配置验证并排除Mac过滤器上的Web身份验证故障

目录

简介

本文档介绍如何使用ISE进行外部身份验证，在“Mac过滤器故障”功能上配置、故障排除和验证本地网络身份验证。

先决条件

为MAC身份验证配置ISE

在ISE/Active Directory上配置的有效用户凭证

要求

Cisco 建议您了解以下主题：

基本了解如何在控制器Web UI中导航

策略、WLAN配置文件和策略标记配置

ISE上的服务策略配置

使用的组件

9800 WLC版本17.12.2

C9120 AXI AP

9300 交换机

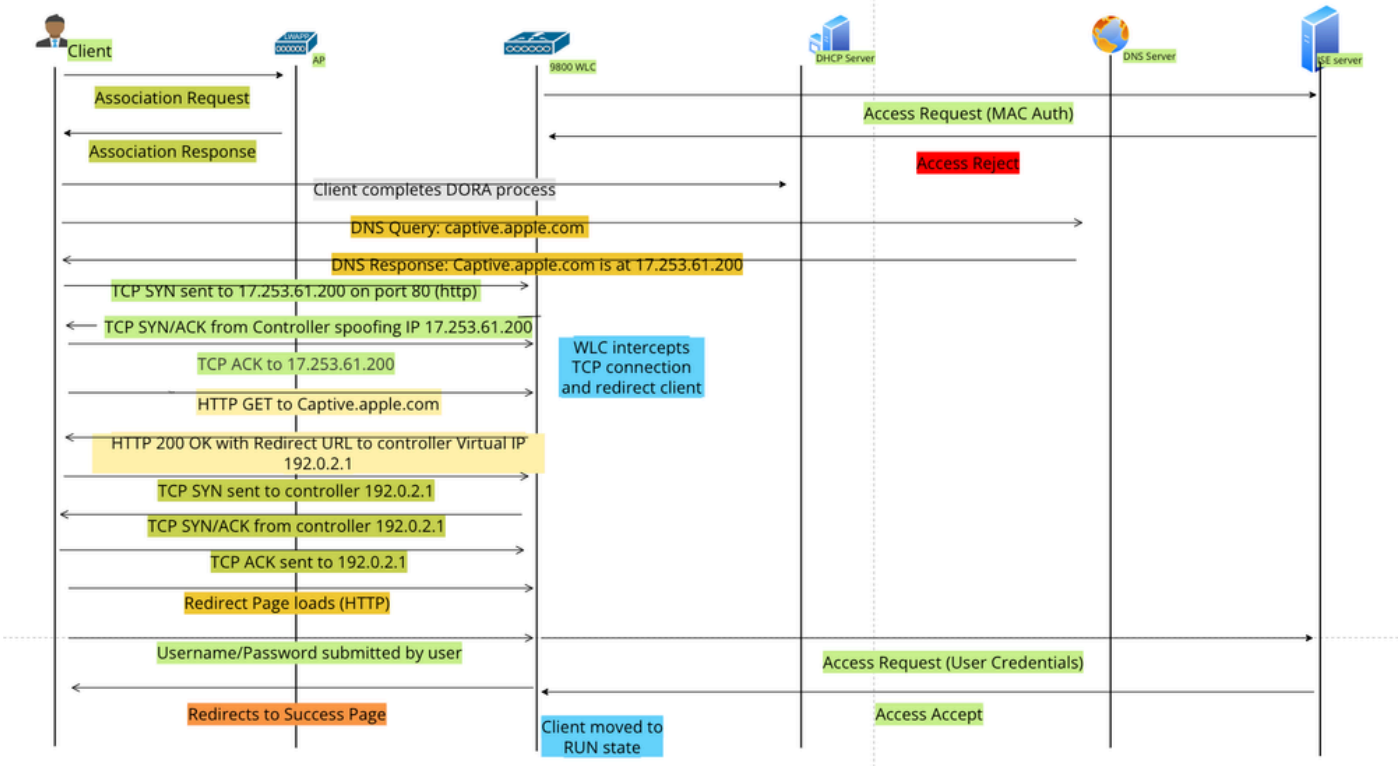
ISE版本3.1.0.518

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

在同时使用MAC身份验证和Web身份验证的WLAN环境中，Web身份验证“On Mac Failure Filter”（On Mac故障过滤器）功能可用作后退机制。

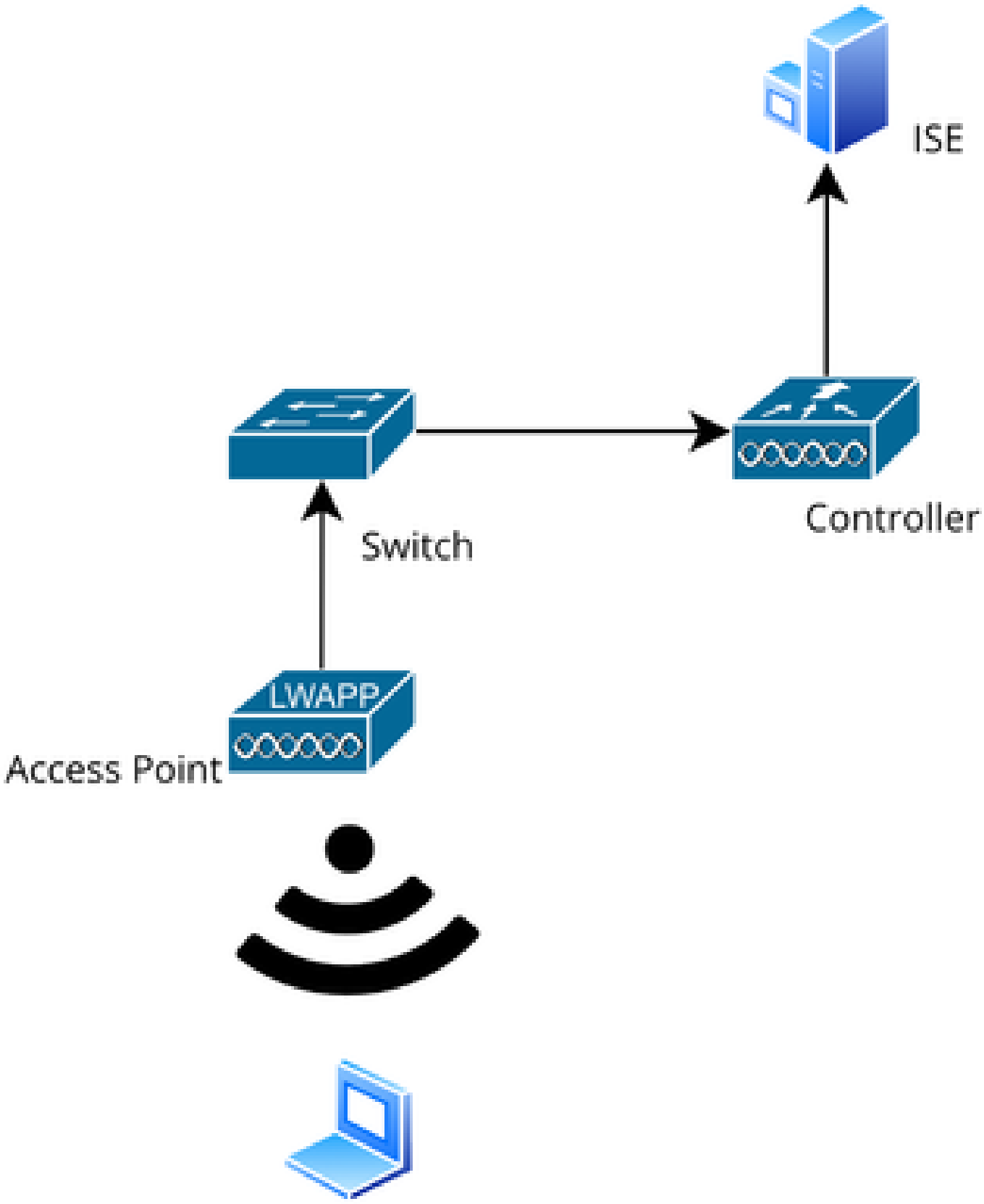
- 回退机制：当客户端尝试使用MAC过滤器针对外部RADIUS服务器(ISE)或本地服务器连接到WLAN且无法进行身份验证时，此功能会自动启动第3层Web身份验证。
- 身份验证成功：如果客户端通过MAC过滤器成功进行身份验证，则会绕过Web身份验证，从而允许客户端直接连接到WLAN。
- 避免取消关联：此功能有助于防止因MAC过滤器身份验证失败而导致取消关联。



