在9800 WLC上配置外部Web身份验证并排除故障

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

配置

- 网络图
- 配置Web参数设置
  - CLI配置摘要：
- 配置AAA设置
- 配置策略和标记

验证

故障排除

- 永远在线跟踪
- 条件调试和无线电主动跟踪
- 嵌入式数据包捕获
- 客户端故障排除
  - HAR浏览器故障排除
  - 客户端数据包捕获
- 成功尝试的示例

简介

本文档介绍如何在Catalyst 9800无线LAN控制器(WLC)上配置外部Web身份验证(EWA)并对其进行故障排除。

先决条件

本文档假设Web服务器已正确配置为允许外部通信，并且网页已正确配置为发送WLC验证用户和将客户端会话移至RUN状态所需的所有参数。

注意：由于外部资源访问由WLC通过访问列表权限限制，因此所有脚本、字体、图像等都将受到限制。需要下载网页中使用的内容，并保留在web服务器的本地位置。

用户身份验证的必要参数包括：

- 已点击按钮：需要将此参数设置为值“4”，WLC才能检测作为身份验证尝试的操作。
- redirectUrl:此参数中的值由控制器用于在身份验证成功时将客户端定向到特定网站。
- err_flag:此参数用于指示某些错误，例如信息不完整或凭证不正确，在成功的身份验证时，将

其设置为“0”。
- username：此参数仅用于webauth参数映射，如果参数映射设置为consent，则可以忽略它。必须填写无线客户端用户名。
- password：此参数仅用于webauth参数映射，如果参数映射设置为consent，则可以忽略它。必须填写无线客户端密码。

要求

Cisco 建议您了解以下主题：

- 超文本标记语言(HTML)Web开发
- Cisco IOS®-XE无线功能
- Web浏览器开发工具

使用的组件

本文档中的信息基于以下软件和硬件版本：

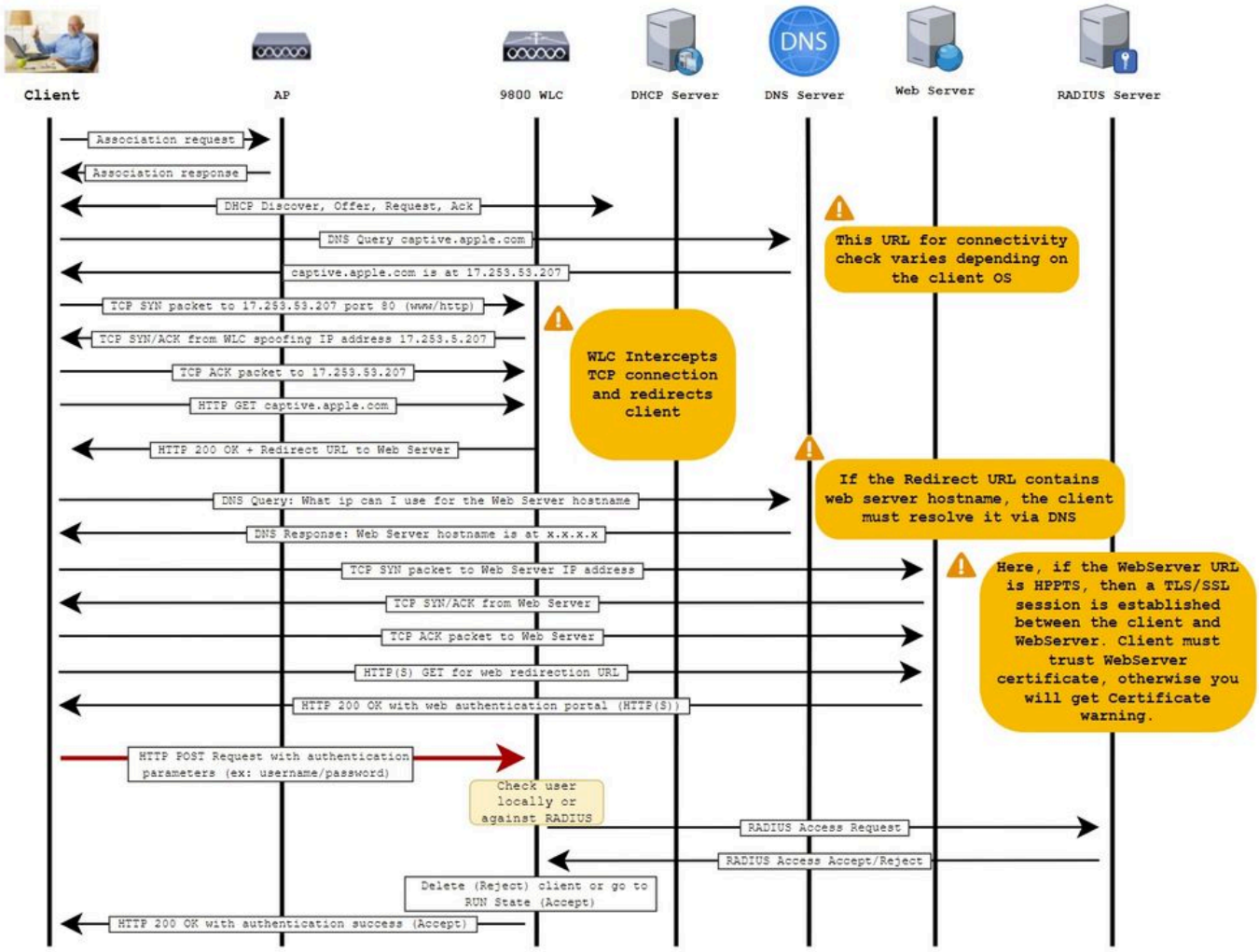
- C9800-CL WLC Cisco IOS®-XE版本17.3.3
- 具有Internet信息服务(IIS)功能的Microsoft Windows Server 2012
- 2802和9117接入点

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

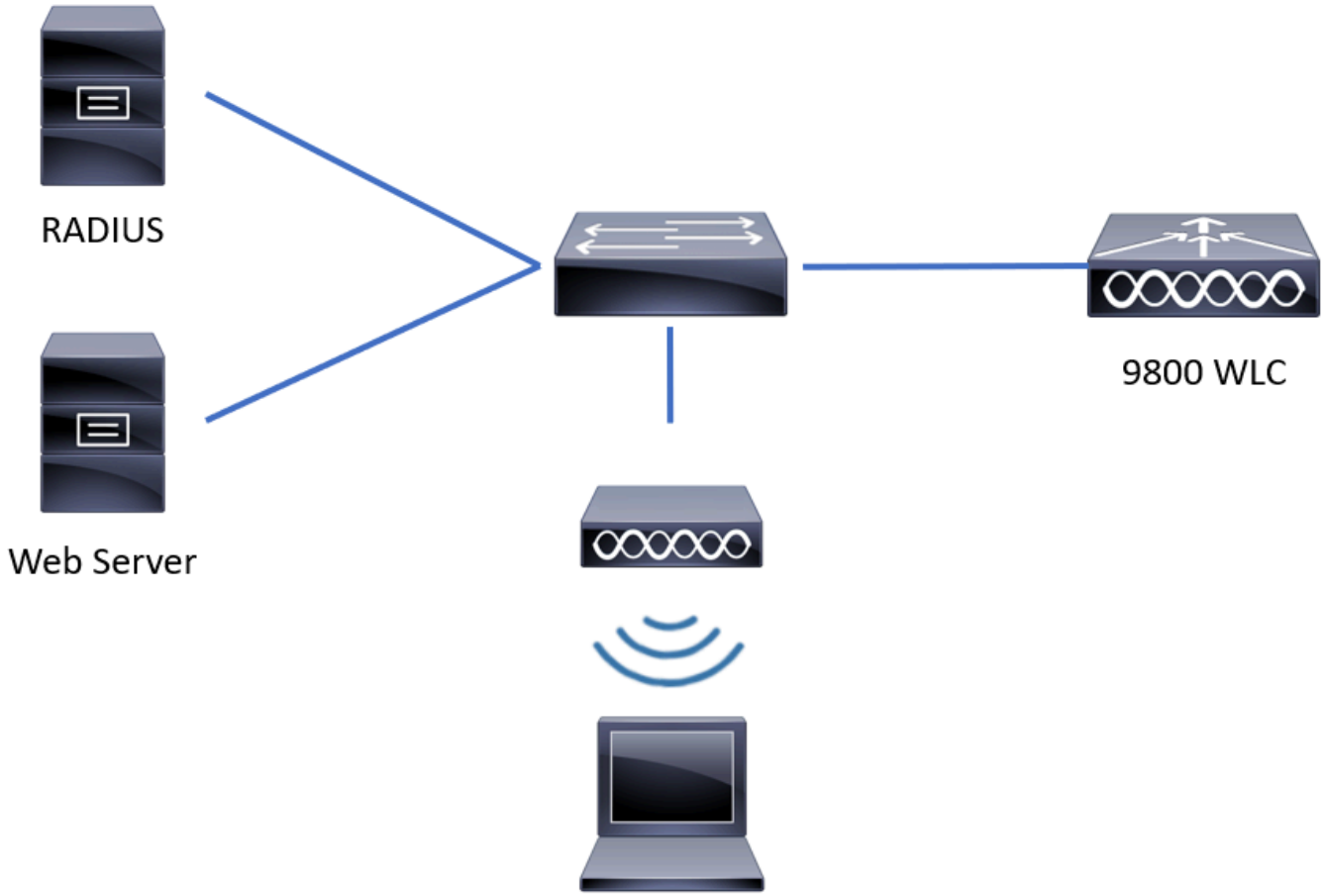
外部Web身份验证利用托管在WLC外部的专用Web服务器或多用途服务器（如身份服务引擎[ISE]）上的Web门户，允许对Web组件进行粒度访问和管理。在映像中呈现了将客户端成功加入外部Web身份验证WLAN所涉及的握手。该映像列出了无线客户端、WLC、解析统一资源位置(URL)的域名系统(DNS)服务器和Webserver（WLC在本地验证用户凭证）之间的顺序交互。此工作流程有助于排除任何故障情况。


注意：在从客户端向WLC进行HTTP POST调用之前，如果在参数映射中启用了安全Web身份验证，并且如果WLC没有由受信任证书颁发机构签名的信任点，则浏览器中会显示安全警报。客户端需要绕过此警告并接受表单重新提交，以便控制器将客户端会话置于RUN状态。


Client
AP
9800 WLC
DHCP Server
DNS Server
Web Server
RADIUS Server
Association request
Association response
DHCP Discover, Offer, Request, Ack
DNS Query captive.apple.com
captive.apple.com is at 17.253.53.207
TCP SYN packet to 17.253.53.207 port 80 (www/http)
TCP SYN/ACK from WLC spoofing IP address 17.253.5.207
TCP ACK packet to 17.253.53.207
HTTP GET captive.apple.com
HTTP 200 OK + Redirect URL to Web Server
DNS Query: What ip can I use for the Web Server hostname
DNS Response: Web Server hostname is at x.x.x.x
TCP SYN packet to Web Server IP address
TCP SYN/ACK from Web Server
TCP ACK packet to Web Server
HTTP(S) GET for web redirection URL
HTTP 200 OK with web authentication portal (HTTP(S))
HTTP POST Request with authentication parameters (ex: username/password)
Check user locally or against RADIUS
RADIUS Access Request
RADIUS Access Accept/Reject
Delete (Reject) client or go to RUN State (Accept)
HTTP 200 OK with authentication success (Accept)
This URL for connectivity check varies depending on the client OS
WLC Intercepts TCP connection and redirects client
If the Redirect URL contains web server hostname, the client must resolve it via DNS
Here, if the WebServer URL is HPPTS, then a TLS/SSL session is established between the client and WebServer. Client must trust WebServer certificate, otherwise you will get Certificate warning.

