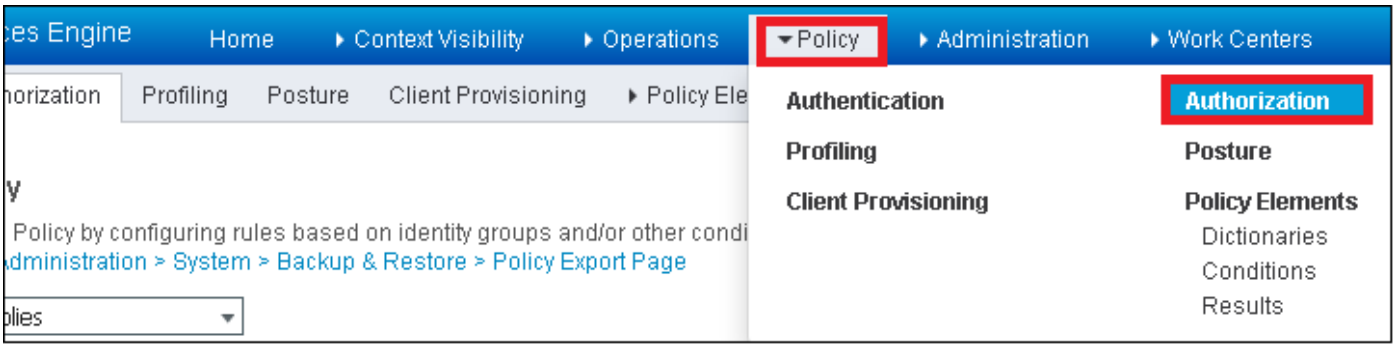

Enter the needed information.



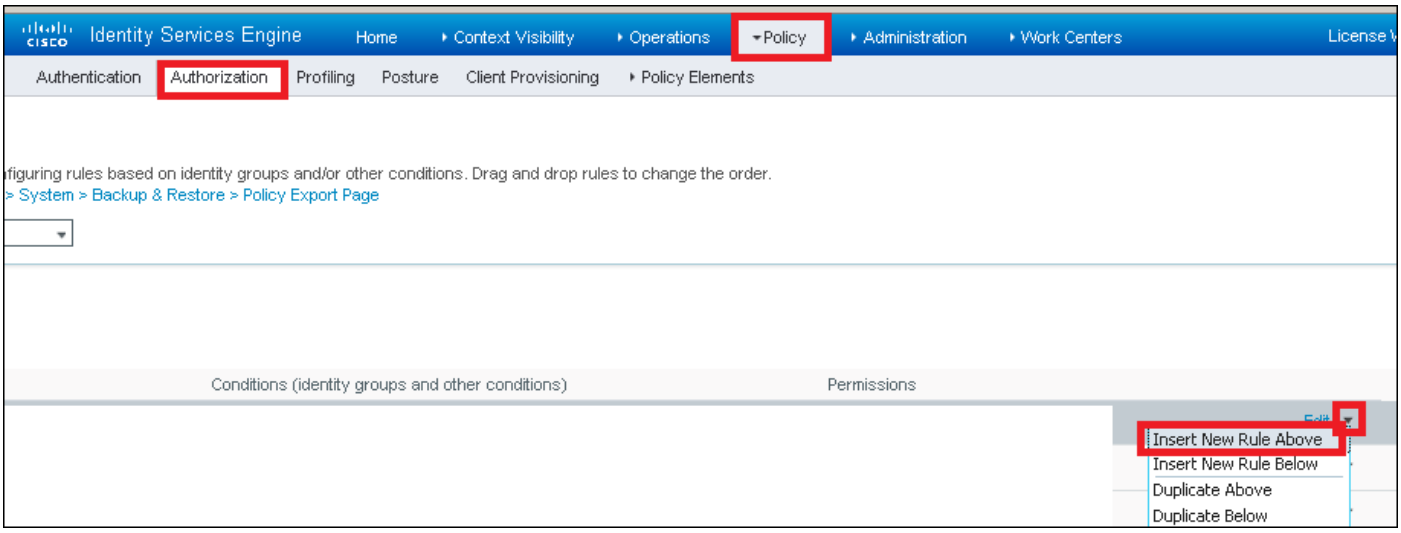

**Note**: **Name** and **Login Password**fields must be the ethernet MAC address of the AP, all lower case and no separators.