Web身份验证流程

配置

网络图



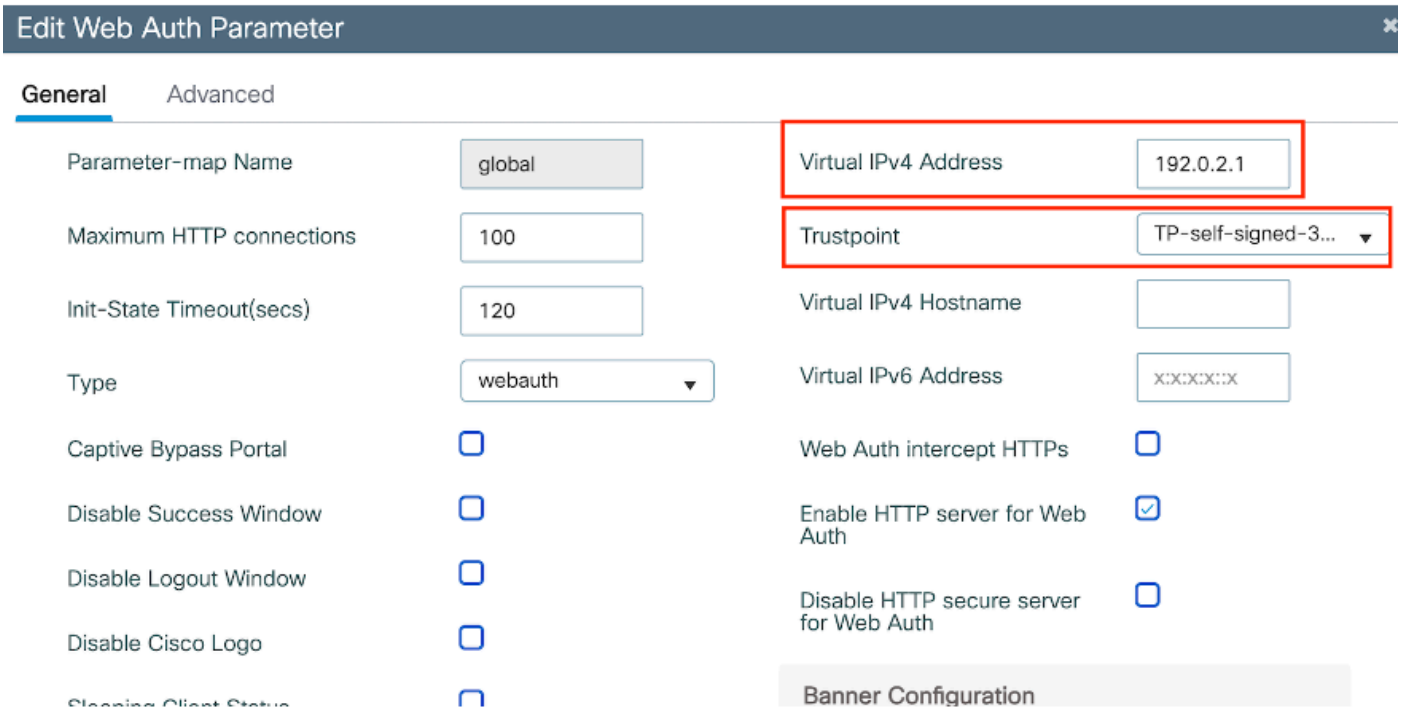
网络拓扑

配置

配置Web参数

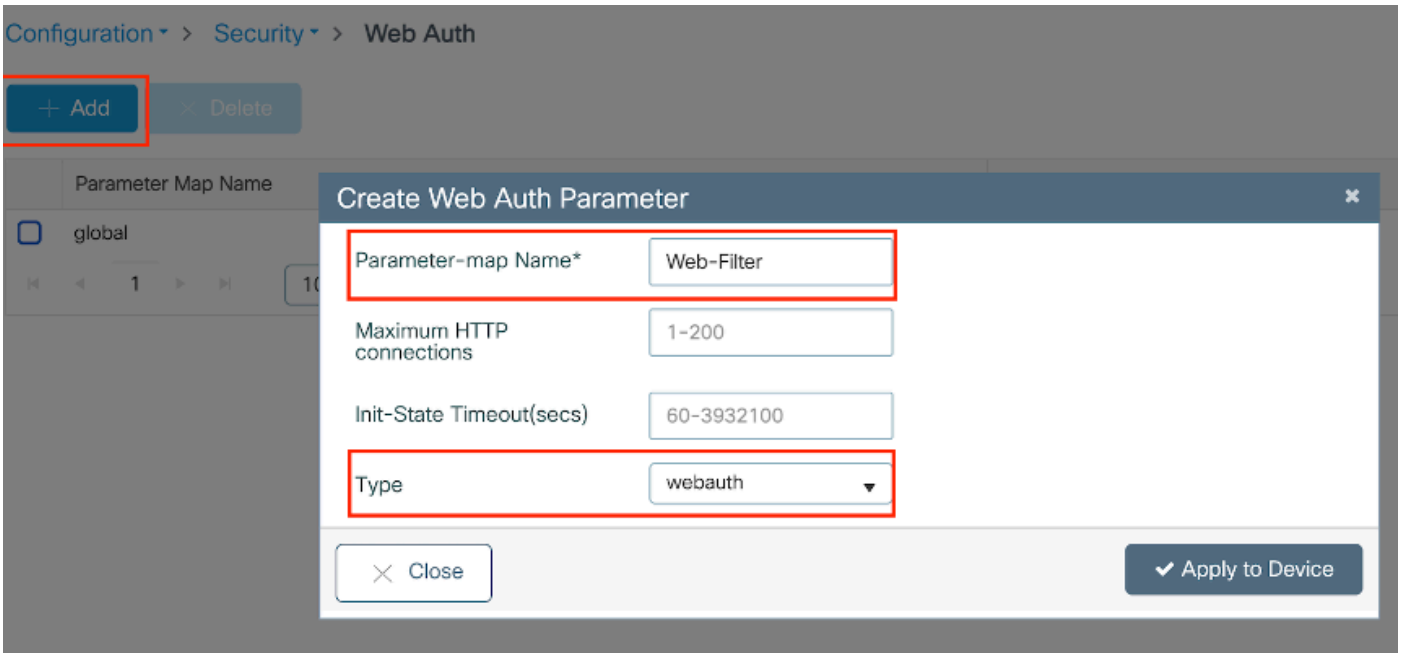
导航到配置>安全> Web身份验证并选择全局参数映射

从全局参数映射验证虚拟IP和信任点配置。所有自定义Web身份验证参数配置文件均从全局参数映射继承虚拟IP和信任点配置。



全局Web身份验证参数配置文件

第1步：选择“添加”(Add)创建自定义Web身份验证参数映射。输入配置文件名称，然后选择“Type”作为“Webauth”。



Web身份验证参数配置文件

如果客户端也获得IPv6地址，您还必须在参数映射中添加虚拟IPv6地址。使用文档范围2001：db8：：/32中的IP

如果您的客户端获得IPv6地址，他们很可能会尝试在V6而不是V4中获取HTTP网络身份验证重定向，这就是您还需要设置虚拟IPv6的原因。

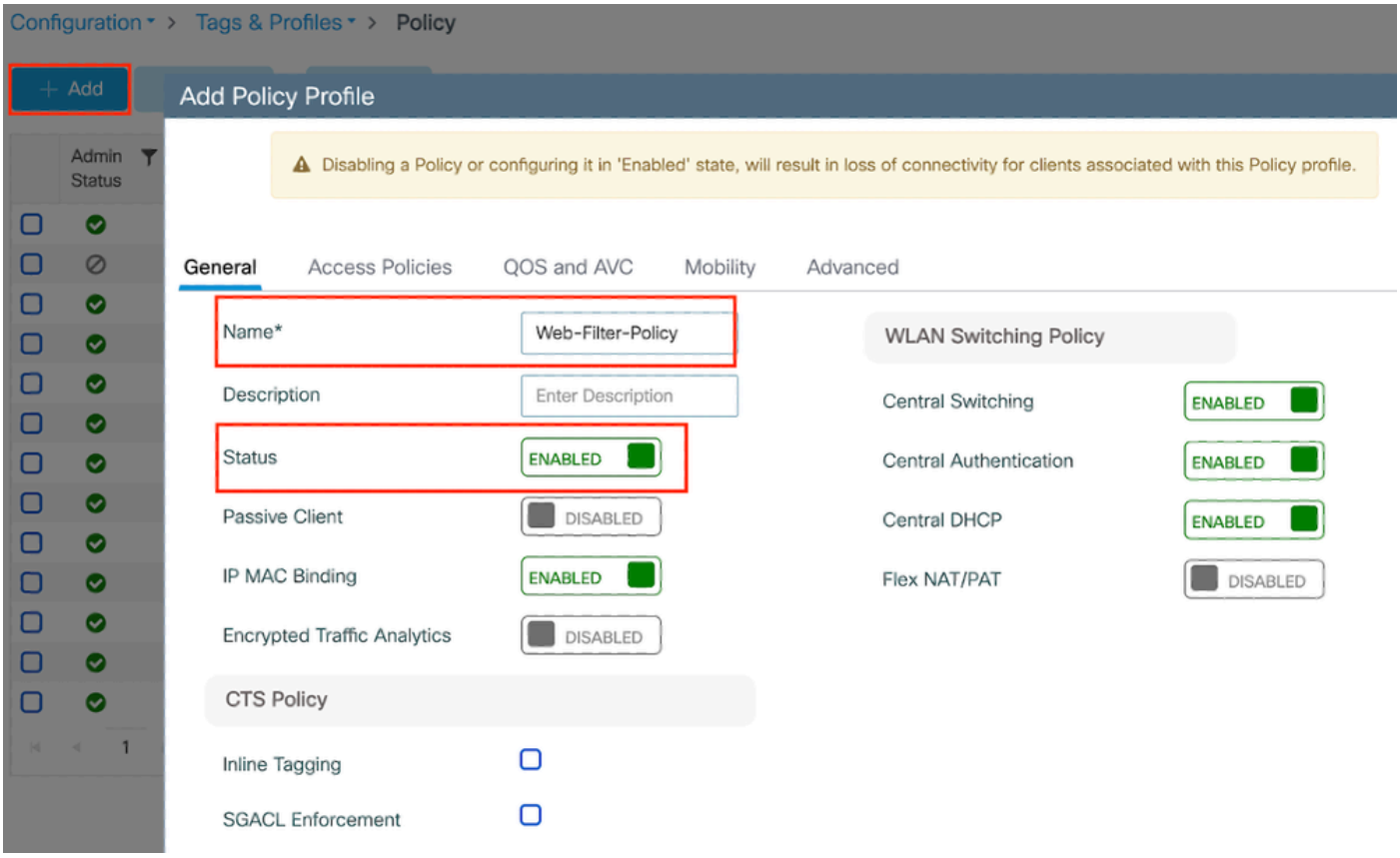
CLI 配置：

```
parameter-map type webauth Web-Filter
 type webauth
```

配置策略配置文件

第1步：创建策略配置文件

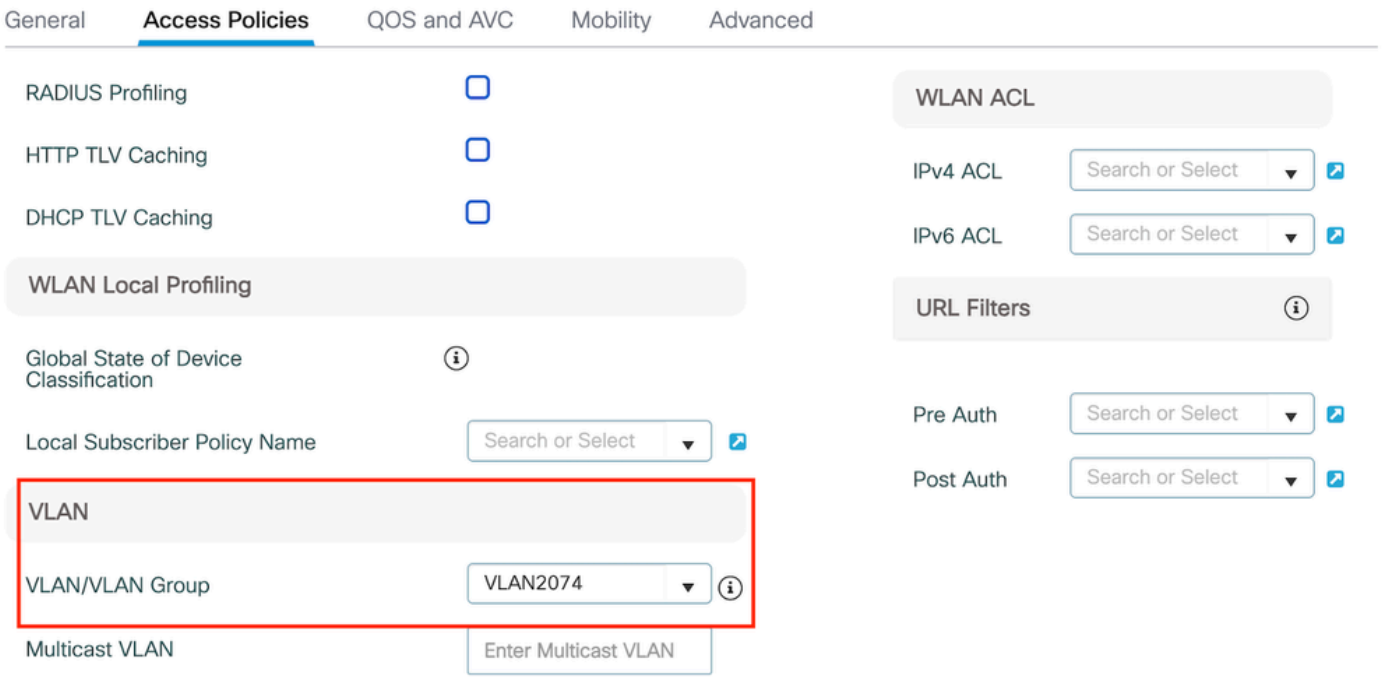
导航到Configuration > Tags & Profiles > Policy。选择“添加”。在“常规”选项卡中，指定配置文件的名称并启用状态切换。



策略配置文件

步骤2：

在Access Policies（访问策略）选项卡下，从VLAN部分下拉列表中选择客户端VLAN。



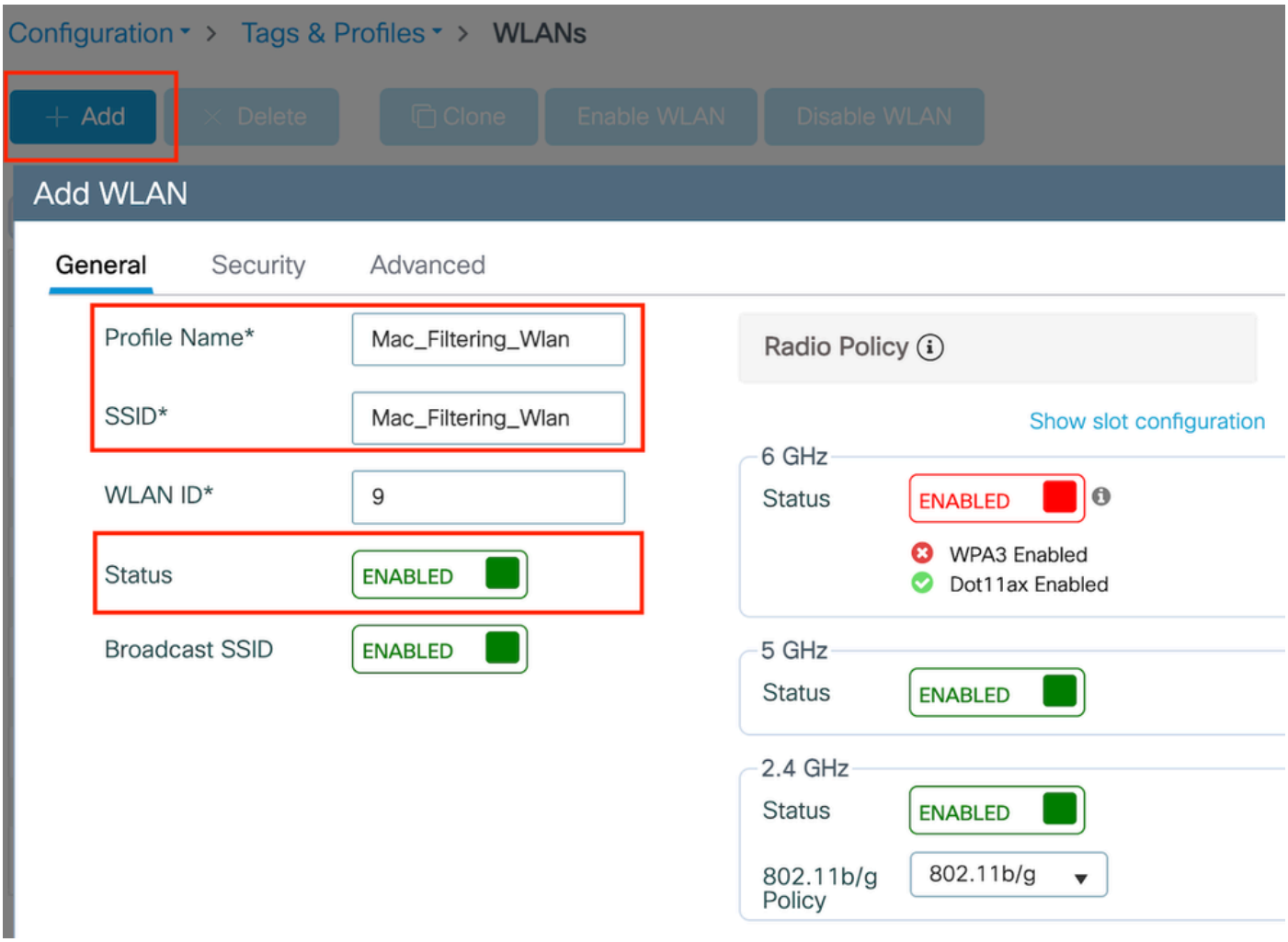
Access Policy选项卡

CLI 配置：

```
wireless profile policy Web-Filter-Policy
 vlan VLAN2074
 no shutdown
```