配置

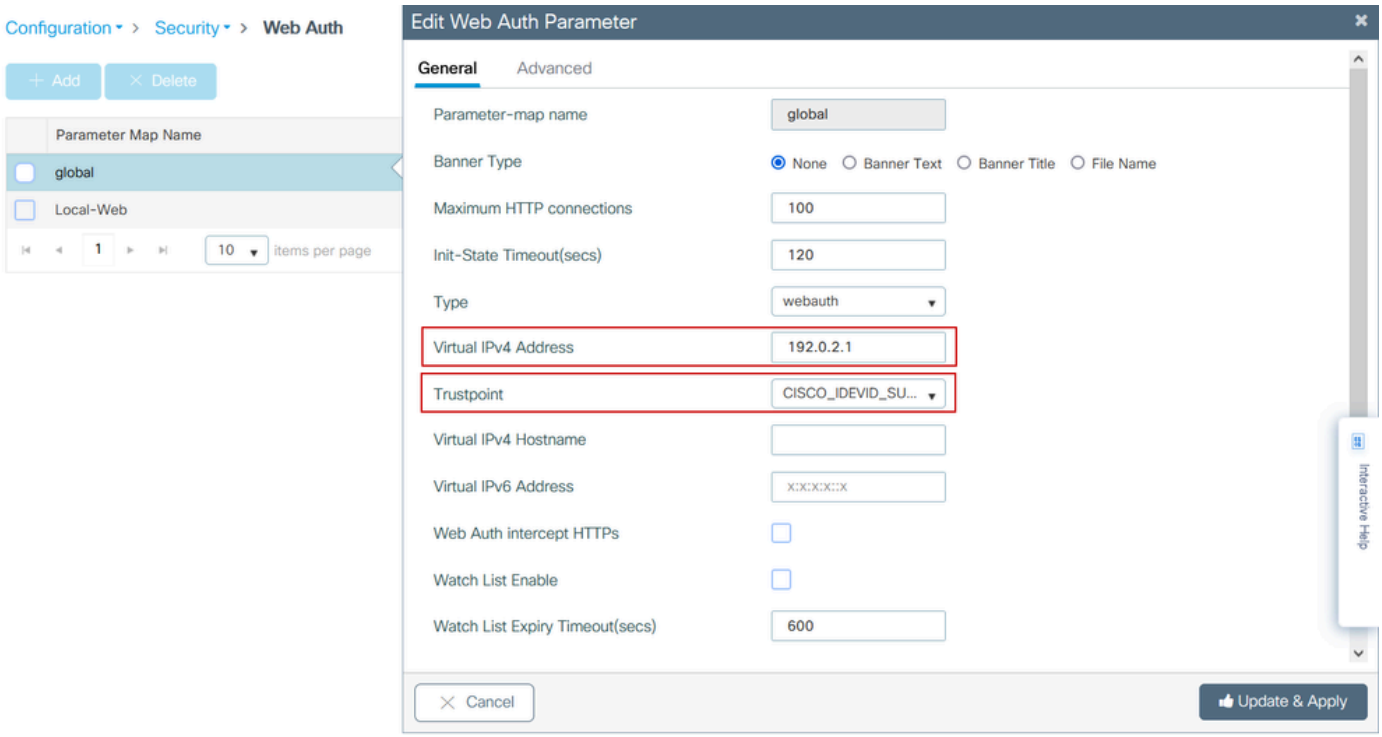
网络图



配置Web参数设置

步骤1.导航至配置>安全> Web Auth，然后选择全局参数映射。验证是否配置了虚拟IPv4地址和信任点，以提供正确的重定向功能。如果您的网络中启用了IPv6，那么配置虚拟IPv6地址也是必要的。如果没有虚拟IP，可能无法进行重定向，并且凭据可能以不安全的方式发送。

注意：默认情况下，浏览器使用HTTP网站启动重定向进程，如果需要HTTPS重定向，则必须选中Web Auth intercept HTTPs;但是，不建议使用此配置，因为它会增加CPU使用率。



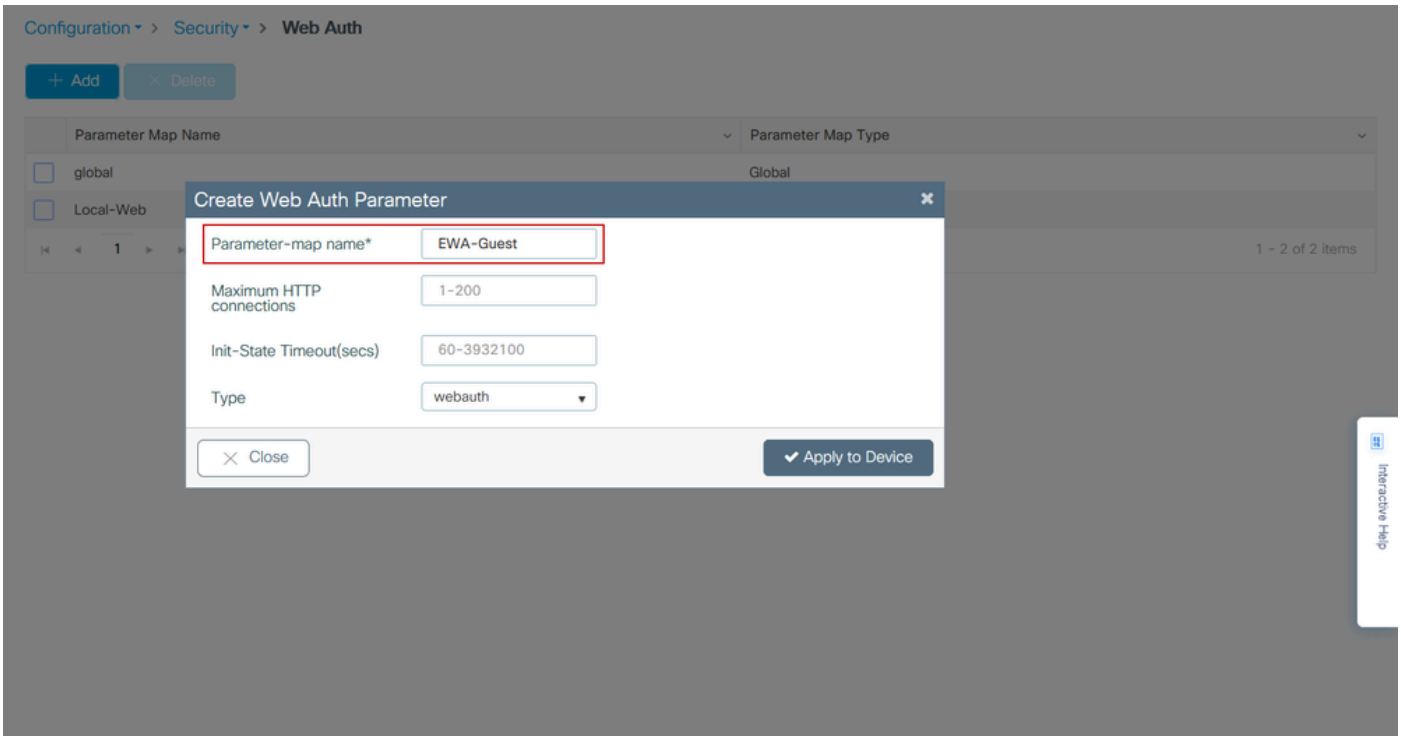
CLI 配置：

```
<#root>
9800#
configure terminal
9800(config)#
parameter-map type webauth global
9800(config-params-parameter-map)#
virtual-ip ipv4 192.0.2.1
9800(config-params-parameter-map)#
trustpoint CISCO_IDEVID_SUDI

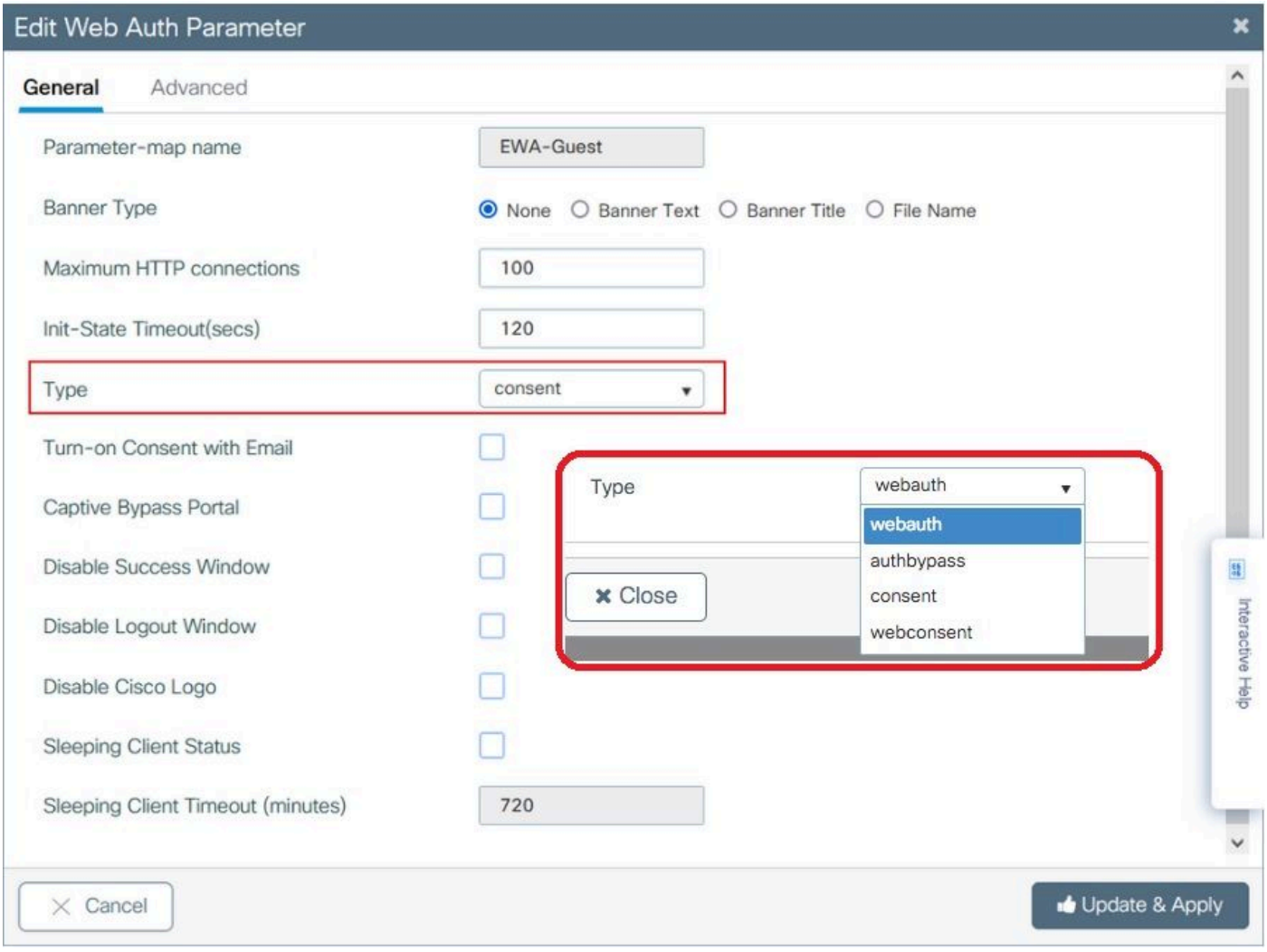
9800(config-params-parameter-map)#
secure-webauth-disable

9800(config-params-parameter-map)#
webauth-http-enable
```