### Authorization Policy to Authenticate APs

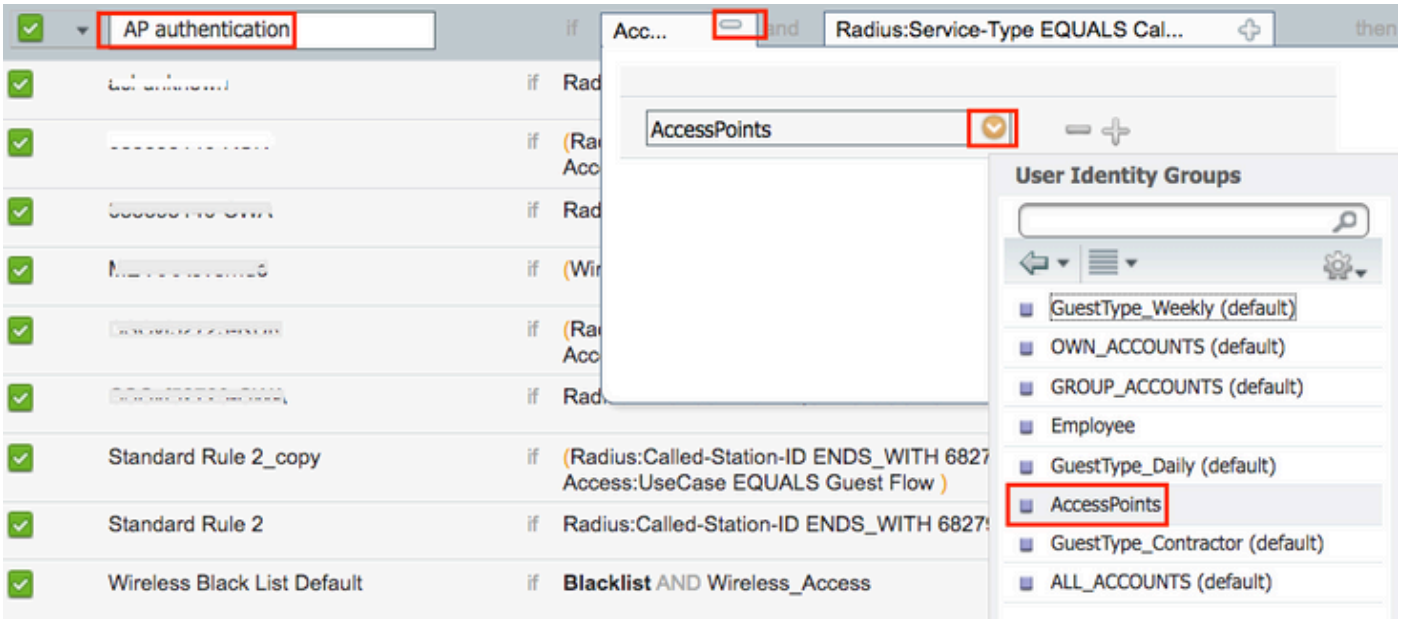
Navigate to**Policy** > **Authorization** as shown in the image.



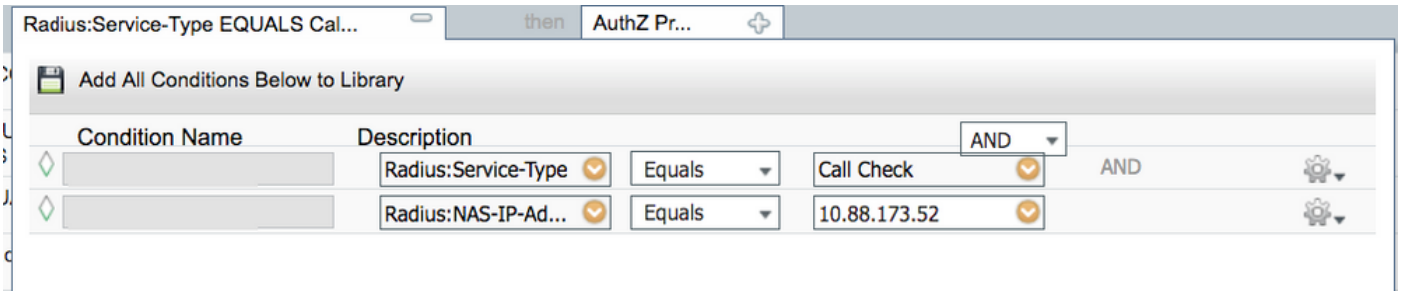

Insert a new rule as shown in the image.



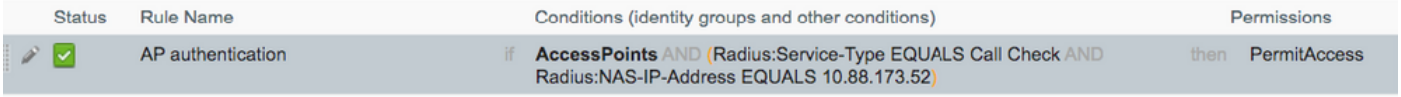

First, select a name for the rule and the Identity group where the Access Point is stored (AccessPoints). Select **User Identity Groups** if you decided to authenticate the MAC address as username password or **Endpoint Identity Groups** if you choose to authenticate the AP MAC address as endpoints.




After that, select other conditions that cause the authorization process to fall under 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 do not lose their association. If, however, after authorization list is enabled, they lose communication with the controller and attempt to rejoin, 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 rejoin the controller.

# Verify

Verify if 9800 WLC has enabled the AP authentication list.

<#root>

```
# show ap auth-list
```

```
Authorize APs against MAC : Disabled
Authorize APs against Serial Num : Enabled
Authorization Method List : <auth-list-name>
```

Verify radius configuration:

<#root>

```
#
show run aaa
```

# Troubleshoot

WLC 9800 provides ALWAYS-ON trace 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**: Volume of logs generated vary retroactively from a few hours to several days.

To view the traces that 9800 WLC collected by default, you can connect via SSH/Telnet to the 9800 WLC through these steps. (Ensure that you log the session to a text file).

Step 1. Check the controller 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 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 Trace Configs:


Packet Infra debugs:

Ip Address                                                Port
-------------------------------------------------------|----------
```

**Note**: If you see any condition listed, it means the traces are 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. Assume the tested MAC address 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.bbbb.cccc> | <a.b.c.d> } to-file always-on-
```

You can either display the content in 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 the Radio Active (RA) trace, which provides debug level traces for all processes that interact with  the specified condition (client mac address in this case).

Step 5. Ensure there are no debug conditions enabled.

```
# clear platform condition all
```

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

This commands 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 in the 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:

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

Copy the file to an external server:

```
# copy bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log tftp://a.b.c.
```

Display the content:

```
# more bootflash:ra_trace_MAC_aaaabbbbcccc_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 you only take 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 r
```

---

**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.

## References

Join mesh APs to 9800 WLC