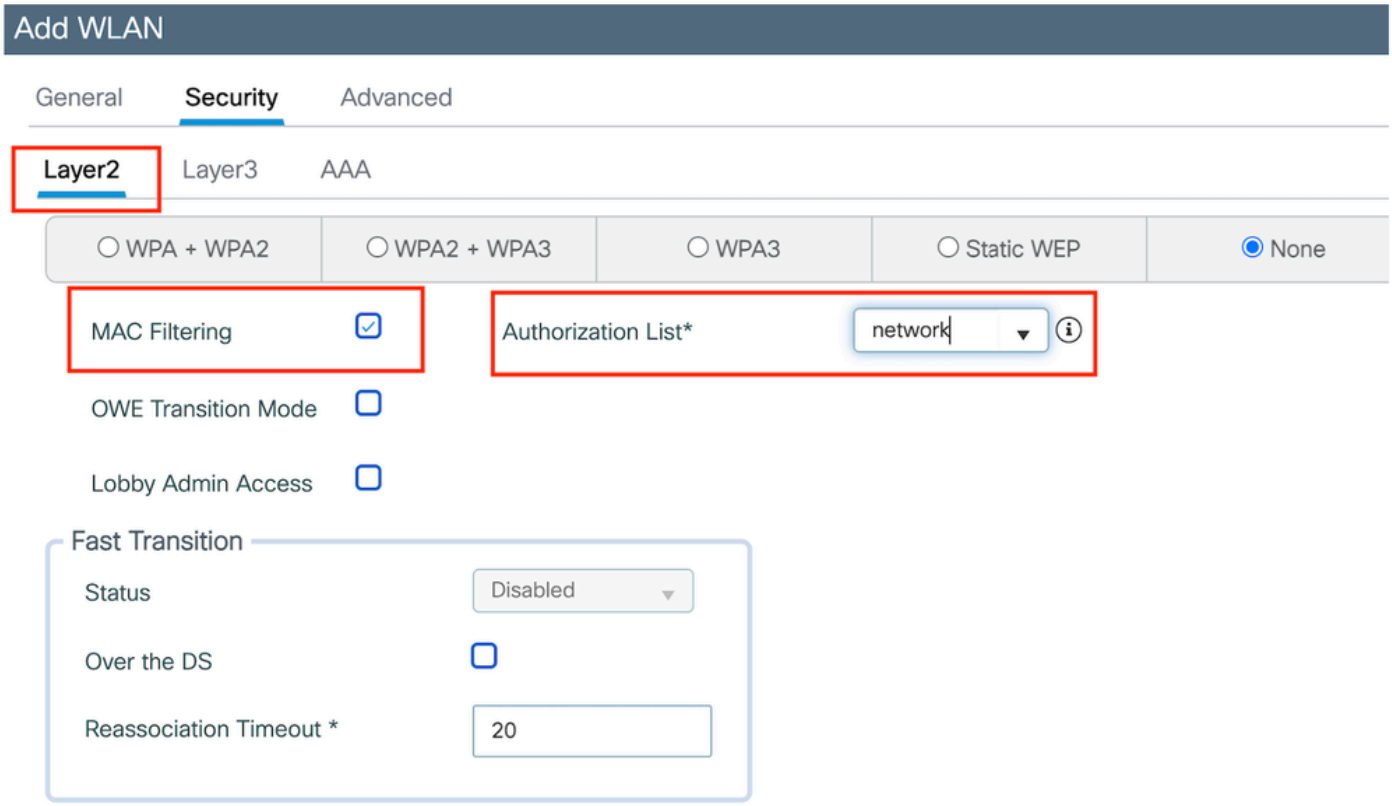
配置WLAN配置文件

第1步：导航至配置>标签和配置文件> WLAN。选择“添加”以创建新配置文件。定义配置文件名称和SSID名称，并启用状态字段。



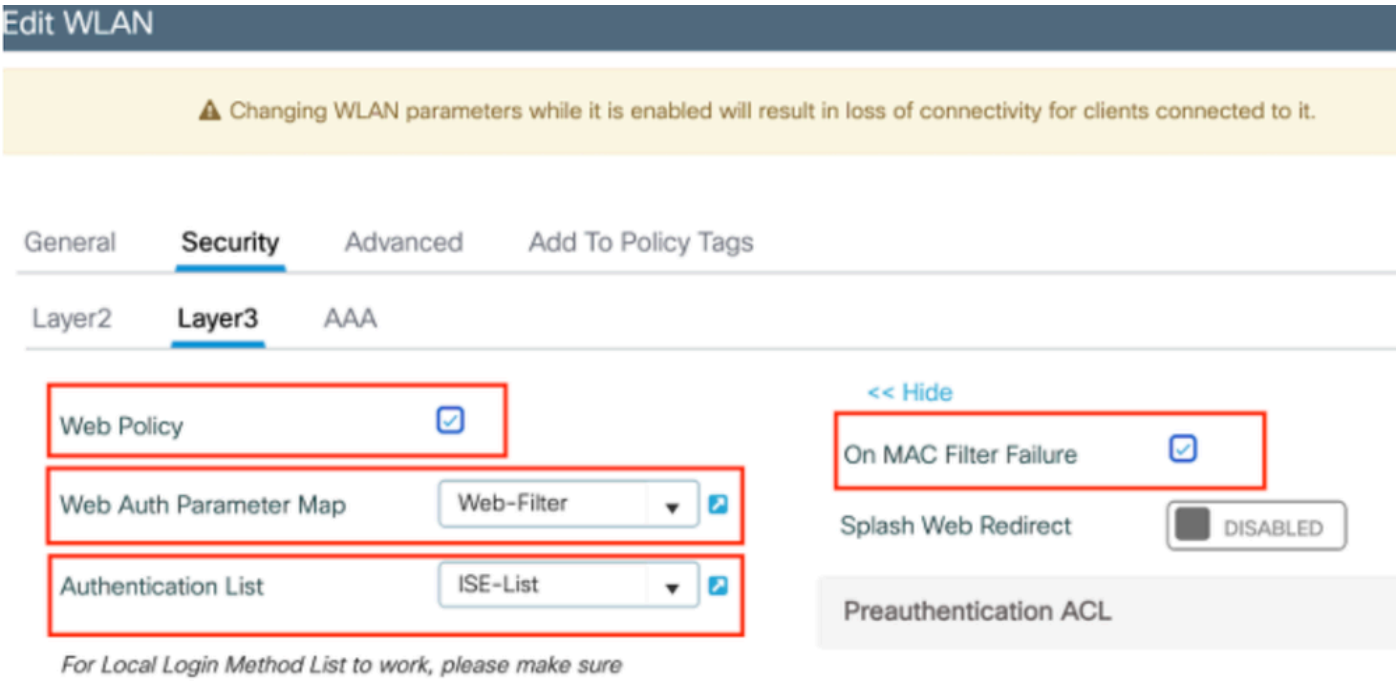
WLAN配置文件

第2步：在“安全”(Security)选项卡下，启用“Mac过滤”(Mac Filtering)复选框，并在授权列表中配置RADIUS服务器（ISE或本地服务器）。此设置将ISE用于Mac身份验证和Web身份验证。



WLAN第2层安全性

第3步：导航到安全>第3层。启用Web策略并将其与Web身份验证参数映射配置文件关联。选中“On Mac Filter Failure”复选框并从Authentication下拉列表中选择RADIUS服务器。



WLAN Layer3 Security选项卡

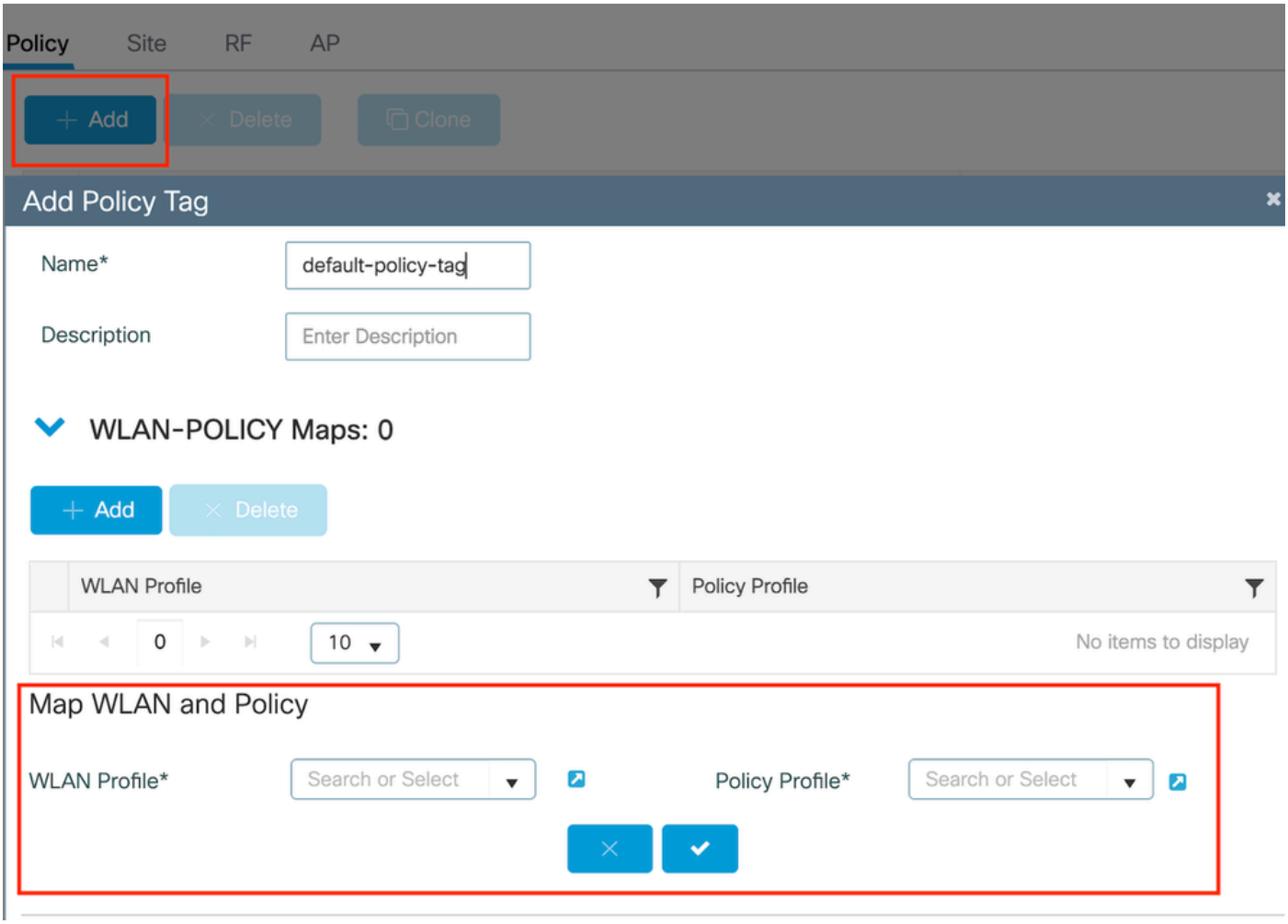
CLI 配置

```
wlan Mac_Filtering_Wlan 9 Mac_Filtering_Wlan
```

```
 mac-filtering network
 radio policy dot11 24ghz
 radio policy dot11 5ghz
 no security ft adaptive
 no security wpa
 no security wpa wpa2
 no security wpa wpa2 ciphers aes
 no security wpa akm dot1x
 security web-auth
 security web-auth authentication-list ISE-List
 security web-auth on-macfilter-failure
 security web-auth parameter-map Web-Filter
 no shutdown
```

第4步：配置策略标记、创建WLAN配置文件和策略配置文件映射

导航到Configuration > Tags & Profiles > Tags > Policy。点击Add以定义策略标记的名称。在WLAN-Policy Maps下，选择“Add”以映射之前创建的WLAN和策略配置文件。



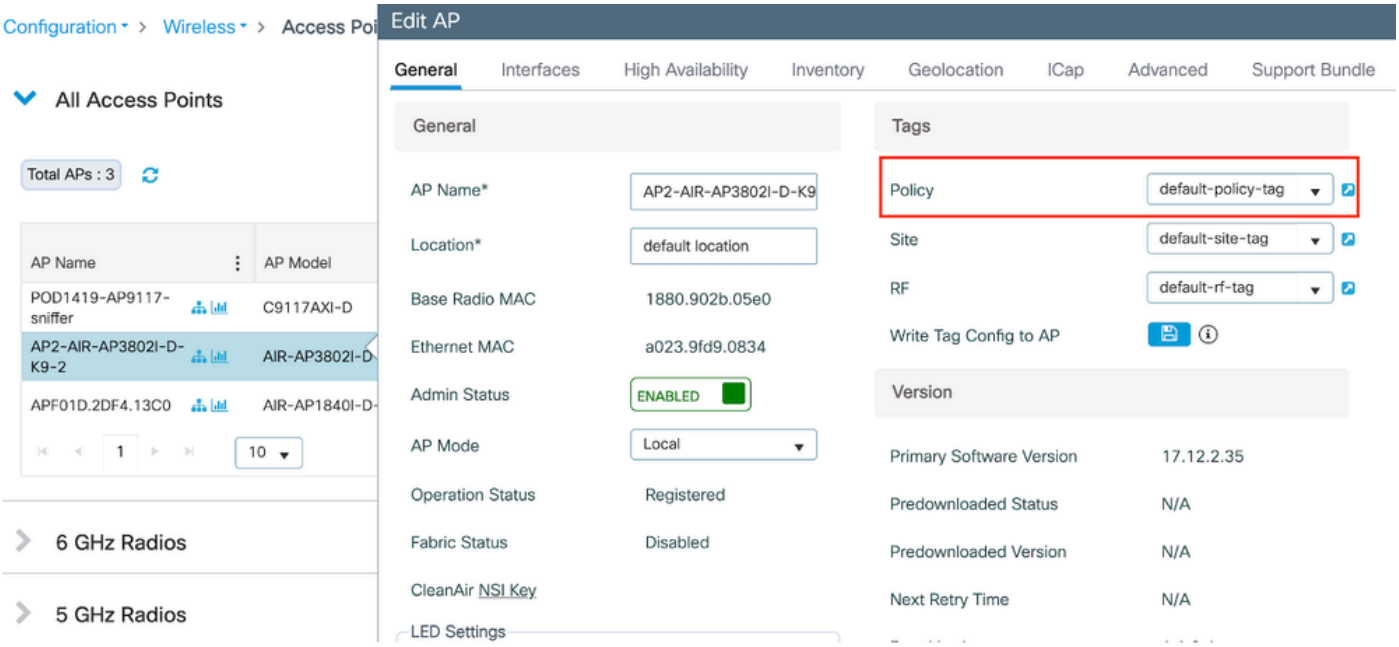
策略标记映射

CLI 配置：

```
wireless tag policy default-policy-tag
 description "default policy-tag"
```

```
wlan Mac_Filtering_Wlan policy Web-Filter-Policy
```

第5步：导航至配置(Configuration) >无线(Wireless) >接入点(Access Point)。选择负责广播此SSID的接入点。在Edit AP菜单中，分配创建的策略标记。

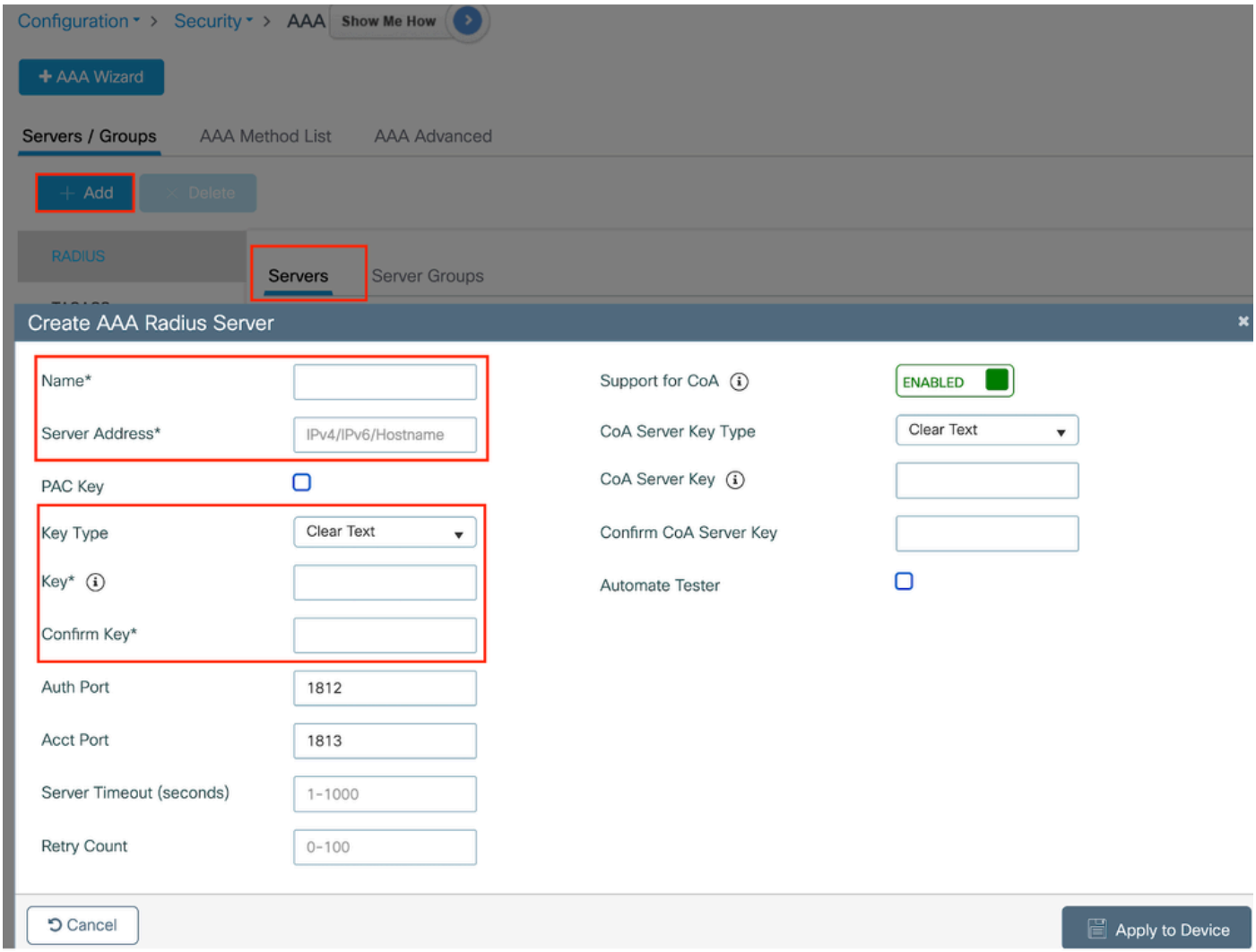


将策略TAG映射到AP

配置AAA设置：

第1步：创建Radius服务器：

导航至Configuration > Security > AAA。单击“服务器/组”部分下的“添加”选项。在“创建AAA Radius服务器”页上，输入服务器名称、IP地址和共享密钥。