步骤2.选择+添加并配置指向外部服务器的新参数映射的名称。或者，配置客户端被排除之前的最大HTTP身份验证失败次数和客户端可以保持Web身份验证状态的时间（以秒为单位）。

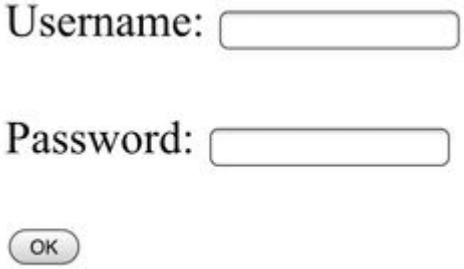
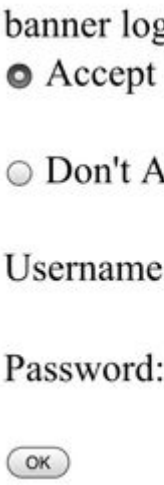


步骤3.选择新创建的参数映射，在General选项卡中，从Type下拉列表配置身份验证类型。

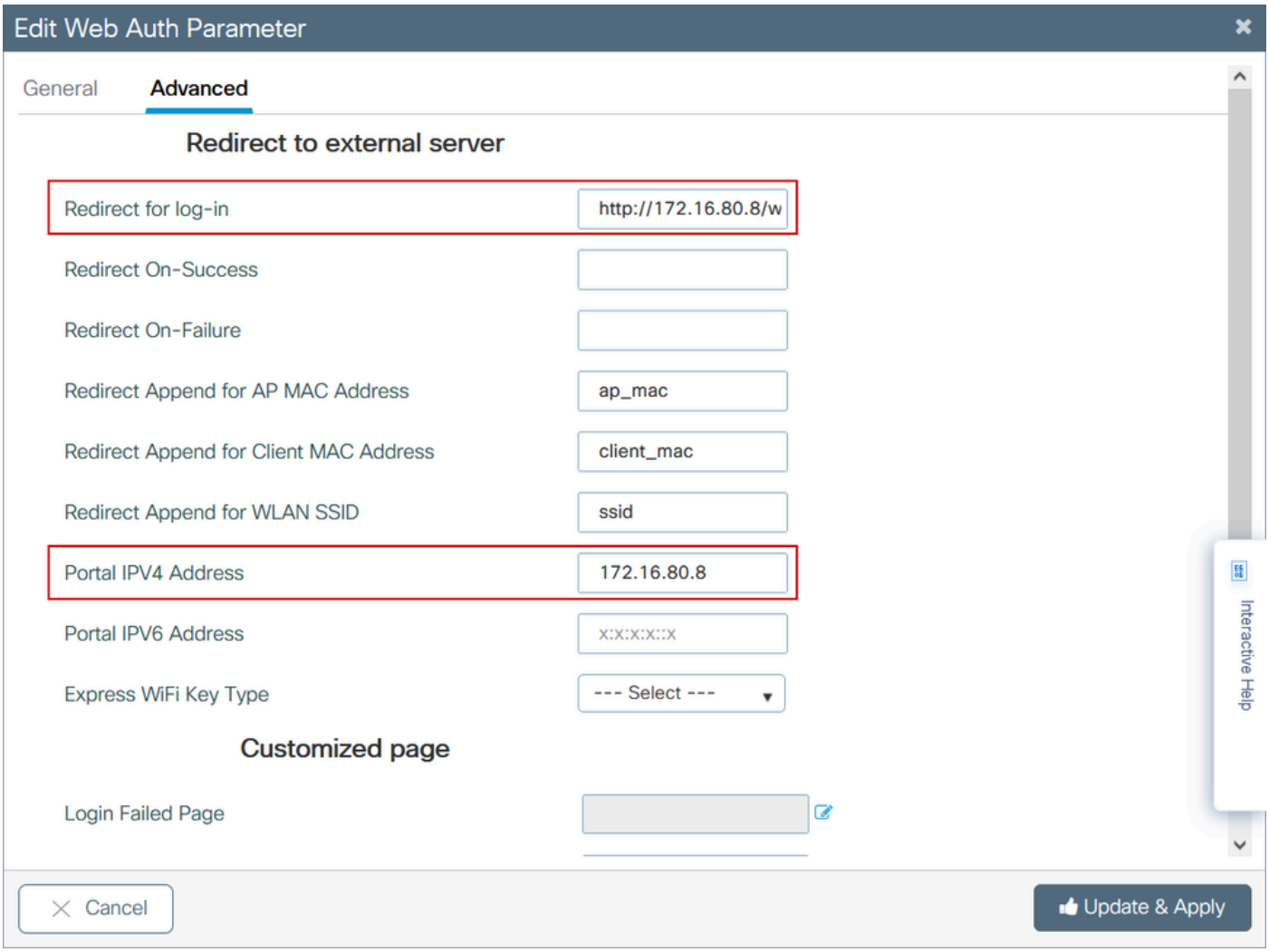


- Parameter-map name =分配给WebAuth参数映射的名称

- 最大HTTP连接数=排除客户端之前身份验证失败的次数
- Init-State Timeout(secs)=客户端可以处于Web身份验证状态的秒数
- Type = Web身份验证的类型

webauth	authbypass	同意	网络同意
	客户端连接到 SSID并获取IP地址，然后获取9800 WLC 检查MAC地址 允许输入 网络，如果是，则将其移动 至RUN状态（如果不是） 不允许加入。 （它不会回退到Web身份验证）		

步骤4.从Advanced选项卡分别使用特定服务器站点URL和IP地址配置登录和门户IPV4地址的重定向。



步骤2、3和4的CLI配置：

```
<#root>

9800(config)#

parameter-map type webauth EWA-Guest

9800(config-params-parameter-map)#

type consent

9800(config-params-parameter-map)#

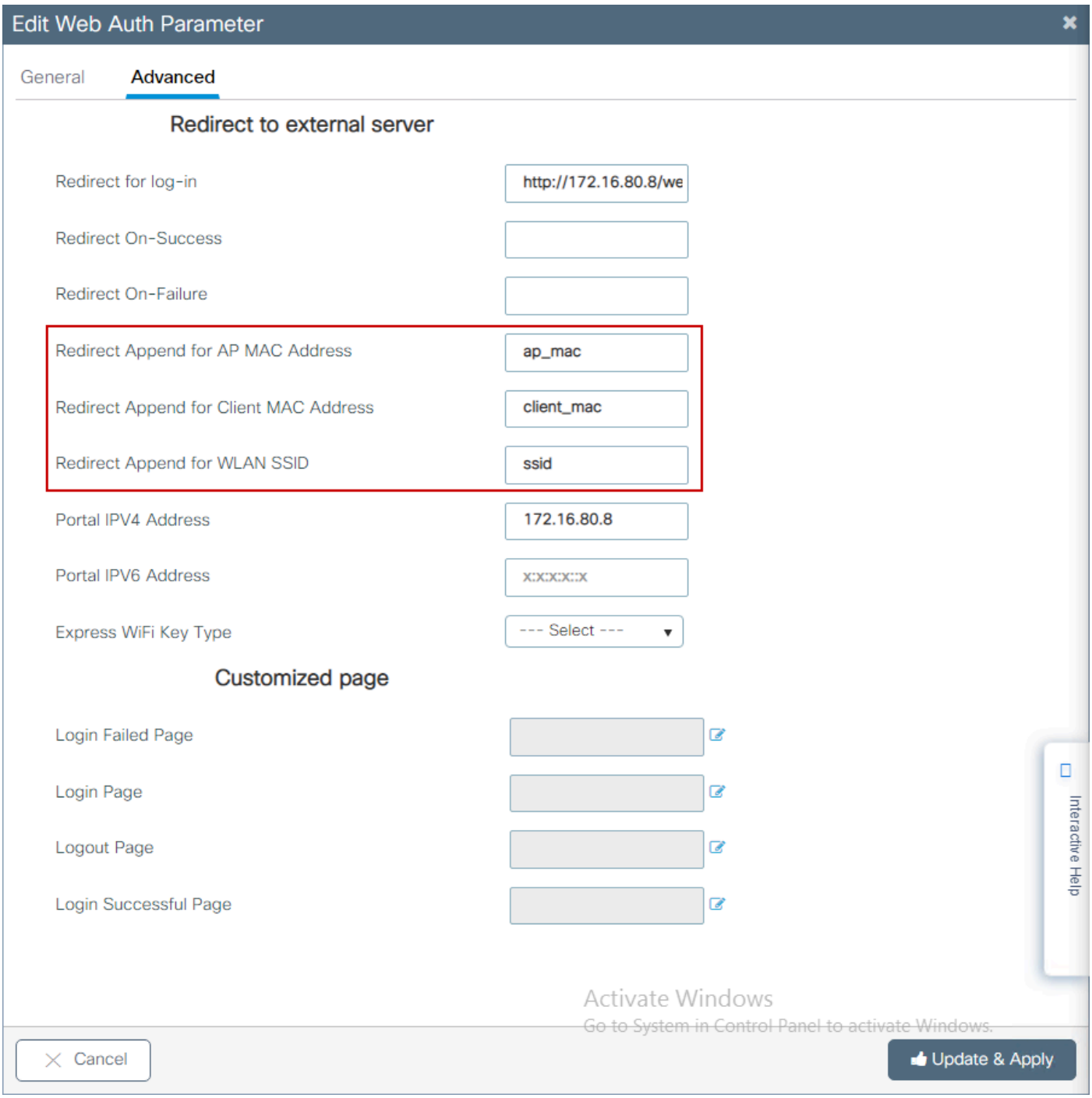
redirect for-login http://172.16.80.8/webauth/login.html

9800(config-params-parameter-map)#

redirect portal ipv4 172.16.80.8
```