服务器配置

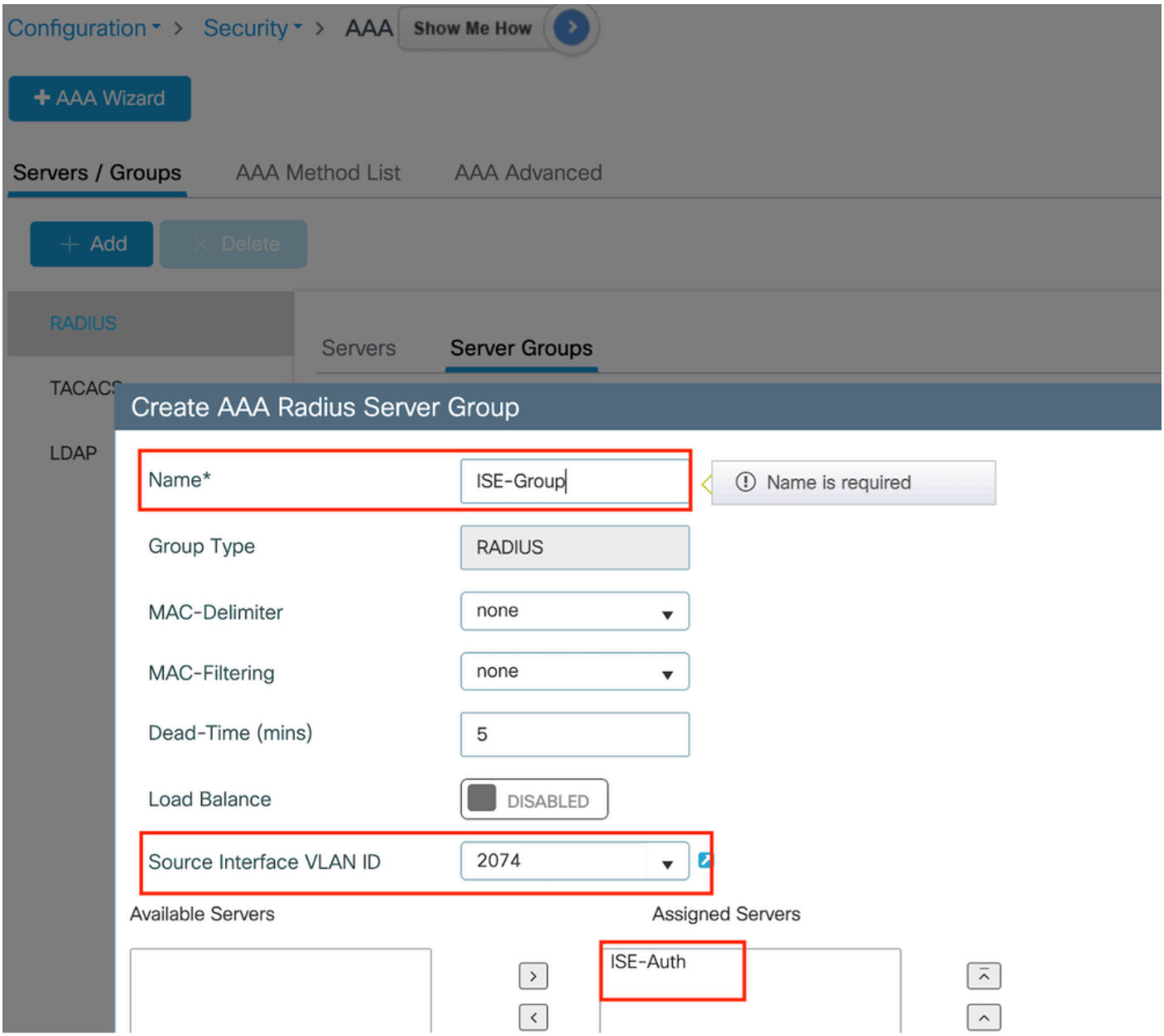
CLI 配置

```
radius server ISE-Auth
 address ipv4 10.197.224.122 auth-port 1812 acct-port 1813
 key *****
 server name ISE-Auth
```

第2步：创建Radius服务器组：

选择“服务器组”部分下的“添加”选项以定义服务器组。切换要包括在同一组配置中的服务器。

无需设置源接口。默认情况下，9800使用其路由表来确定用于连接RADIUS服务器的接口，并且通常使用默认网关。



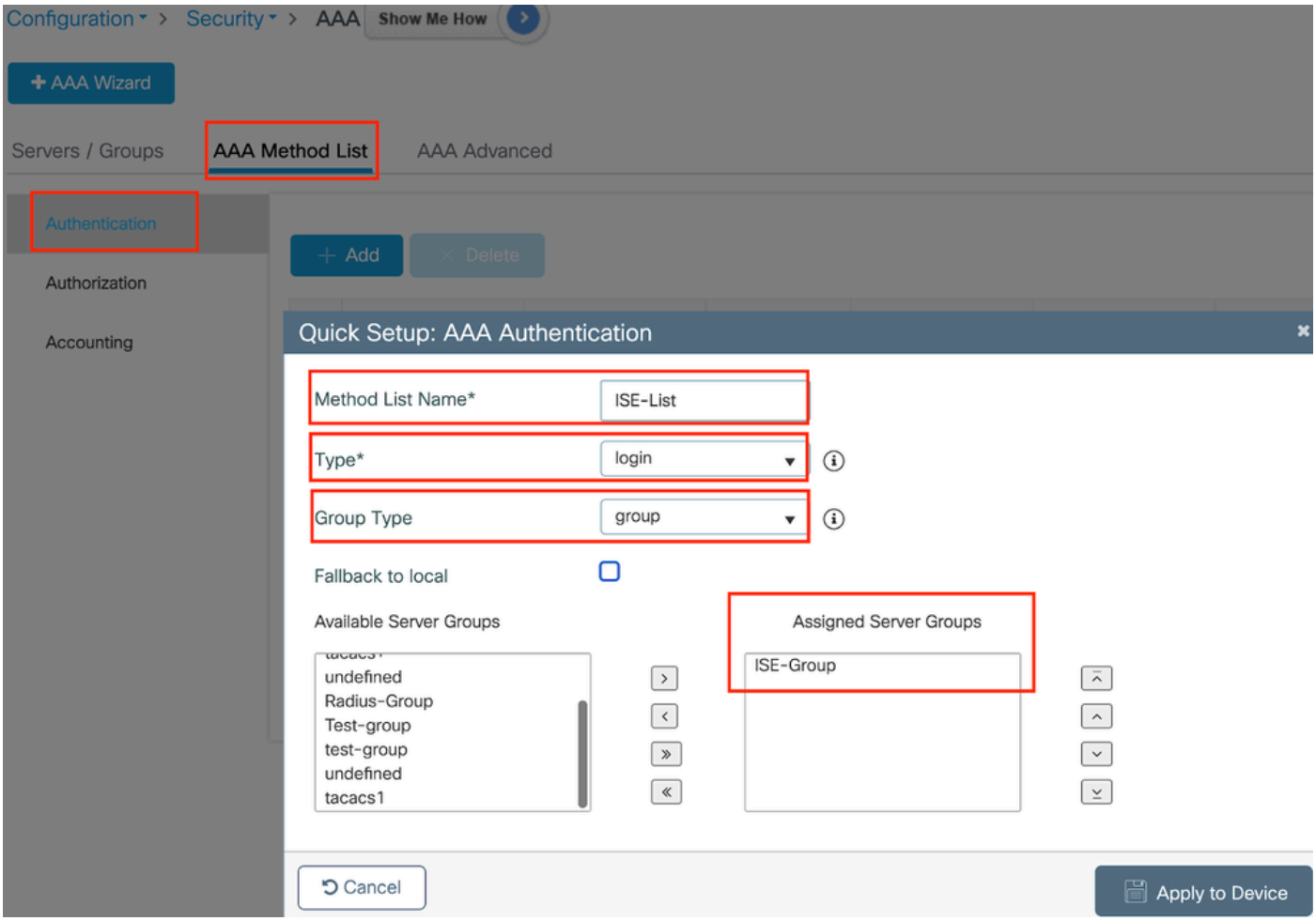
服务器组

CLI 配置

```
aaa group server radius ISE-Group
 server name ISE-Auth
 ip radius source-interface Vlan2074
 deadtime 5
```

第3步：配置AAA方法列表：

导航到AAA Method List选项卡。在Authentication下，点击Add。定义一个方法列表名称，其中Type为“login”，Group type为“Group”。在Assigned Server Group部分下映射已配置的身份验证服务器组。

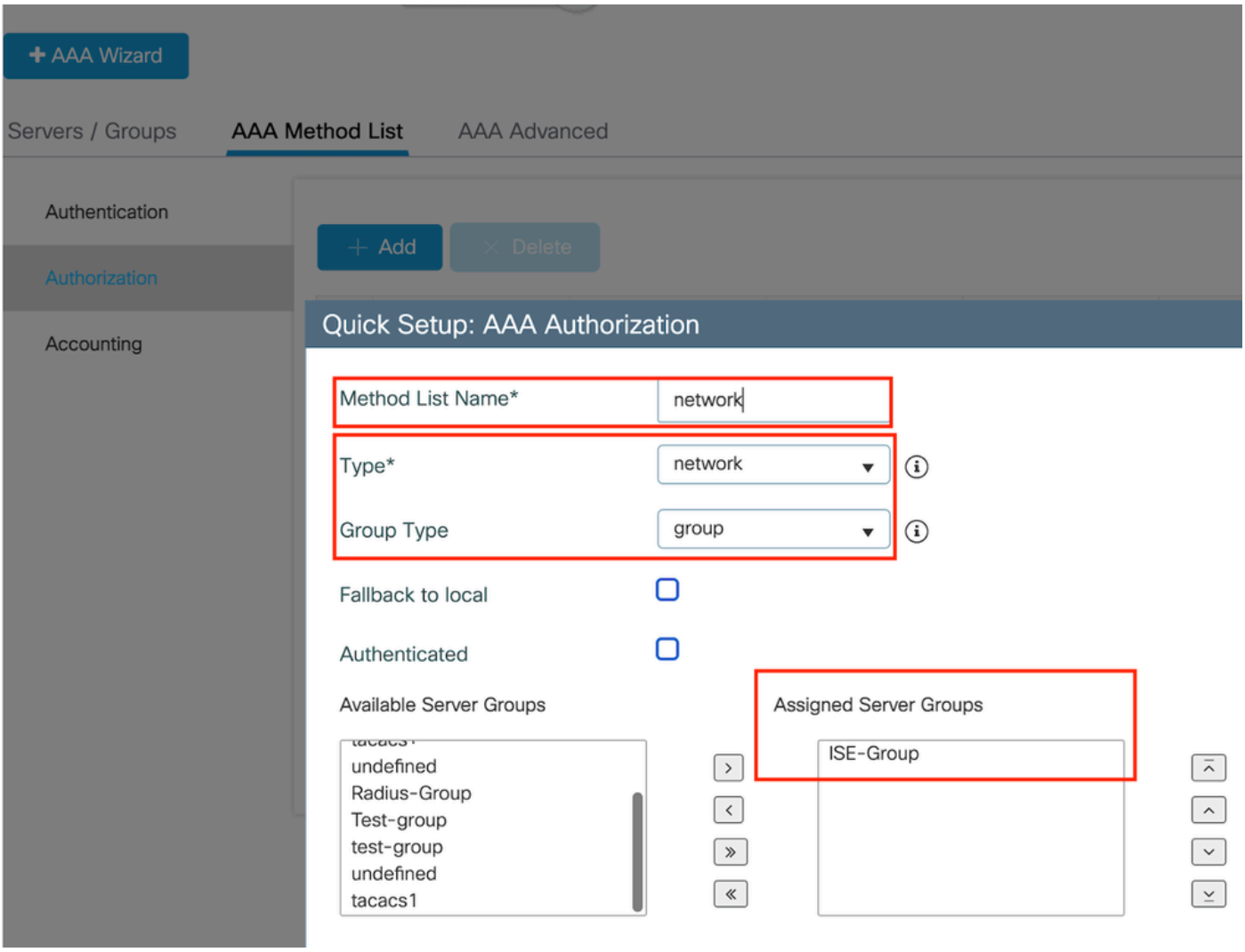


身份验证方法列表

CLI 配置

```
aaa authentication login ISE-List group ISE-Group
```

导航到Authorization Method List部分，然后点击Add。定义方法列表名称，并将类型设置为“network”，将组类型设置为“Group”。将已配置的RADIUS服务器切换到Assigned Server Groups部分。



授权方法列表

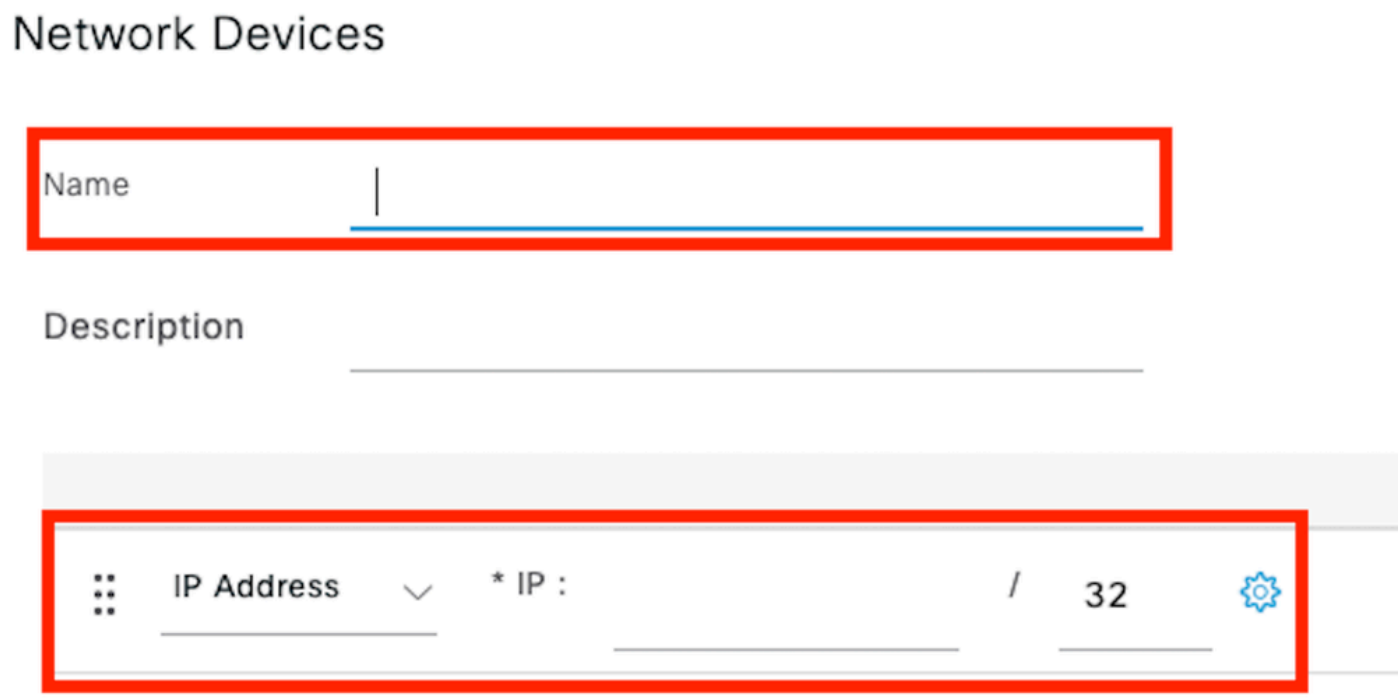
CLI 配置

```
aaa authorization network network group ISE-Group
```

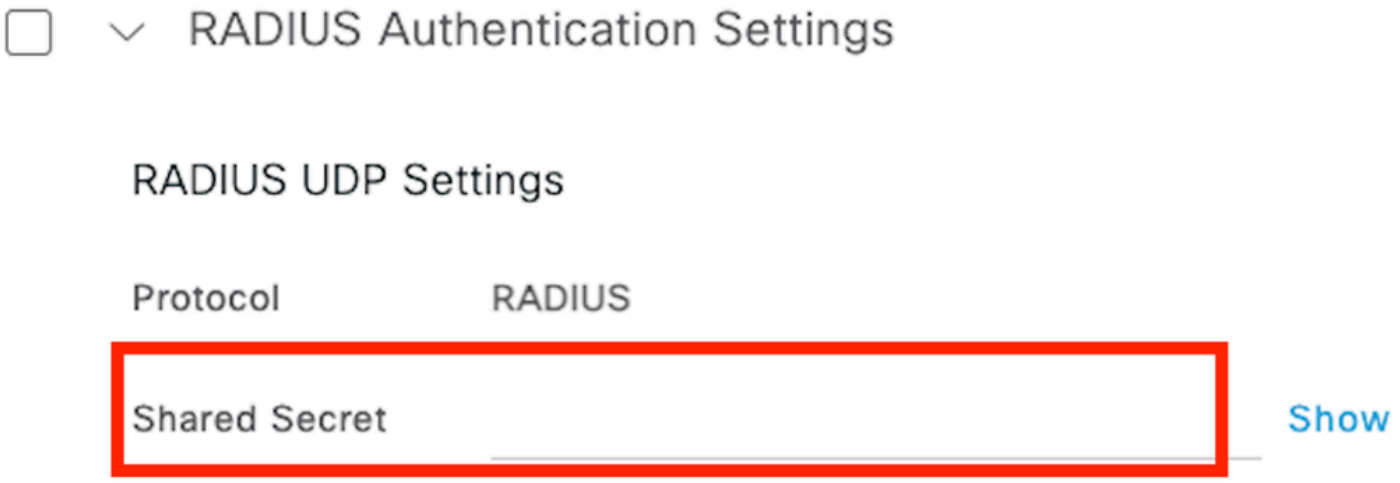
ISE 配置:

在ISE上添加WLC作为网络设备

第1步：导航到管理>网络设备，然后点击添加。在Radius Authentication Settings下输入控制器IP地址、主机名和共享密钥



添加网络设备

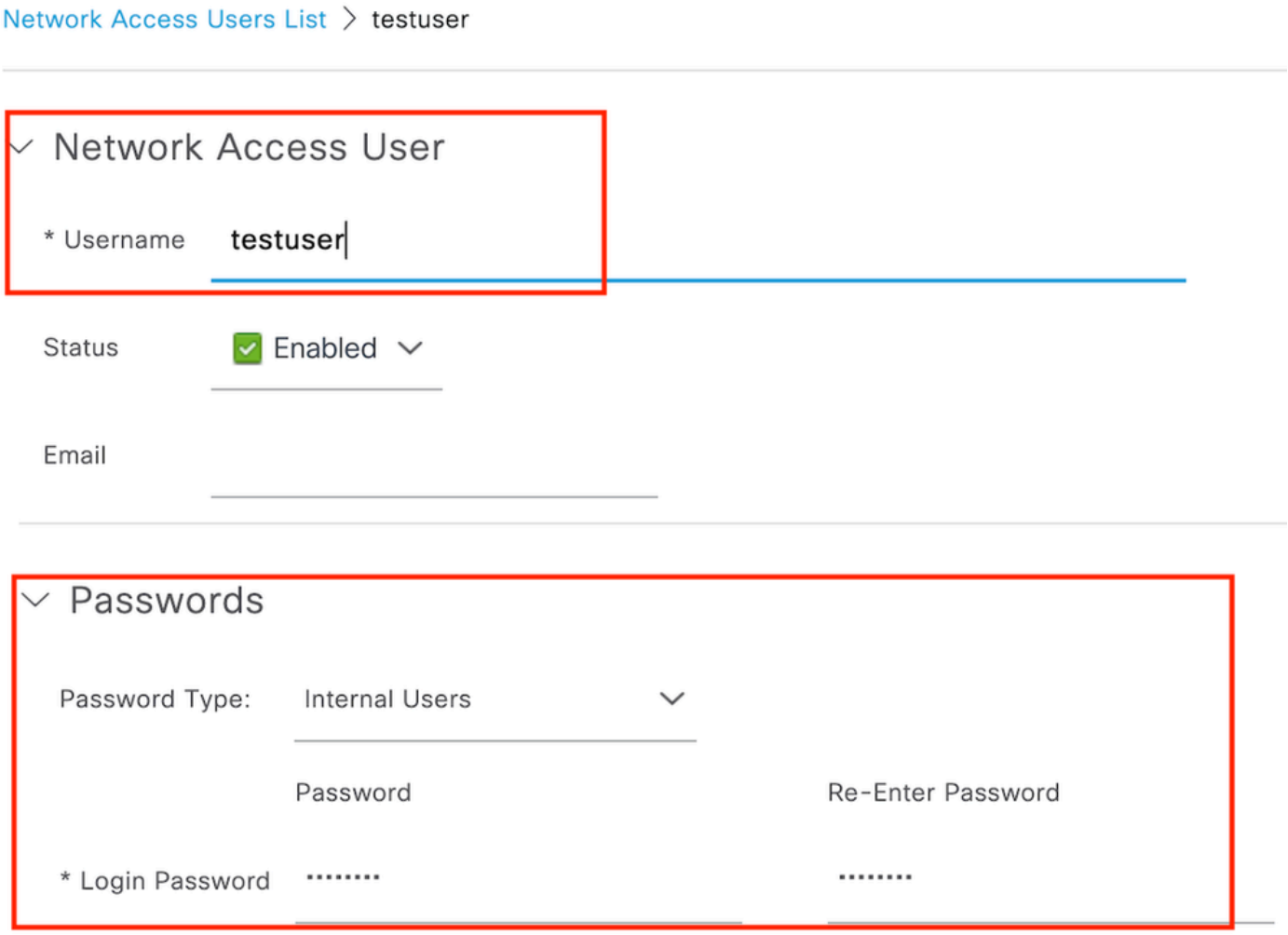


共享密钥

第2步：创建用户条目

在Identity Management > Identities下，选择添加选项。

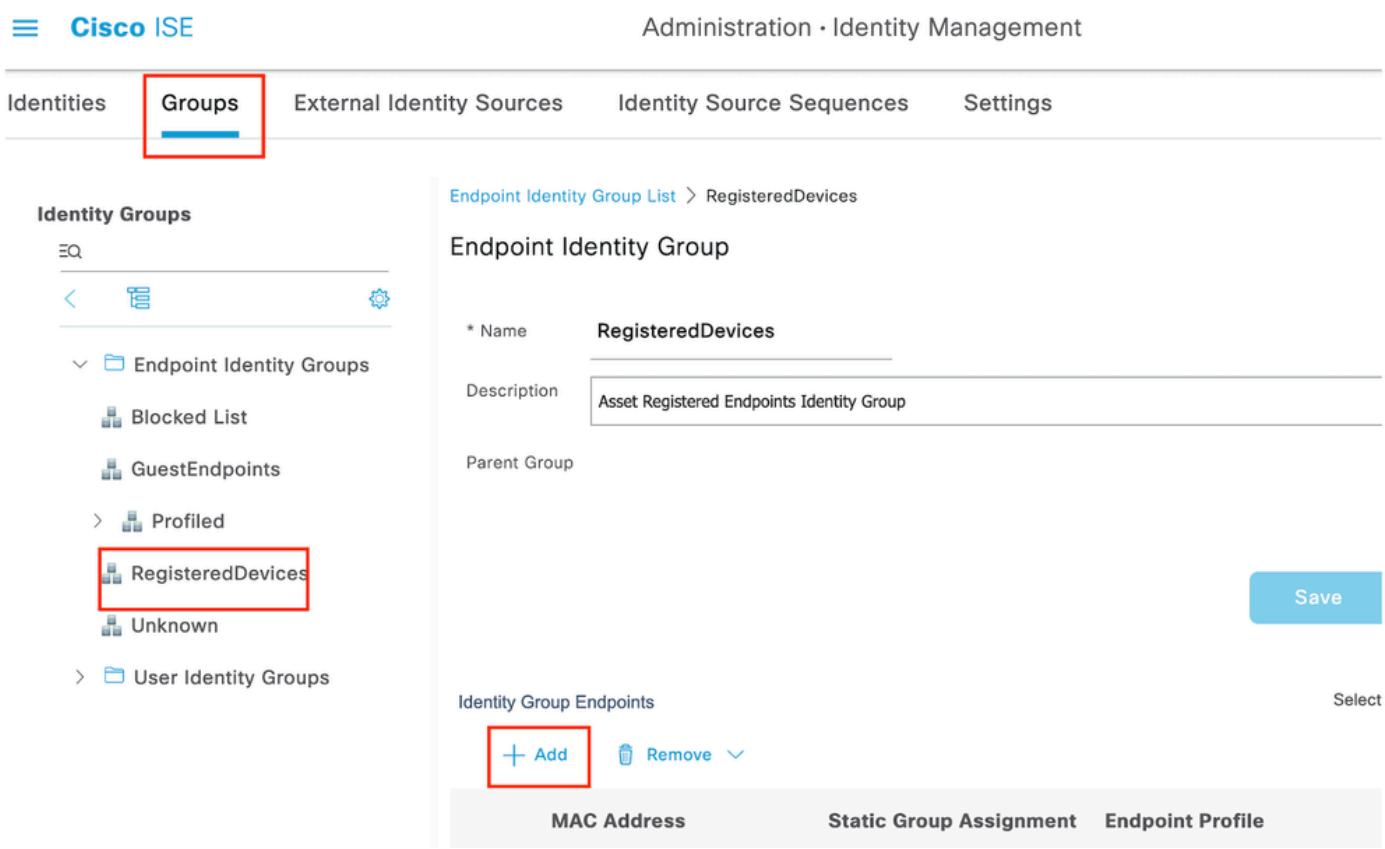
配置客户端必须用于Web身份验证的用户名和密码



添加用户凭证

第3步：导航到管理(Administration) >身份管理(Identity Management) >组(Groups) >注册设备(Registered Devices)，然后点击添加(Add)。