步骤5.(可选)WLC可以通过查询字符串发送其他参数。这通常是使9800与第三方外部门户兼容所必需的。“为AP MAC地址重新定向追加”、“为客户端MAC地址重新定向追加”和“为WLAN SSID重新定向追加”字段允许使用自定义名称将其他参数附加到重定向ACL。选择新创建的参数映射并导航到Advanced选项卡，配置所需参数的名称。可用参数包括：

- AP MAC地址（采用aa:bb:cc:dd:ee:ff格式）
- 客户端MAC地址（采用aa:bb:cc:dd:ee:ff格式）
- SSID名称



CLI 配置：

<#root>

```
9800(config)#
```

parameter-map type webauth EWA-Guest

```
9800(config-params-parameter-map)#
```

```
redirect append ap-mac tag ap_mac

9800(config-params-parameter-map)#
redirect append wlan-ssid tag ssid

9800(config-params-parameter-map)#
redirect append client-mac tag client_mac
```

在本示例中，发送到客户端的重定向URL会导致：

```
http://172.16.80.8/webauth/consent.html?switch_url=http://192.0.2.1/login.html&ap_mac=&ssid=&client_ma
```

注：当您添加门户IPV4地址时 信息它会自动添加允许从无线客户端到外部Web身份验证服务器的HTTP和HTTPS流量的ACL，因此您不必配置任何额外的预身份验证ACL如果您要允许多个IP地址或URL，唯一的选项是配置URL过滤器，以便在进行身份验证之前允许任何IP匹配给定URL。除非使用URL过滤器，否则无法静态添加多个门户IP地址。

注意:全局参数映射是唯一可以定义虚拟IPv4和IPv6地址、Webauth拦截HTTP、强制网络旁路门户、监视列表启用和监视列表过期超时设置的映射。

CLI配置摘要：

本地Web服务器

```
parameter-map type webauth <web-parameter-map-name>
 type { webauth | authbypass | consent | webconsent }
 timeout init-state sec 300
 banner text ^Cbanner login^C
```

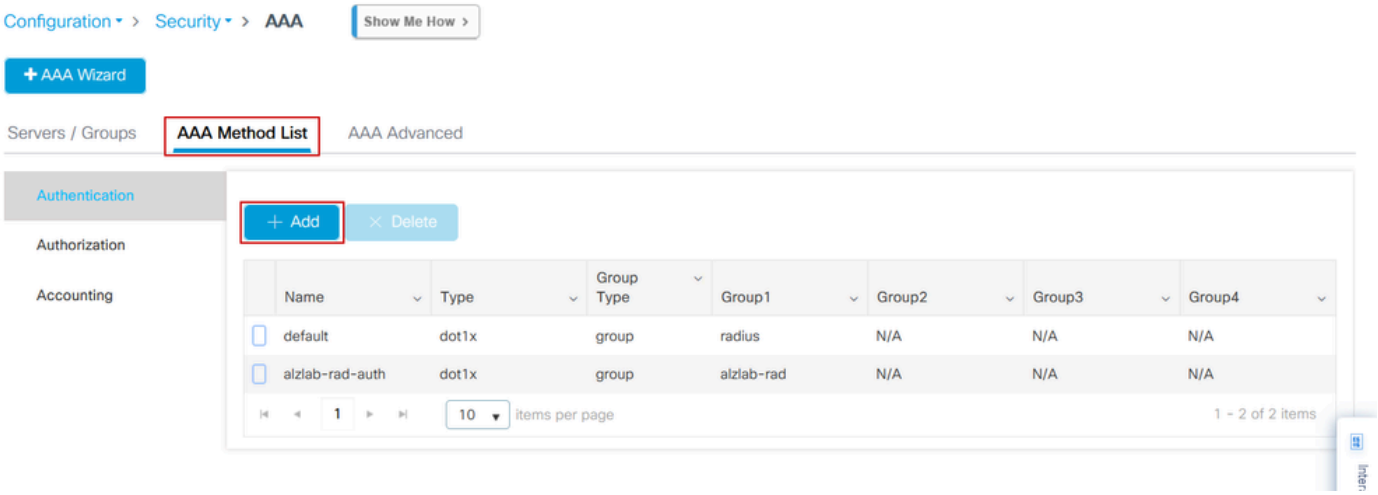
外部Web服务器

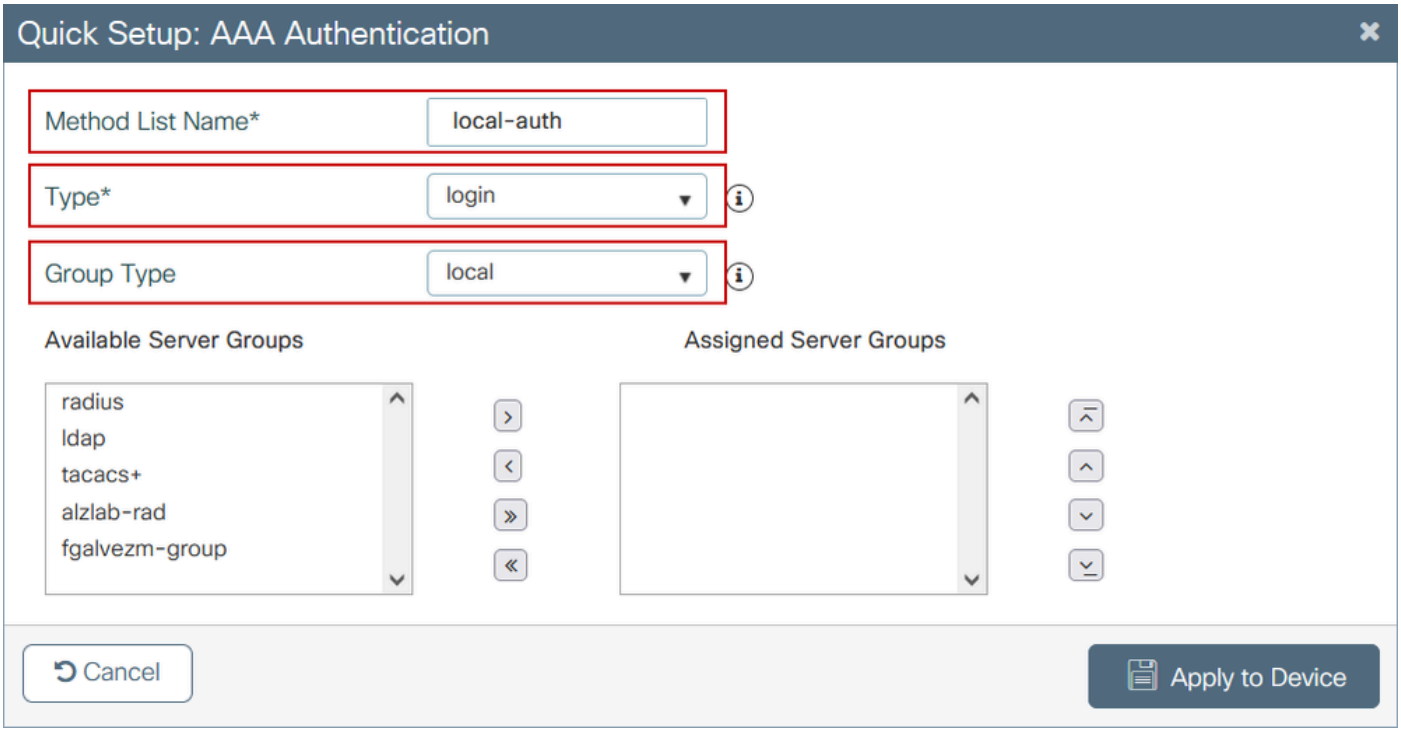
```
parameter-map type webauth <web-parameter-map-name>
 type webauth
 timeout init-state sec 300
 redirect for-login <URL-for-webauth>
 redirect portal ipv4 <external-server's-IP
 max-http-conns 10
```