输入设备MAC地址以在服务器上创建条目。

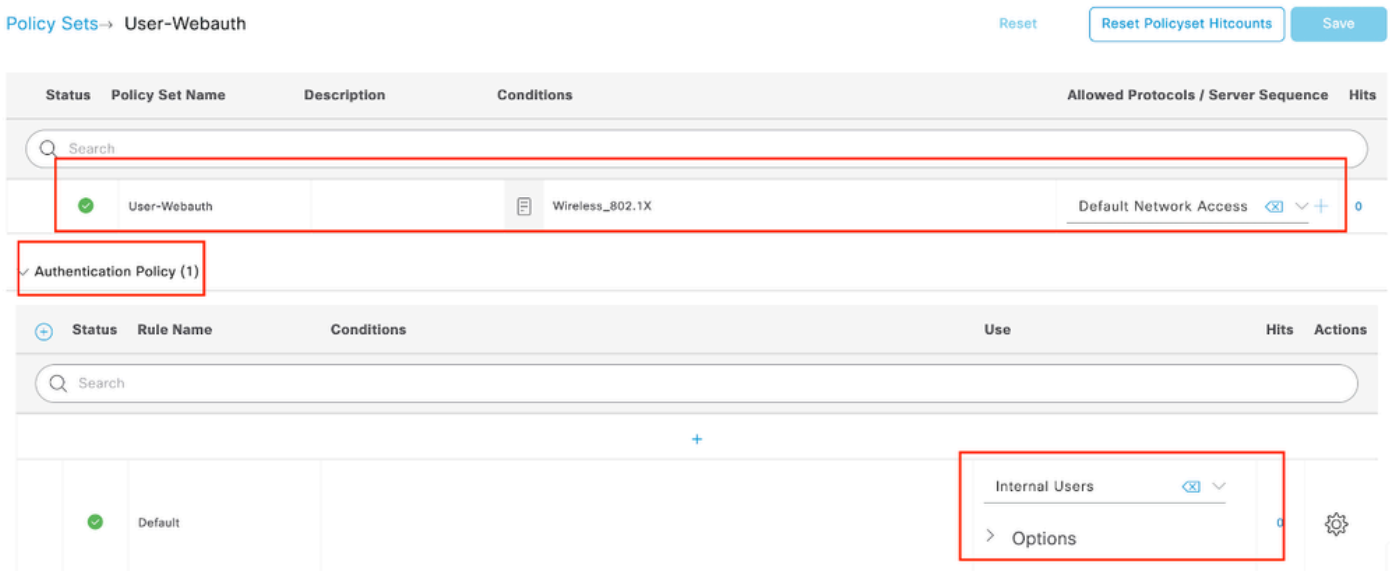


添加设备MAC地址

第4步：创建服务策略

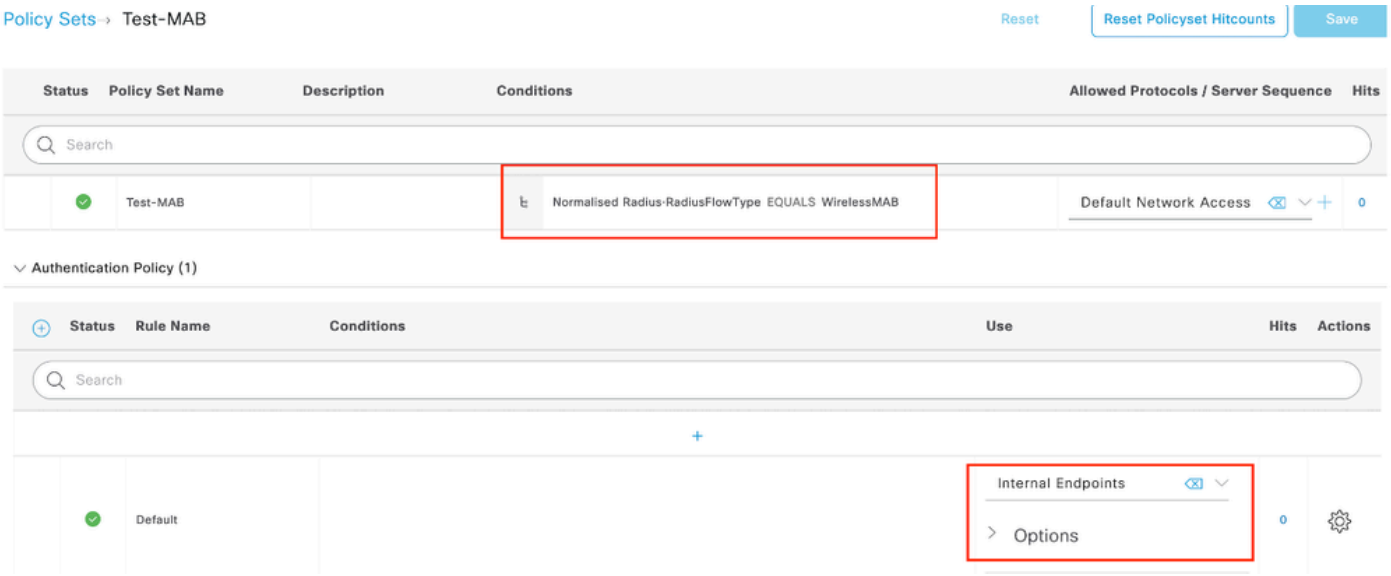
导航到Policy > Policy sets并选择“+”符号以创建新策略集

此策略集用于用户Web身份验证，其中在身份管理中创建客户端的用户名和密码



Web身份验证服务策略

同样，创建MAB服务策略并在身份验证策略下映射内部终端。



MAB身份验证服务策略

验证

控制器配置

<#root>

show wireless tag policy detailed

default-policy-tag

```
Policy Tag Name : default-policy-tag
Description     : default policy-tag
Number of WLAN-POLICY maps: 1
WLAN Profile Name                Policy Name
------------------------------------------------------------------------
```

Mac_Filtering_Wlan

Web-Filter-Policy

<#root>

show wireless profile policy detailed

Web-Filter-Policy

```
Policy Profile Name                :
```

Web-Filter-Policy

```
Description                        :
```

```
Status                              :
```

ENABLED

```
VLAN                                :
```

2074

```
Multicast VLAN                      : 0
```

<#root>

```
show wlan name
```

Mac_Filtering_Wlan

```
WLAN Profile Name     :
```

Mac_Filtering_Wlan

```
================================================
Identifier                                : 9
Description                               :
Network Name (SSID)                       :
```

Mac_Filtering_Wlan

```
Status                                    :
```

Enabled

```
Broadcast SSID                            :
```

Enabled

```
Mac Filter Authorization list name        :
```

network

```
Webauth On-mac-filter Failure             :
```

Enabled

```
    Webauth Authentication List Name      :
```

ISE-List

```
    Webauth Authorization List Name       : Disabled
    Webauth Parameter Map                 :
```

Web-Filter

<#root>

```
show parameter-map type webauth name Web-Filter
Parameter Map Name              :
```

Web-Filter

```
  Type                          :
```

webauth

```
  Auth-proxy Init State time     : 120 sec
  Webauth max-http connection    : 100
  Webauth logout-window          :
```

Enabled

```
  Webauth success-window         :
```

Enabled

```
  Consent Email                  : Disabled
  Activation Mode                : Replace
  Sleeping-Client                : Disabled
  Webauth login-auth-bypass:
```

<#root>

```
show ip http server status

HTTP server status:
```

Enabled

```
HTTP server port:
```

80

```
HTTP server active supplementary listener ports: 21111
HTTP server authentication method: local
HTTP server auth-retry 0 time-window 0
HTTP server digest algorithm: md5
HTTP server access class: 0
HTTP server IPv4 access class: None
HTTP server IPv6 access class: None
HTTP server base path:
HTTP File Upload status: Disabled
HTTP server upload path:
HTTP server help root:
Maximum number of concurrent server connections allowed: 300
Maximum number of secondary server connections allowed: 50
Server idle time-out: 180 seconds
Server life time-out: 180 seconds
Server session idle time-out: 600 seconds
Maximum number of requests allowed on a connection: 25
Server linger time : 60 seconds
HTTP server active session modules: ALL
HTTP secure server capability: Present
HTTP secure server status:
```

Enabled

```
HTTP secure server port:
```

443

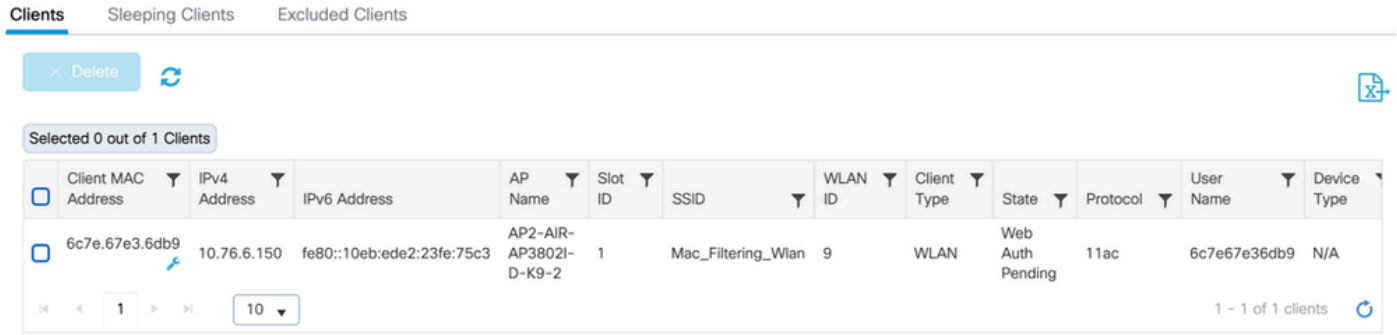
```
show ap name AP2-AIR-AP3802I-D-K9-2 tag detail

Policy tag mapping
------------------
```

```
WLAN Profile Name                Policy Name                    VLAN                          Flex
-------------------------------------------------------------------------------------------------
Mac_Filtering_Wlan               Web-Filter-Policy              2074                          ENAB
```

控制器上的客户端策略状态

导航到Dashboard（控制面板）> Clients（客户端）部分，确认连接的客户端的状态。
客户端当前处于Web身份验证挂起状态



客户端详细信息

```
show wireless client summary
Number of Clients: 1
MAC Address    AP Name                                    Type ID   State             Protocol Meth
-------------------------------------------------------------------------------------------------
6c7e.67e3.6db9 AP2-AIR-AP3802I-D-K9-2                     WLAN 9    Webauth Pending   11ac     Web
```

<#root>

```
show wireless client mac-address 6c7e.67e3.6db9 detail
Client MAC Address :
```

6c7e.67e3.6db9

```
Client MAC Type : Universally Administered Address
Client DUID: NA
Client IPv4 Address :
```

10.76.6.150

```
Client IPv6 Addresses : fe80::10eb:ede2:23fe:75c3
Client Username :
```

6c7e67e36db9

```
AP MAC Address : 1880.902b.05e0
AP Name: AP2-AIR-AP3802I-D-K9-2
AP slot : 1
Client State : Associated
Policy Profile :
```

Web-Filter-Policy

```
Flex Profile : N/A
```

```
Wireless LAN Id: 9
WLAN Profile Name:
```

Mac_Filtering_Wlan

```
Wireless LAN Network Name (SSID): Mac_Filtering_Wlan
BSSID : 1880.902b.05eb

Client ACLs : None
Mac authentication :
```

Failed

```
Policy Manager State:
```

Webauth Pending

```
Last Policy Manager State :
```

IP Learn Complete

```
Client Entry Create Time : 88 seconds
Policy Type : N/A
Encryption Cipher : None

Auth Method Status List
        Method : Web Auth
                Webauth State    :
```

Get Redirect

```
                Webauth Method   :
```

Webauth

在网络身份验证成功后，客户端策略管理器状态会转换为RUN

<#root>

```
show wireless client mac-address 6c7e.67e3.6db9 detail

Client ACLs : None
Mac authentication : Failed
Policy Manager State:
```

Run

```
Last Policy Manager State :
```

Webauth Pending

```
Client Entry Create Time : 131 seconds
Policy Type : N/A
```

故障排除

Web Auth on MAC Failure功能的功能依赖于控制器功能在MAB出现故障时触发Web身份验证。我们的主要目标是从控制器中有效地收集RA跟踪，以进行故障排除和分析。

放射性痕量收集

激活Radio Active Tracing以在CLI中为指定的MAC地址生成客户端调试跟踪。

启用放射性跟踪的步骤：

确保禁用所有条件调试

```
clear platform condition all
```

启用对指定MAC地址的调试

```
debug wireless mac <H.H.H> monitor-time <Time is seconds>
```

重现问题后，请禁用调试以停止RA跟踪收集。

```
no debug wireless mac <H.H.H>
```

一旦RA跟踪停止，就会在控制器bootflash中生成调试文件。

```
show bootflash: | include ra_trace
2728        179 Jul 17 2024 15:13:54.0000000000 +00:00 ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_Da
```

将文件复制到外部服务器。

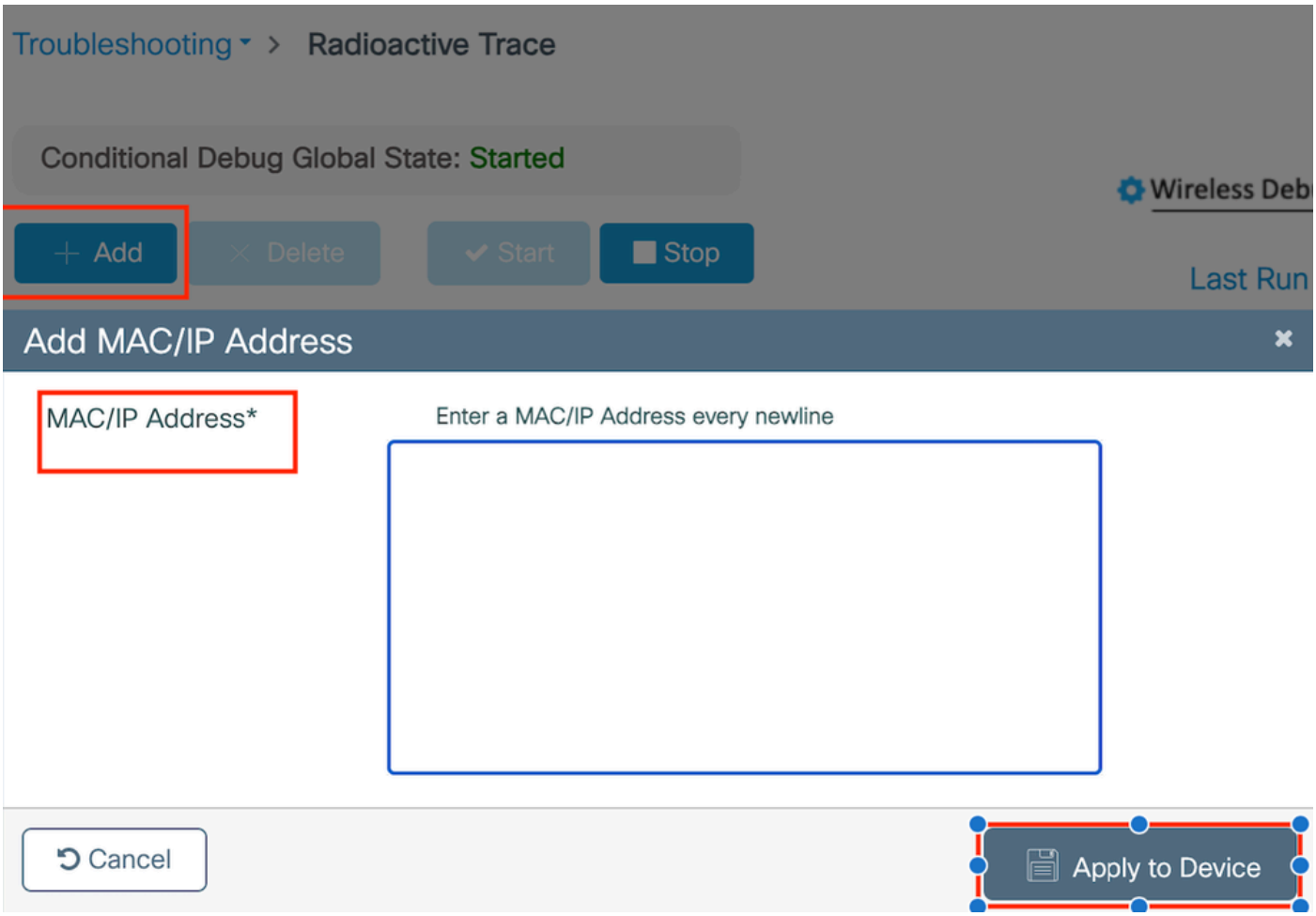
```
copy bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log tftp://<IP addr
```

显示调试日志：

```
more bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log
```

在GUI中启用RA跟踪，

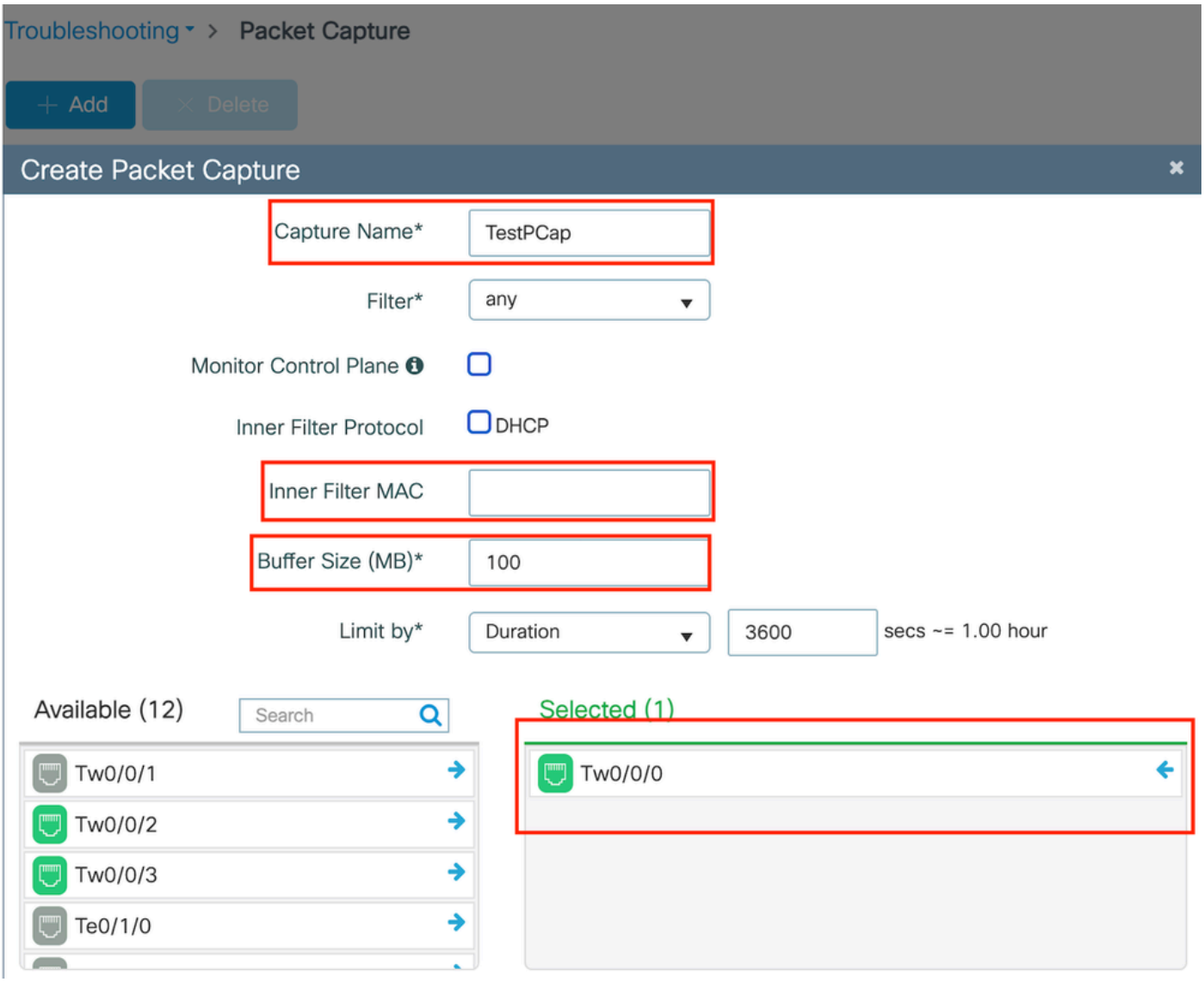
第1步：导航至故障排除>放射性跟踪。选择添加新条目的选项，然后在指定的添加MAC/IP地址(Add MAC/IP Address)选项卡中输入客户端MAC地址。



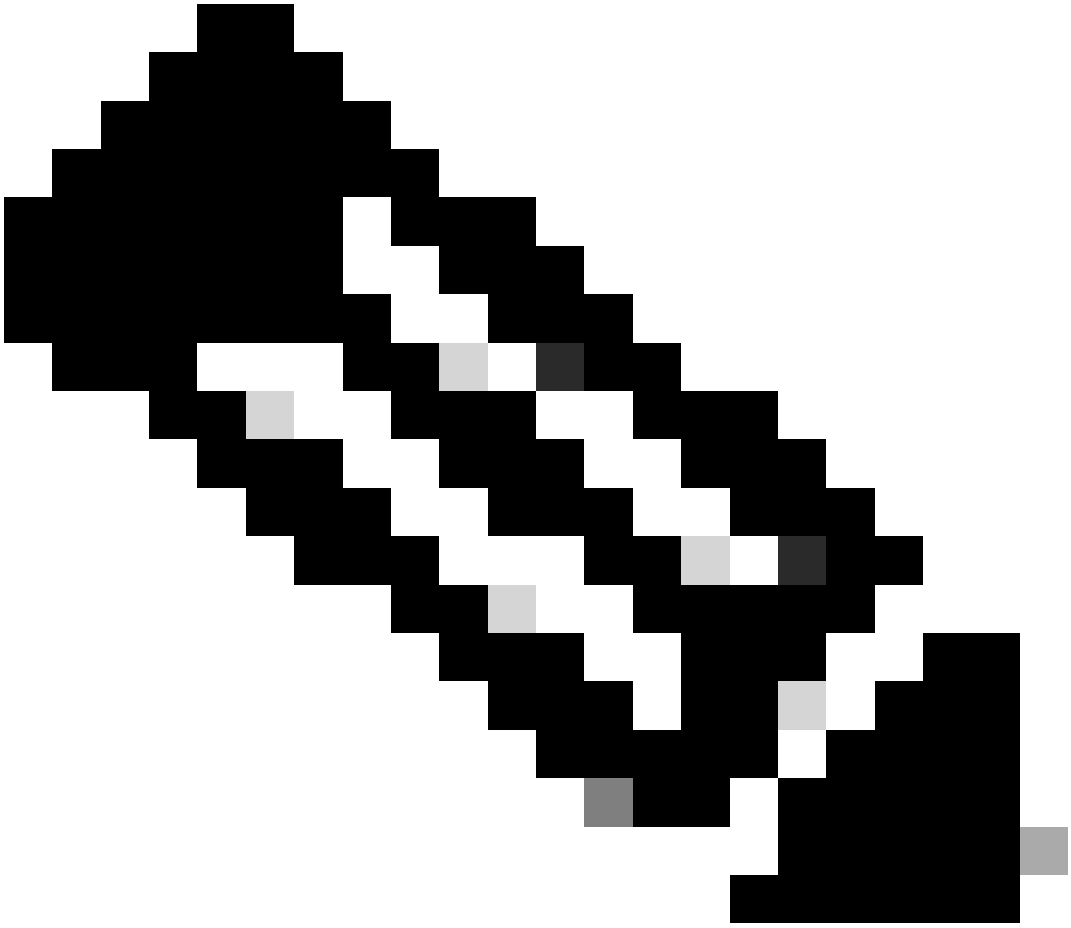
无线电主动跟踪

嵌入式数据包捕获：

导航到Troubleshooting > Packet Capture。输入捕获名称并指定客户端MAC地址作为内部过滤器MAC。将缓冲区大小设置为100并选择上行链路接口来监控传入和传出数据包。

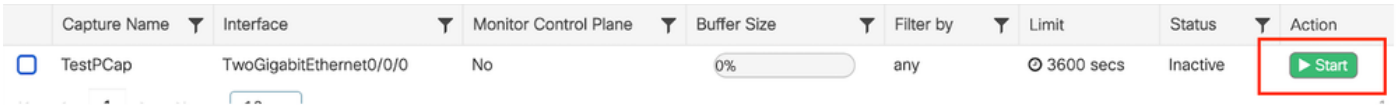

Troubleshooting
Packet Capture
Add
Delete
Create Packet Capture
Capture Name*
TestPCap
Filter*
any
Monitor Control Plane
Inner Filter Protocol
DHCP
Inner Filter MAC
Buffer Size (MB)*
100
Limit by*
Duration
3600
secs ~= 1.00 hour
Available (12)
Search
Tw0/0/1
Tw0/0/2
Tw0/0/3
Te0/1/0
Selected (1)
Tw0/0/0

嵌入式数据包捕获



注意：选择“监控控制流量”选项以查看重定向到系统CPU并重新注入数据平面的流量。

选择Start捕获数据包



开始捕获

CLI 配置

```
monitor capture TestPCap inner mac <H.H.H>
monitor capture TestPCap buffer size 100
monitor capture TestPCap interface twoGigabitEthernet 0/0/0 both
monitor capture TestPCap start

<Reporduce the issue>
```

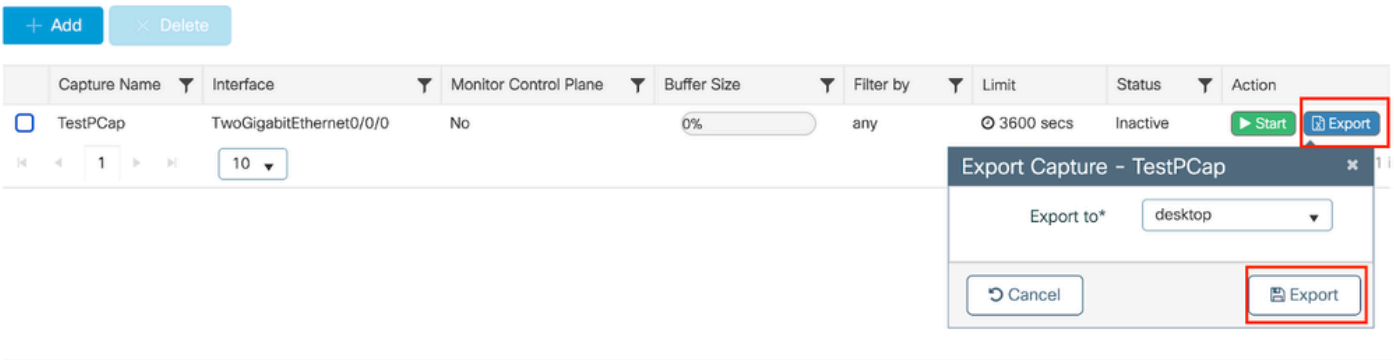
```
monitor capture TestPCap stop
```

```
show monitor capture TestPCap

Status Information for Capture TestPCap
  Target Type:
 Interface: TwoGigabitEthernet0/0/0, Direction: BOTH
  Status : Inactive
  Filter Details:
  Capture all packets
  Inner Filter Details:
  Mac: 6c7e.67e3.6db9
  Continuous capture: disabled
  Buffer Details:
  Buffer Type: LINEAR (default)
  Buffer Size (in MB): 100
  Limit Details:
  Number of Packets to capture: 0 (no limit)
  Packet Capture duration: 3600
  Packet Size to capture: 0 (no limit)
  Maximum number of packets to capture per second: 1000
  Packet sampling rate: 0 (no sampling)
```

将数据包捕获导出到外部TFTP服务器

```
monitor capture TestPCap export tftp://<IP address>/ TestPCap.pcap
```



导出数据包捕获

示例场景在MAC身份验证成功期间，客户端设备连接到网络，其MAC地址由RADIUS服务器通过配置的策略进行验证，并在验证后，由网络接入设备授予访问权限，从而允许网络连接。

客户端关联后，控制器向ISE服务器发送访问请求，

User name是客户端的MAC地址，因为这是MAB身份验证

```
2024/07/16 21:12:52.711298748 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS: Send Access-Request t
```

```
2024/07/16 21:12:52.711310730 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  authenticator 19 c6
2024/07/16 21:12:52.711326401 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  User-Name
2024/07/16 21:12:52.711329615 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  User-Password
2024/07/16 21:12:52.711337331 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Service-Type
2024/07/16 21:12:52.711340443 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Vendor, Cisco
2024/07/16 21:12:52.711344513 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:   Cisco AVpair
2024/07/16 21:12:52.711349087 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Framed-MTU
2024/07/16 21:12:52.711351935 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Message-Authenticato
2024/07/16 21:12:52.711377387 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  EAP-Key-Name
2024/07/16 21:12:52.711382613 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Vendor, Cisco
2024/07/16 21:12:52.711385989 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:   Cisco AVpair
```

ISE发送Access-Accept，因为我们有有效的用户条目

```
2024/07/16 21:12:52.779147404 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS: Received from id 1812,
2024/07/16 21:12:52.779156117 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  authenticator 5d dc
2024/07/16 21:12:52.779161793 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  User-Name
2024/07/16 21:12:52.779165183 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Class
2024/07/16 21:12:52.779219803 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Message-Authenticato
```

```
2024/07/16 21:12:52.779417578 {wncd_x_R0-0}{1}: [mab] [17765]: (info): [6c7e.67b7.2d29:capwap_90000005]
2024/07/16 21:12:52.779436247 {wncd_x_R0-0}{1}: [mab] [17765]: (info): [6c7e.67b7.2d29:capwap_90000005]
```

客户端策略状态转换为Mac Auth已完成

```
2024/07/16 21:12:52.780181486 {wncd_x_R0-0}{1}: [client-auth] [17765]: (info): MAC: 6c7e.67b7.2d29  Cli
2024/07/16 21:12:52.780238297 {wncd_x_R0-0}{1}: [client-orch-sm] [17765]: (debug): MAC: 6c7e.67b7.2d29
```

在成功MAB身份验证后，客户端处于IP learn状态

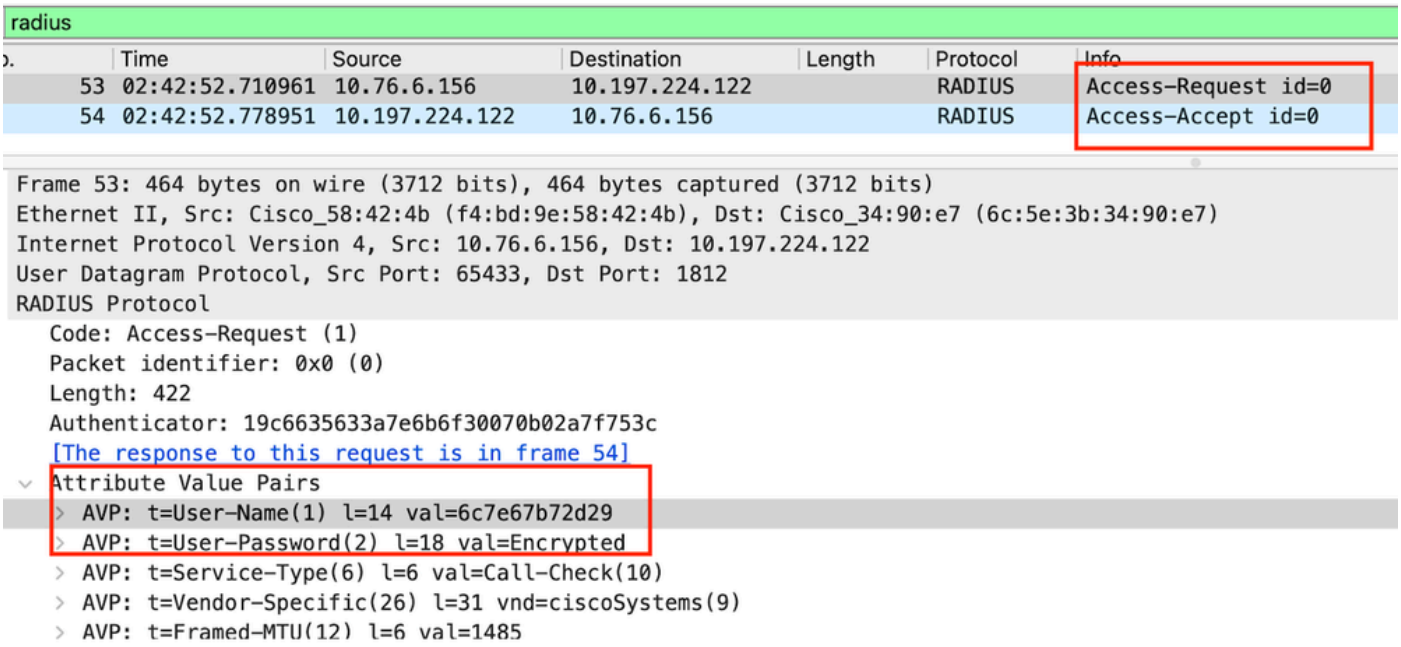
```
2024/07/16 21:12:55.791404789 {wncd_x_R0-0}{1}: [client-orch-state] [17765]: (note): MAC: 6c7e.67b7.2d2
2024/07/16 21:12:55.791739386 {wncd_x_R0-0}{1}: [client-iplearn] [17765]: (info): MAC: 6c7e.67b7.2d29
```

```
2024/07/16 21:12:55.794130301 {iosrp_R0-0}{1}: [buginf] [4440]: (debug): AUTH-FEAT-SISF-EVENT: IP updat
```