配置AAA设置

此配置部分仅适用于为webauth或webconsent身份验证类型配置的参数映射。

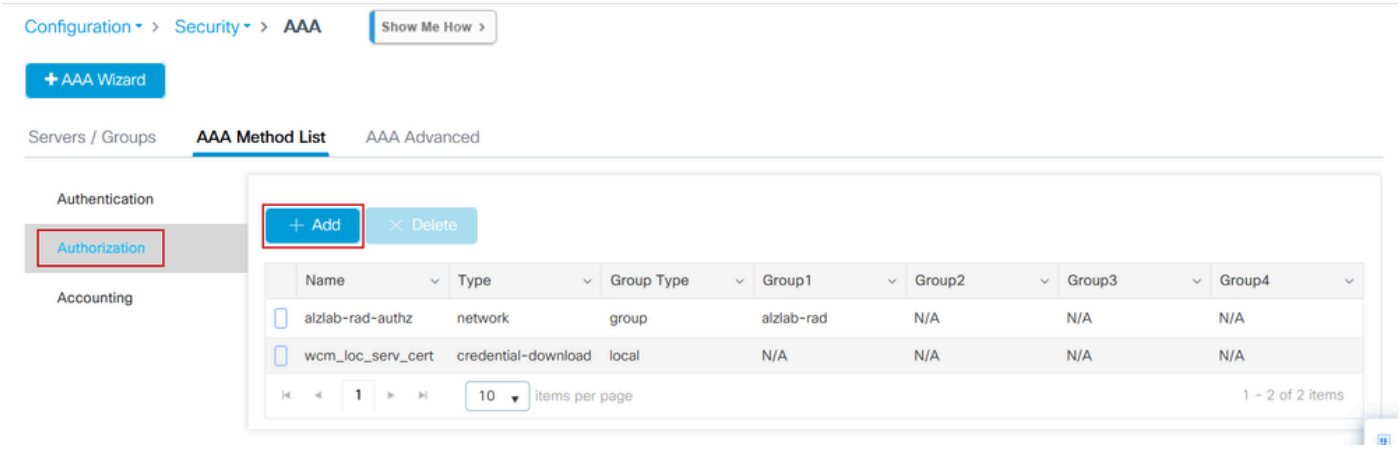
步骤1.导航到Configuration > Security > AAA，然后选择AAA Method List。配置新的方法列表，选择+ Add并填写列表详细信息；确保Type设置为“login”，如图所示。

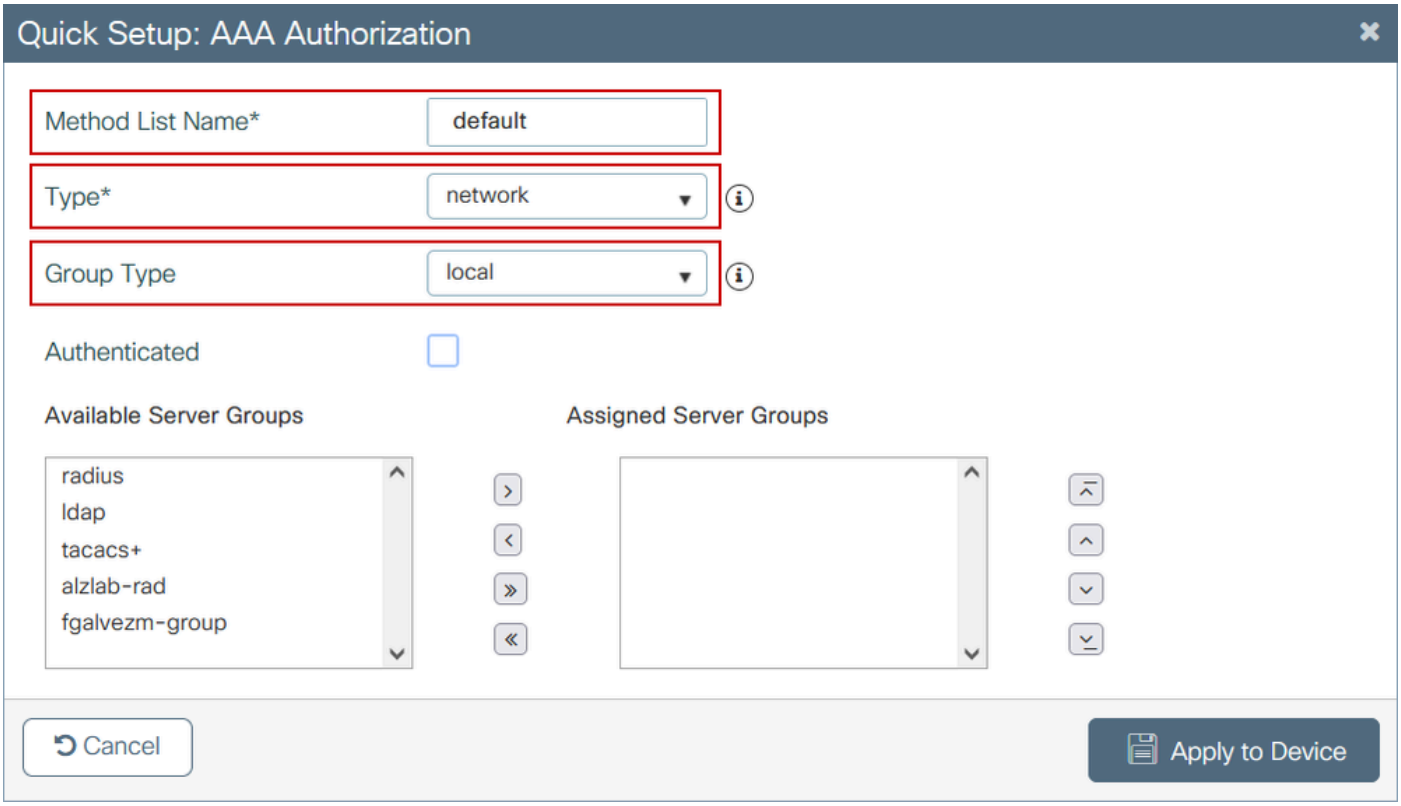




第2步：选择Authorization，然后选择+ Add以创建新方法列表。如图所示，使用网络类型将其命名为默认值。

注意：在WLAN第3层安全配置期间，控制器会通告此信息：要使Local Login Method List正常工作，请确保设备上存在“aaa authorization network default local”配置。这意味着必须定义名称为default的授权方法列表才能正确配置本地Web身份验证。在本节中，配置此特定授权方法列表。





步骤1和2的CLI配置：

<#root>

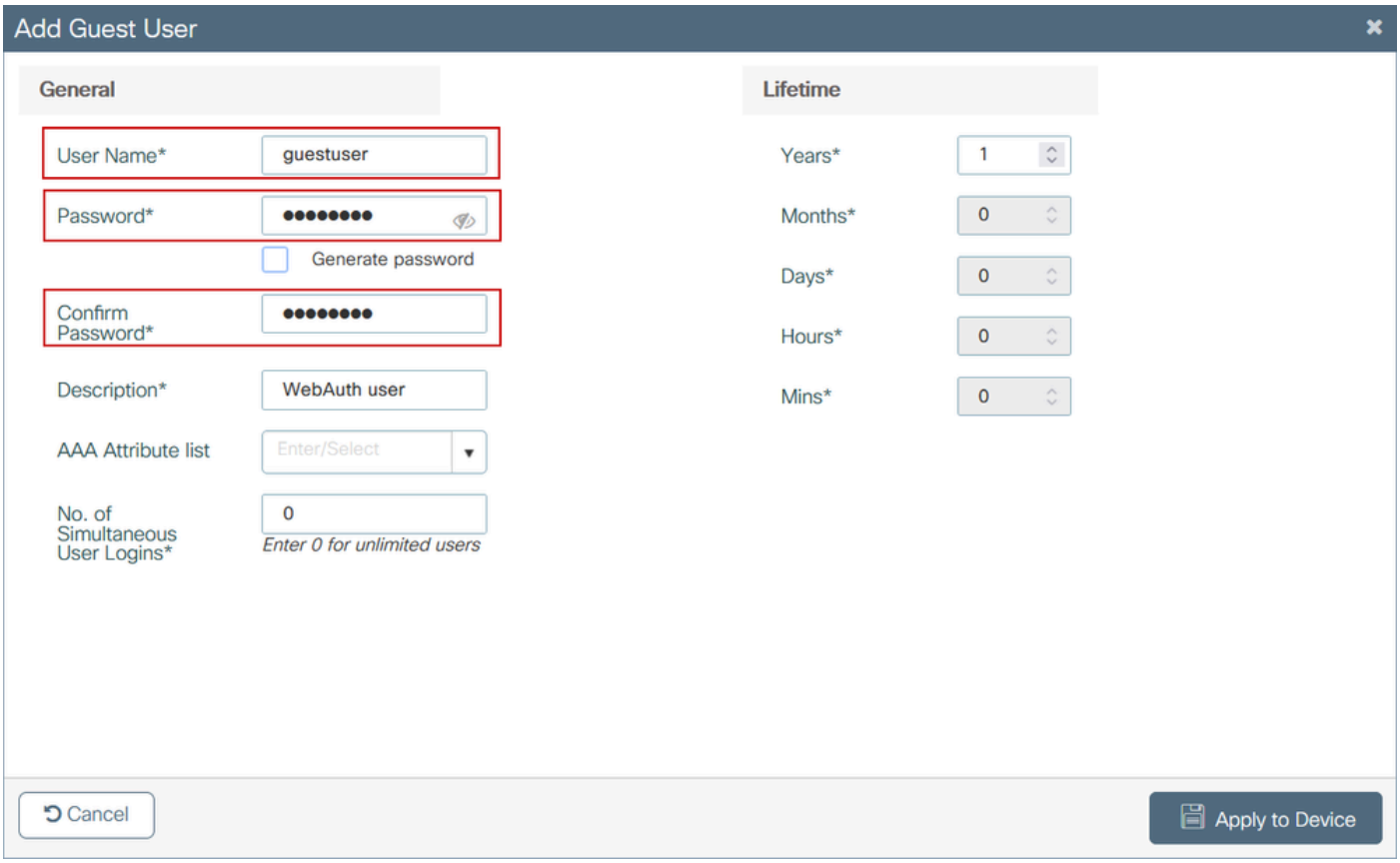
```
9800(config)#
aaa new-model

9800(config)#
aaa authentication login local-auth local

9800(config)#
aaa authorization network default local
```

注意：如果需要外部RADIUS身份验证，请阅读以下与9800 WLC上的RADIUS服务器配置相关的说明：9800 WLC上的AAA配置。确保身份验证方法列表将“login”设置为type而不是dot1x。

步骤3.导航到配置>安全>访客用户。选择+ Add并配置访客用户帐户详细信息。



CLI 配置：

```
<#root>
9800(config)#
user-name guestuser

9800(config-user-name)#
description "WebAuth user"

9800(config-user-name)#
password 0



9800(config-user-name)#
type network-user description "WebAuth user" guest-user lifetime year 1
```

```
If permanent users are needed then use this command:
9800(config)#
```

username guestuserperm privilege 0 secret 0

步骤4.（可选）根据参数映射定义，自动创建多个访问控制列表(ACL)。这些ACL用于定义哪些流量触发重定向到Web服务器，哪些流量允许通过。如果存在特定要求（例如多个Web服务器IP地址或URL过滤器），则导航到Configuration > Security > ACL select + Add并定义必要的规则；允许语句被重定向，而拒绝语句定义流量通过。