客户端策略管理器状态更新为RUN，对完成MAB身份验证的客户端跳过Web身份验证

```
2024/07/16 21:13:11.210786952 {wncd_x_R0-0}{1}: [errmsg] [17765]: (info): %CLIENT_ORCH_LOG-6-CLIENT_ADD
```

使用嵌入式数据包捕获进行验证



Radius数据包

客户端设备的MAC身份验证失败的示例

在成功关联后为客户端启动MAC身份验证

```
2024/07/17 03:20:59.842211775 {wncd_x_R0-0}{1}: [mab] [17765]: (info): [6c7e.67e3.6db9:capwap_90000005]
2024/07/17 03:20:59.842280253 {wncd_x_R0-0}{1}: [ewlc-infra-evq] [17765]: (note): Authentication Succes
2024/07/17 03:20:59.842284313 {wncd_x_R0-0}{1}: [client-auth] [17765]: (info): MAC: 6c7e.67e3.6db9  Cli
2024/07/17 03:20:59.842320572 {wncd_x_R0-0}{1}: [mab] [17765]: (info): [6c7e.67e3.6db9:capwap_90000005]
```

ISE将发送Access-Reject，因为ISE中不存在此设备条目

```
2024/07/17 03:20:59.842678322 {wncd_x_R0-0}{1}: [mab] [17765]: (info): [6c7e.67e3.6db9:capwap_90000005]
2024/07/17 03:20:59.842877636 {wncd_x_R0-0}{1}: [auth-mgr] [17765]: (info): [6c7e.67e3.6db9:capwap_9000
```

由于MAB失败，已为客户端设备启动Web-Auth

```
2024/07/17 03:20:59.843728206 {wncd_x_R0-0}{1}: [client-auth] [17765]: (info): MAC: 6c7e.67e3.6db9  Cli
```

一旦客户端发起HTTP GET请求，重定向URL将被推送到客户端设备，因为相应的TCP会话被控制器欺骗。

```
2024/07/17 03:21:37.817434046 {wncd_x_R0-0}{1}: [webauth-httpd] [17765]: (info): capwap_90000005[6c7e.6
2024/07/17 03:21:37.817459639 {wncd_x_R0-0}{1}: [webauth-httpd] [17765]: (debug): capwap_90000005[6c7e.
2024/07/17 03:21:37.817466483 {wncd_x_R0-0}{1}: [webauth-httpd] [17765]: (debug): capwap_90000005[6c7e.
2024/07/17 03:21:37.817482231 {wncd_x_R0-0}{1}: [webauth-state] [17765]: (info): capwap_90000005[6c7e.6
```

客户端向重定向URL发起HTTP Get，页面加载后，登录凭证即被提交。

控制器向ISE发送访问请求

这是Web身份验证，因为在Access-Accept数据包中观察到有效的用户名

```
2024/07/17 03:22:51.132347799 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS: Send Access-Request t
2024/07/17 03:22:51.132362949 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  authenticator fd 40
2024/07/17 03:22:51.132368737 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Calling-Station-Id
2024/07/17 03:22:51.132372791 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  User-Name
2024/07/17 03:22:51.132376569 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Vendor, Cisco
```

从ISE接收的Access - Accept

```
2024/07/17 03:22:51.187040709 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS: Received from id 1812
2024/07/17 03:22:51.187050061 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  authenticator d3 ac
2024/07/17 03:22:51.187055731 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  User-Name
2024/07/17 03:22:51.187059053 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Class
2024/07/17 03:22:51.187102553 {wncd_x_R0-0}{1}: [radius] [17765]: (info): RADIUS:  Message-Authenticato
```

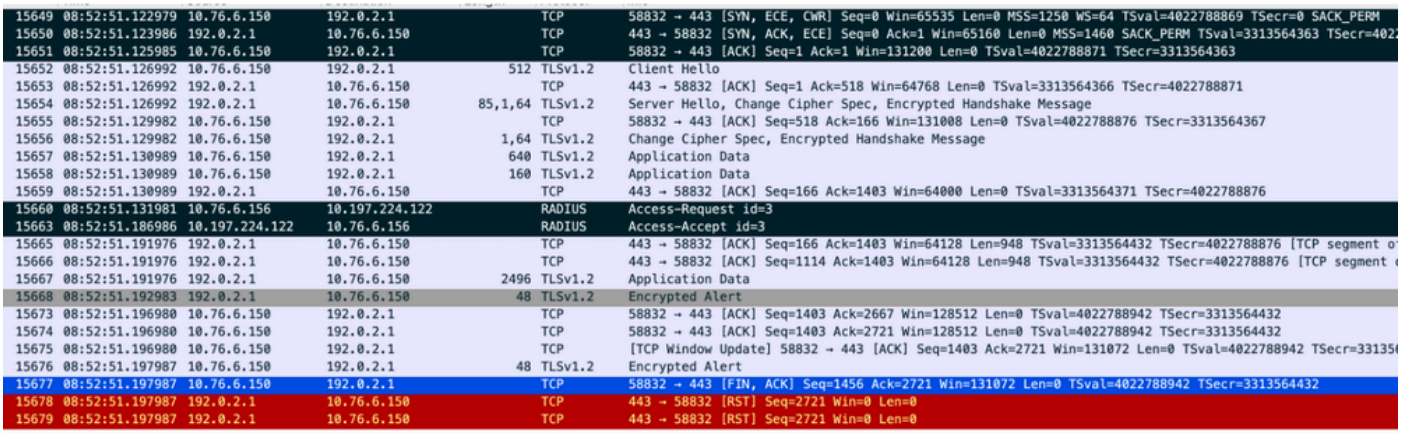
Web身份验证成功，客户端状态转换到运行状态

```
2024/07/17 03:22:51.193775717 {wncd_x_R0-0}{1}: [errmsg] [17765]: (info): %CLIENT_ORCH_LOG-6-CLIENT_ADD
2024/07/17 03:22:51.194009423 {wncd_x_R0-0}{1}: [client-orch-state] [17765]: (note): MAC: 6c7e.67e3.6db
```

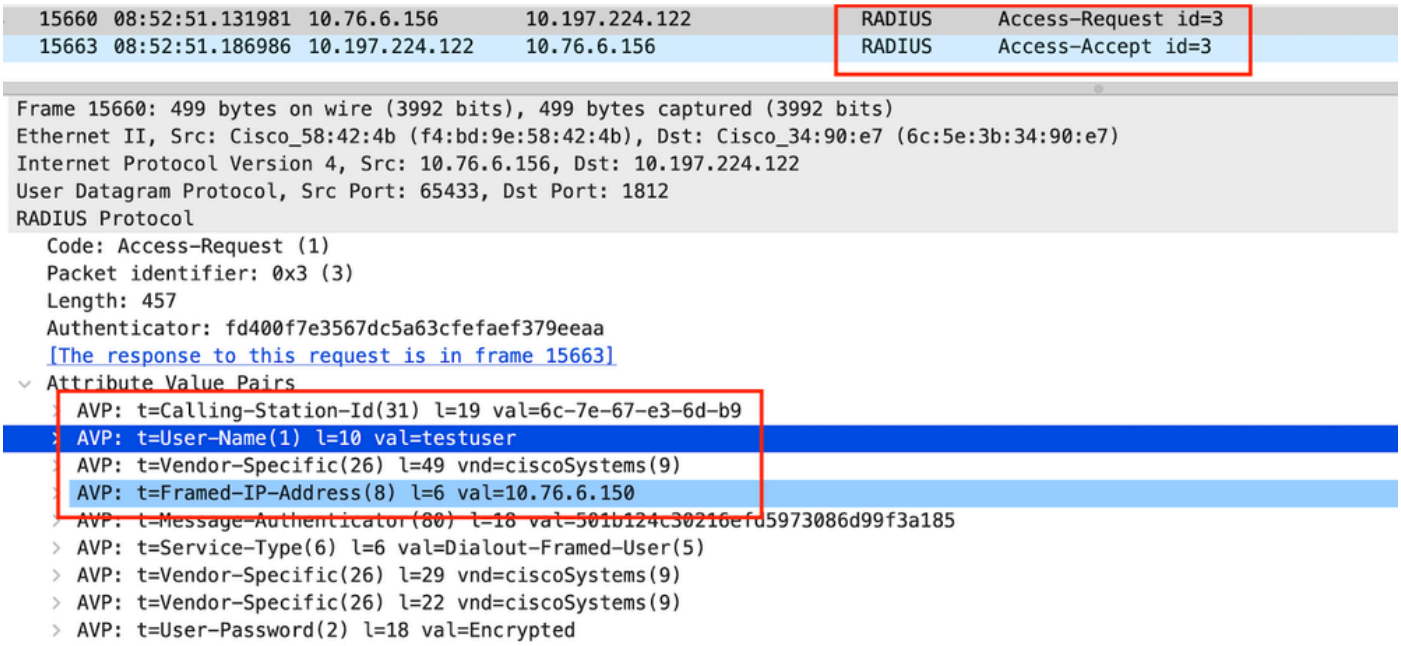
通过EPC捕获进行验证

客户端完成与控制器虚拟IP地址的TCP握手，然后客户端加载重定向门户页面。用户提交用户名和密码后，我们可以观察来自控制器管理IP地址的radius访问请求。

身份验证成功后，客户端TCP会话关闭，并且在控制器上，客户端转换到RUN状态。

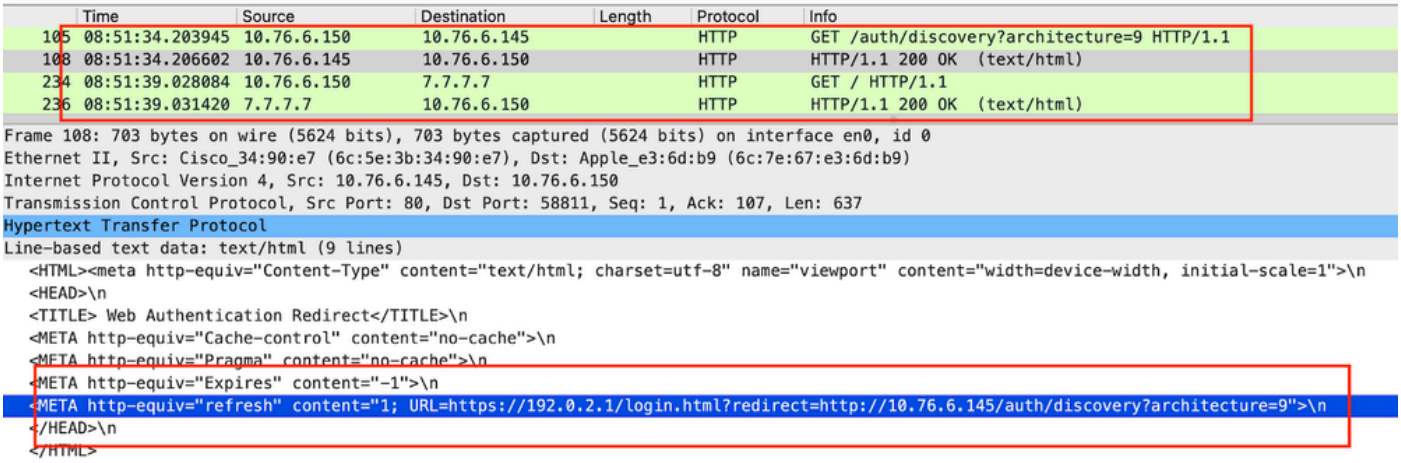


使用RADIUS数据包的TCP流



使用用户凭证发送到ISE的RADIUS数据包

客户端wireshark捕获验证客户端流量是否被重定向到门户页面，并验证与控制器虚拟ip地址/Web服务器的TCP握手



客户端捕获以验证重定向url

客户端与控制器的虚拟IP地址建立TCP握手

	Time	Source	Destination	Length	Protocol	Info
115	08:51:34.208377	10.76.6.150	192.0.2.1		TCP	58812 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3224314628 TSecr=0 SACK_PE
117	08:51:34.211190	192.0.2.1	10.76.6.150		TCP	443 → 58812 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65160 Len=0 MSS=1250 SACK_PERM TSval=3313491061 TSecr
118	08:51:34.211275	10.76.6.150	192.0.2.1		TCP	58812 → 443 [ACK] Seq=1 Ack=1 Win=131200 Len=0 TSval=3224314631 TSecr=3313491061
120	08:51:34.212673	10.76.6.150	192.0.2.1	512	TLSv1.2	Client Hello
122	08:51:34.217896	192.0.2.1	10.76.6.150		TCP	443 → 58812 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TSval=3313491066 TSecr=3224314632
124	08:51:34.220834	192.0.2.1	10.76.6.150	89,830	TLSv1.2	Server Hello, Certificate
125	08:51:34.220835	192.0.2.1	10.76.6.150	783.4	TLSv1.2	Server Key Exchange, Server Hello Done

客户端和Web服务器之间的TCP握手

会话在Web身份验证成功后关闭，

144	08:51:34.235915	10.76.6.150	192.0.2.1		TCP	[TCP Window Update] 58812 → 443 [ACK] Seq=1145 Ack=10183 Win=131072 Len=0 TSval=3224314655 TSe
145	08:51:34.235996	10.76.6.150	192.0.2.1	52	TLSv1.2	Encrypted Alert
146	08:51:34.236029	10.76.6.150	192.0.2.1		TCP	58812 → 443 [FIN, ACK] Seq=1202 Ack=10183 Win=131072 Len=0 TSval=3224314655 TSecr=3313491084
147	08:51:34.238965	192.0.2.1	10.76.6.150	52	TLSv1.2	Encrypted Alert
148	08:51:34.238966	192.0.2.1	10.76.6.150		TCP	443 → 58812 [FIN, ACK] Seq=10240 Ack=1203 Win=64256 Len=0 TSval=3313491089 TSecr=3224314655

客户端完成Web身份验证后TCP会话关闭

相关文章

[了解Catalyst 9800无线LAN控制器上的无线调试和日志收集]

[9800上基于Web的身份验证]

[在9800上配置本地网络身份验证]

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。