自动创建的ACL规则如下：

<#root>

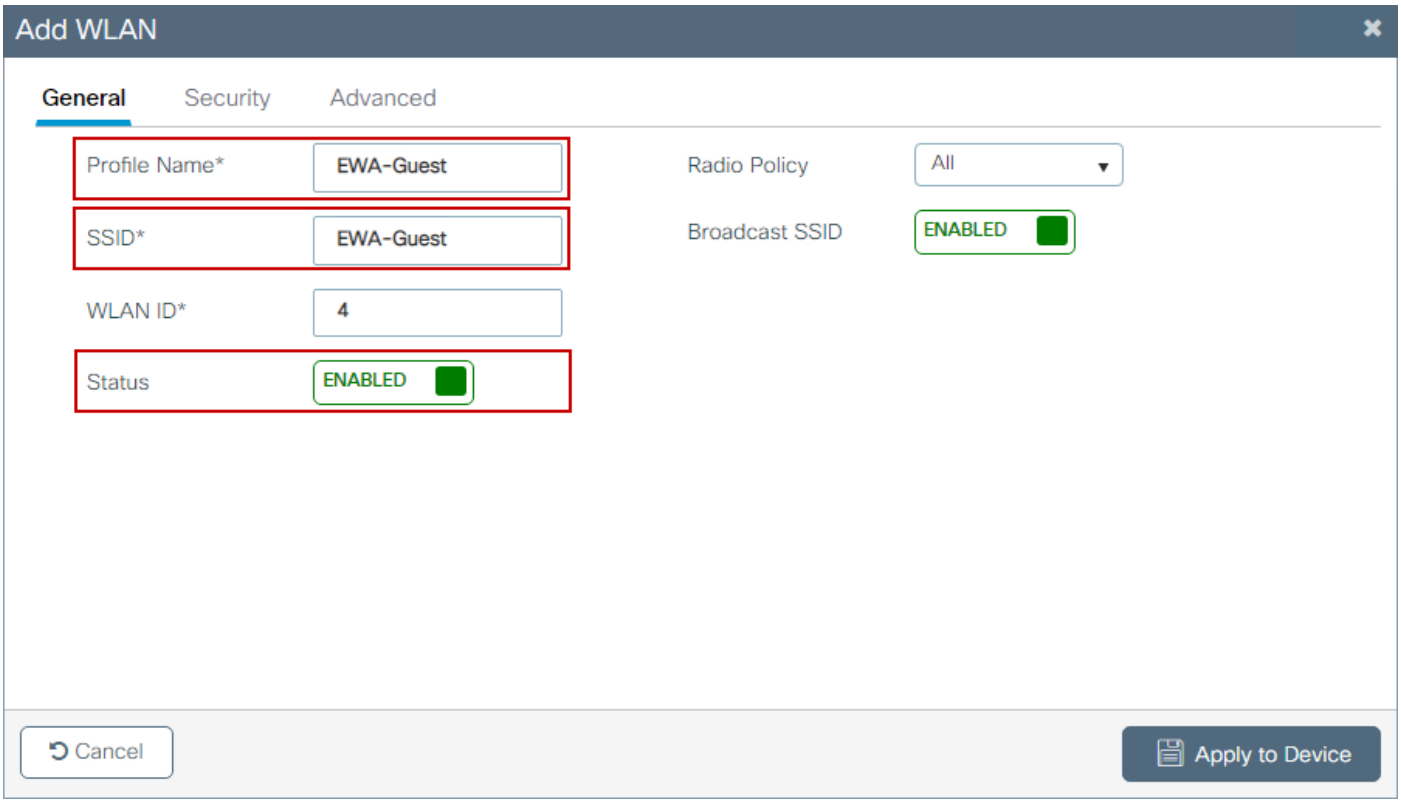
```
alz-9800#
```

show ip access-list

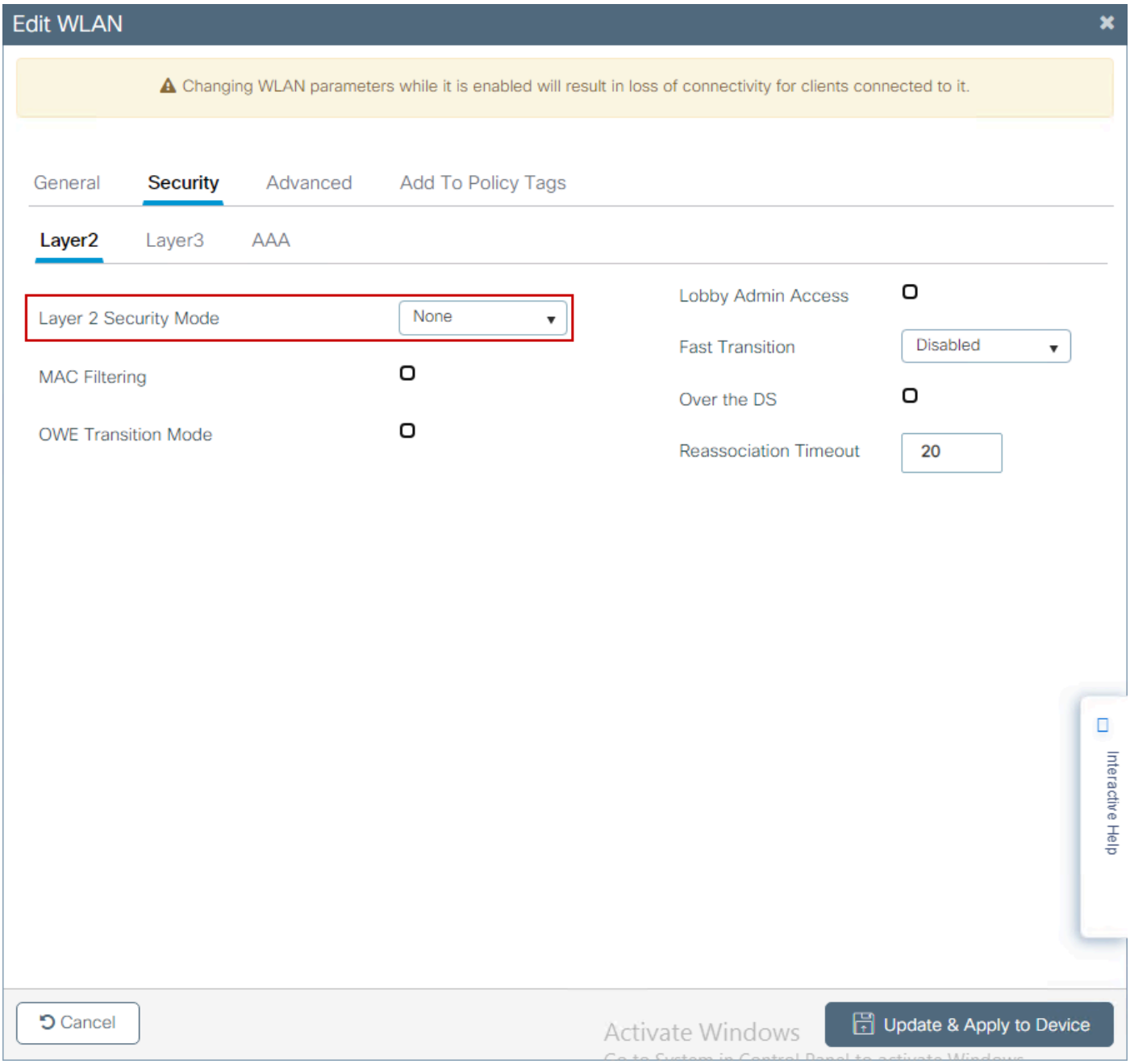
```
Extended IP access list WA-sec-172.16.80.8
10 permit tcp any host 172.16.80.8 eq www
20 permit tcp any host 172.16.80.8 eq 443
30 permit tcp host 172.16.80.8 eq www any
40 permit tcp host 172.16.80.8 eq 443 any
50 permit tcp any any eq domain
60 permit udp any any eq domain
70 permit udp any any eq bootpc
80 permit udp any any eq bootps
90 deny ip any any (1288 matches)
Extended IP access list WA-v4-int-172.16.80.8
10 deny tcp any host 172.16.80.8 eq www
20 deny tcp any host 172.16.80.8 eq 443
30 permit tcp any any eq www
40 permit tcp any host 192.0.2.1 eq 443
```

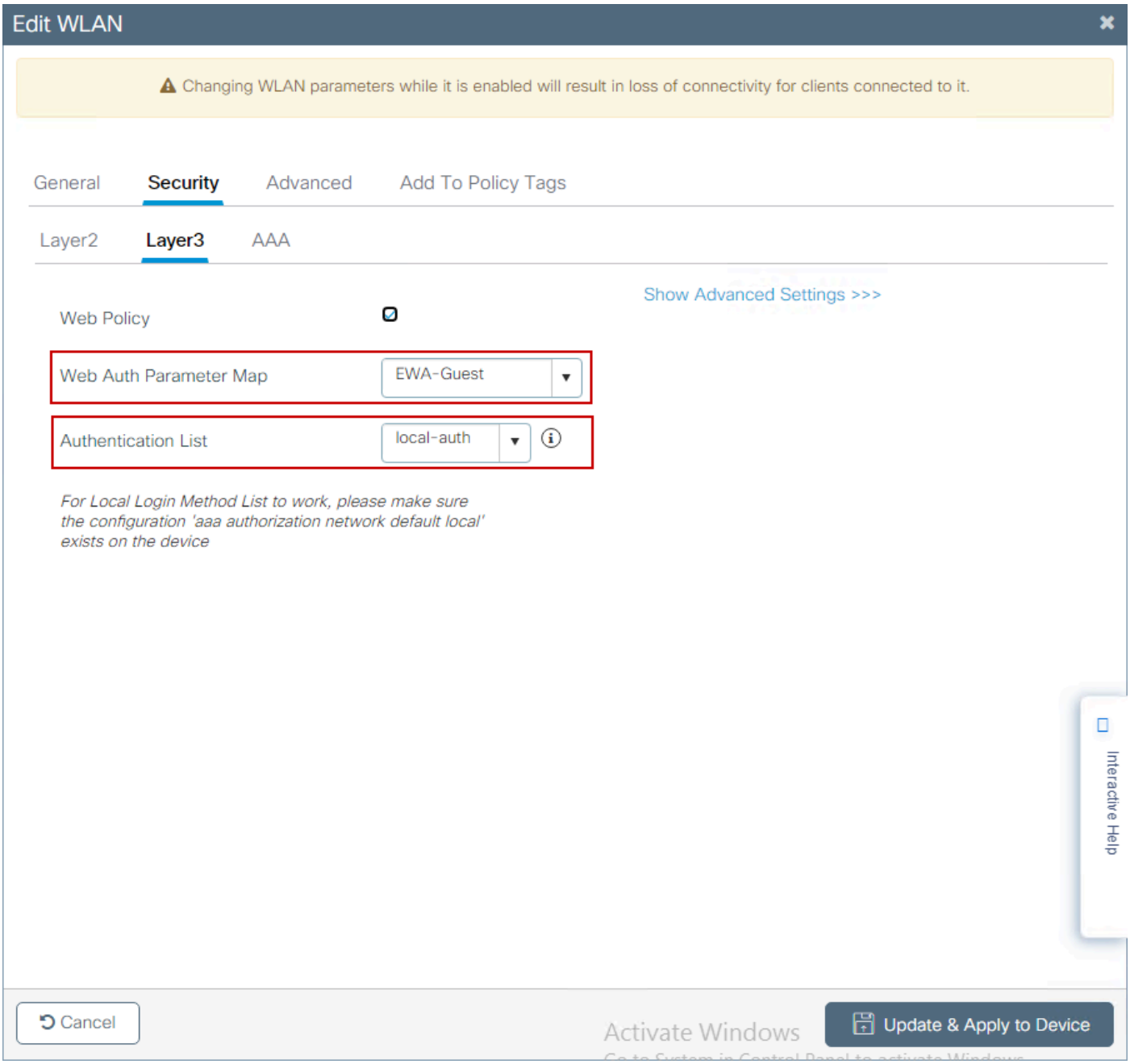
配置策略和标记

第1步：导航到Configuration > Tags & Profiles > WLANs，选择+ Add以创建新的WLAN。在General选项卡中定义配置文件和SSID名称以及状态。



步骤2.选择Security选项卡，如果不需要任何空中加密机制，则将Layer 2 authentication设置为None。在Layer 3选项卡中，选中Web Policy框，从下拉菜单中选择参数映射，然后从下拉菜单中选择身份验证列表。或者，如果之前定义了自定义ACL，请选择Show Advanced Settings，然后从下拉菜单中选择适当的ACL。


Edit WLAN
Changing WLAN parameters while it is enabled will result in loss of connectivity for clients connected to it.
General
Security
Advanced
Add To Policy Tags
Layer2
Layer3
AAA
Layer 2 Security Mode
None
MAC Filtering
OWE Transition Mode
Lobby Admin Access
Fast Transition
Disabled
Over the DS
Reassociation Timeout
20
Interactive Help
Cancel
Activate Windows
Update & Apply to Device



CLI配置：

<#root>

```
9800(config)#
wlan EWA-Guest 4 EWA-Guest

9800(config-wlan)#
no security ft adaptive

9800(config-wlan)#
no security wpa

9800(config-wlan)#
```

```
no security wpa wpa2

9800(config-wlan)#
no security wpa wpa2 ciphers aes

9800(config-wlan)#
no security wpa akm dot1x

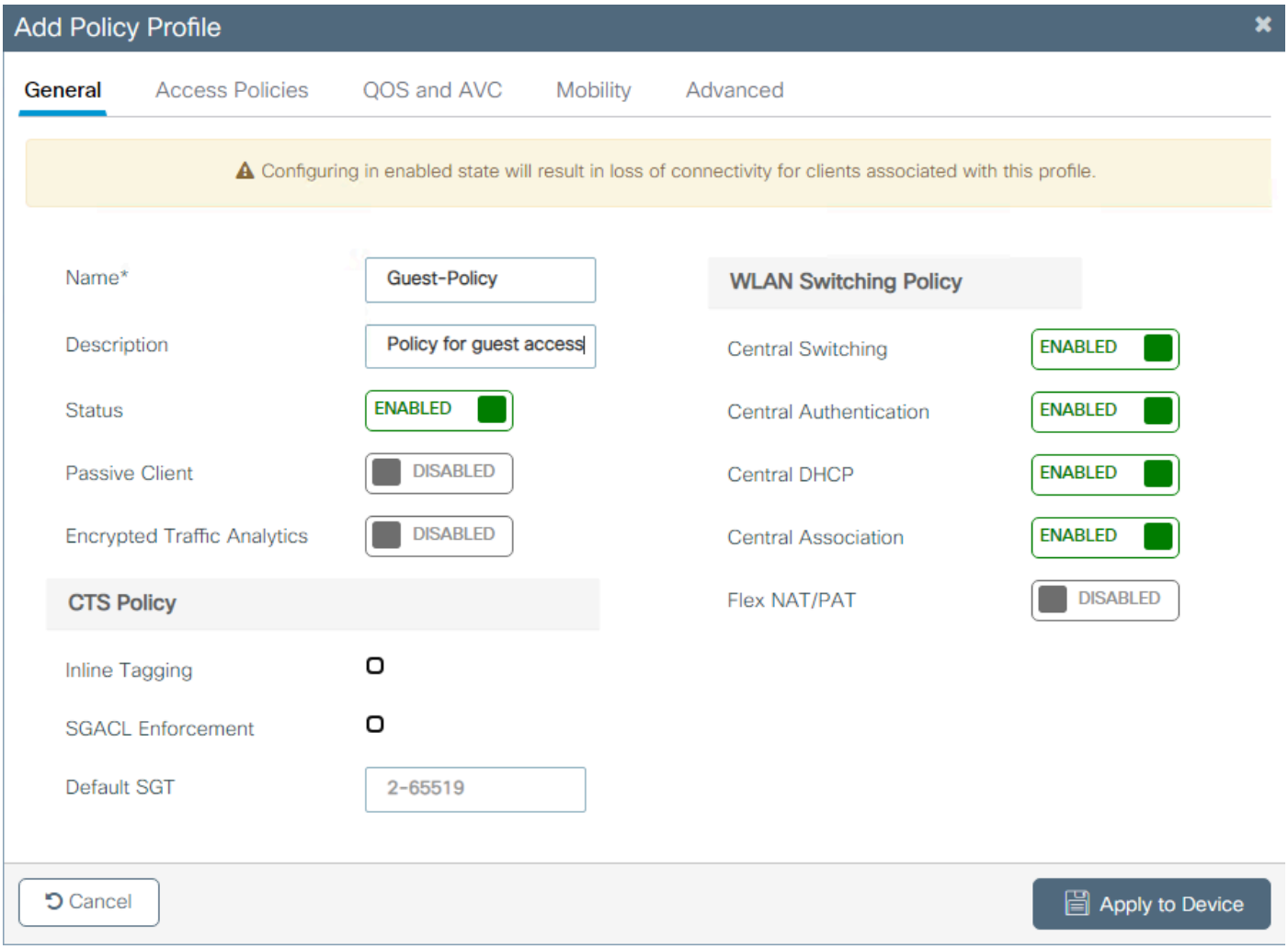
9800(config-wlan)#
security web-auth

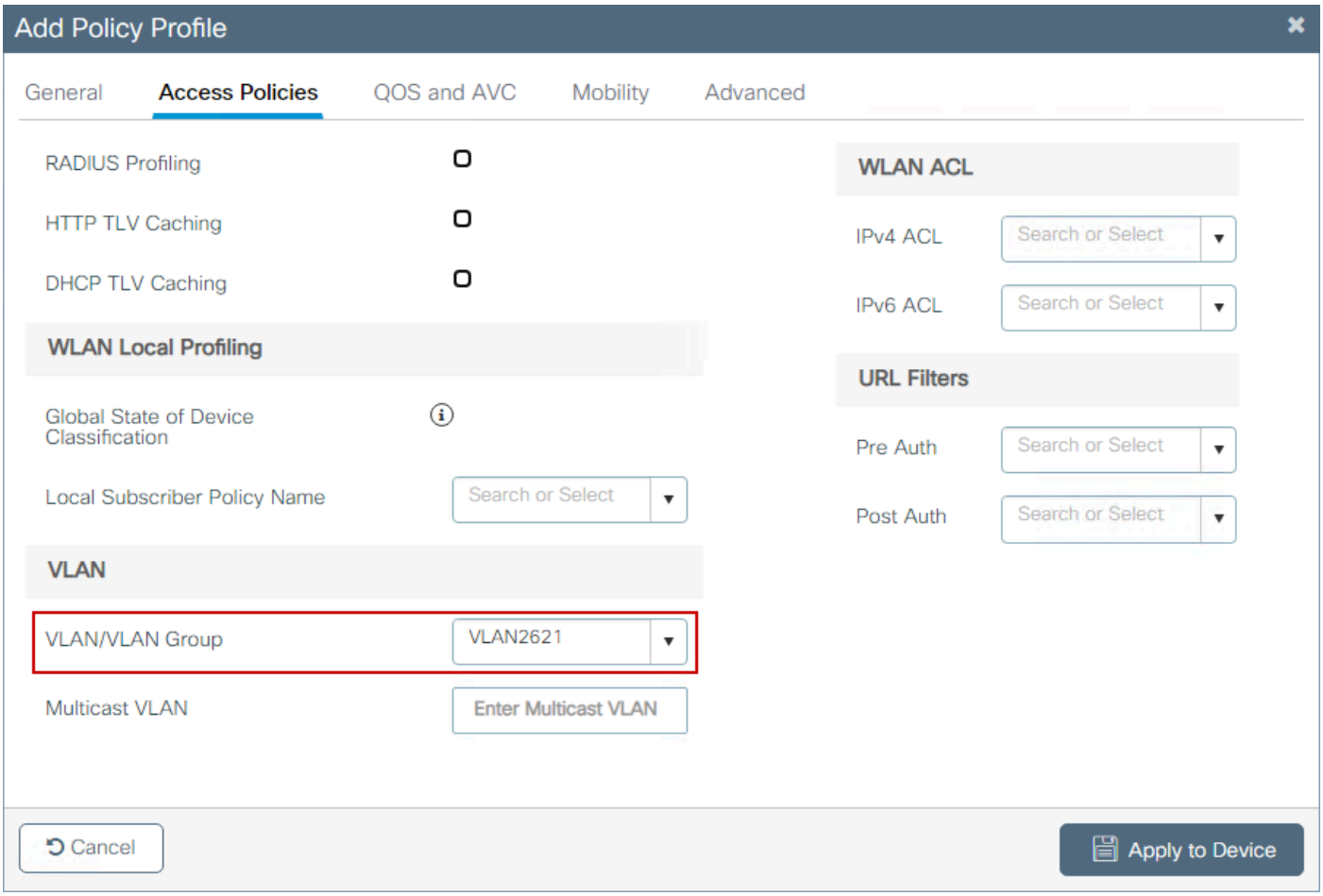
9800(config-wlan)#
security web-auth authentication-list local-auth

9800(config-wlan)#
security web-auth parameter-map EWA-Guest

9800(config-wlan)#
 no shutdown
```

第3步：导航至配置>标记和配置文件>策略，然后选择+ Add。定义策略名称和状态；确保WLAN Switching Policy（WLAN交换策略）下的Central settings（中央设置）已为本地模式AP启用。在访问策略选项卡中，从VLAN/VLAN组下拉菜单中选择正确的VLAN，如图所示。


Add Policy Profile
General
Access Policies
QOS and AVC
Mobility
Advanced
Configuring in enabled state will result in loss of connectivity for clients associated with this profile.
Name*
Guest-Policy
Description
Policy for guest access
Status
ENABLED
Passive Client
DISABLED
Encrypted Traffic Analytics
DISABLED
CTS Policy
Inline Tagging
SGACL Enforcement
Default SGT
2-65519
WLAN Switching Policy
Central Switching
ENABLED
Central Authentication
ENABLED
Central DHCP
ENABLED
Central Association
ENABLED
Flex NAT/PAT
DISABLED
Cancel
Apply to Device



CLI 配置：

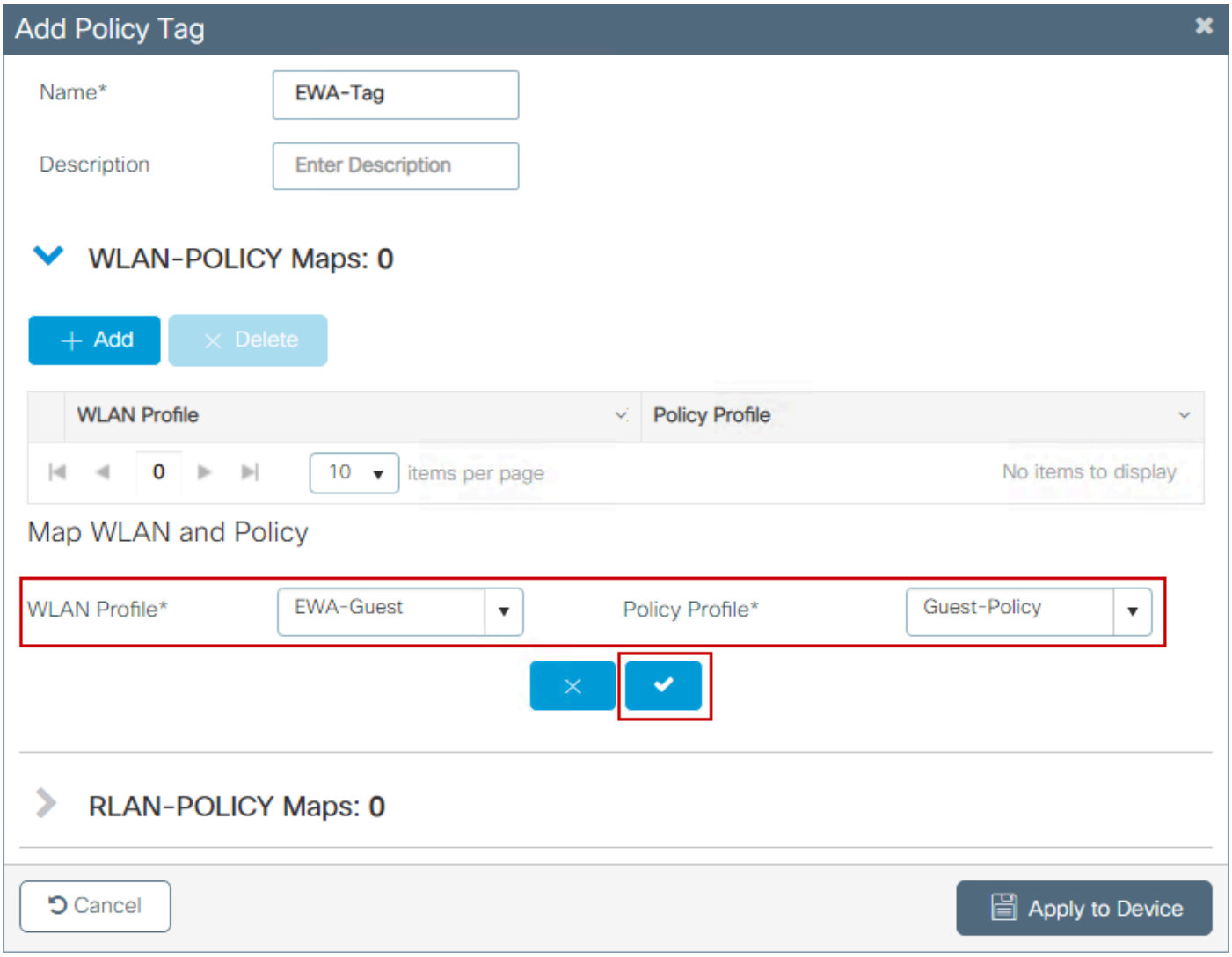
```
<#root>
9800(config)#
wireless profile policy Guest-Policy

9800(config-wireless-policy)#
description "Policy for guest access"

9800(config-wireless-policy)#
vlan VLAN2621

9800(config-wireless-policy)#
no shutdown
```

第4步：导航至Configuration > Tags & Profiles > Tags，在Policy选项卡中选择+ Add。定义标记名称，然后在WLAN-POLICY Maps下选择+ Add并添加之前创建的WLAN和策略配置文件。



CLI 配置：

<#root>

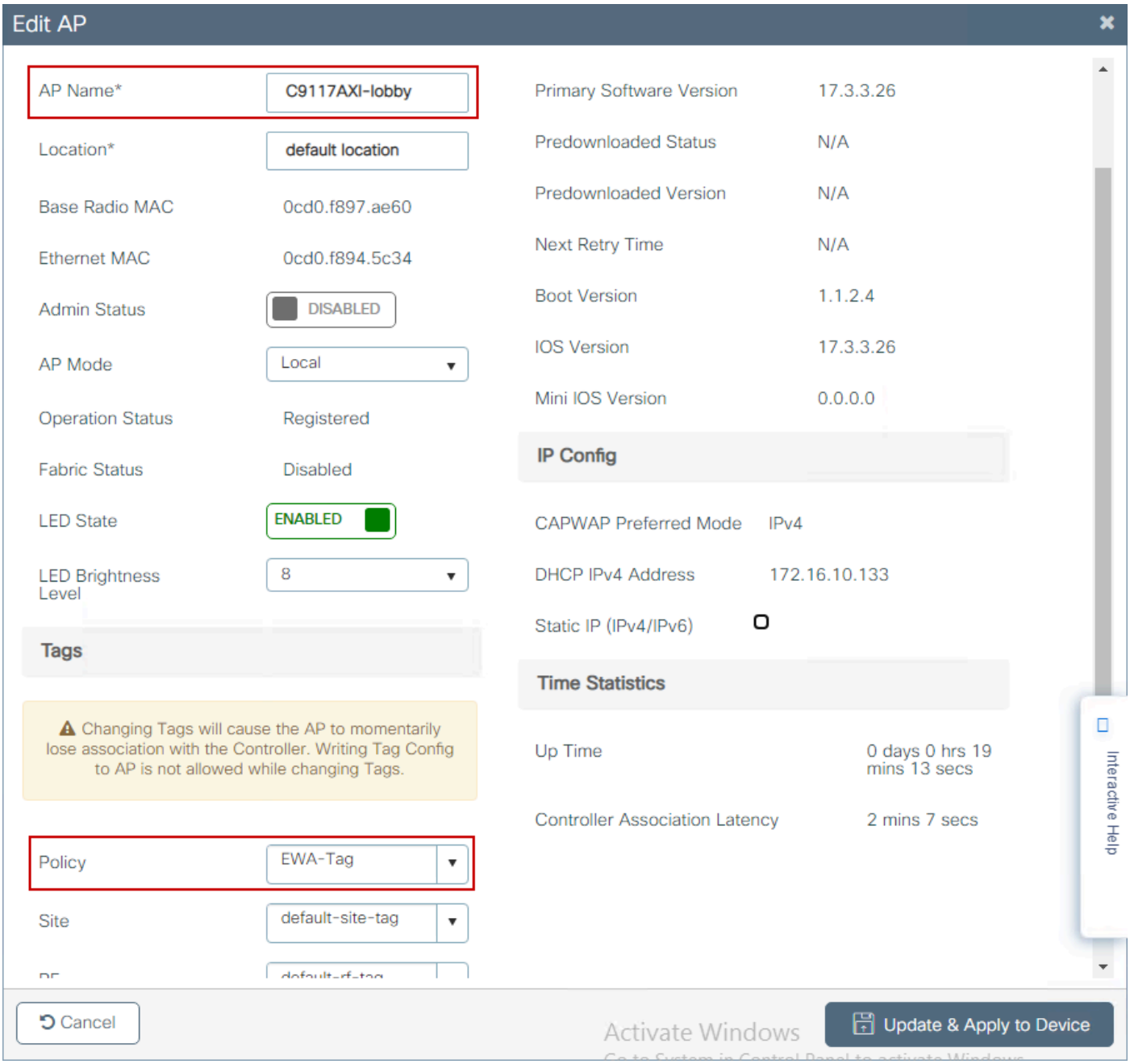
```
9800(config)#
```

`wireless tag policy EWA-Tag`

```
9800(config-policy-tag)#
```

`wlan EWA-Guest policy Guest-Policy`

步骤5.导航到Configuration > Wireless > Access Points，然后选择用于广播此SSID的AP。从Edit AP菜单中，从Policy下拉菜单中选择新创建的标记。

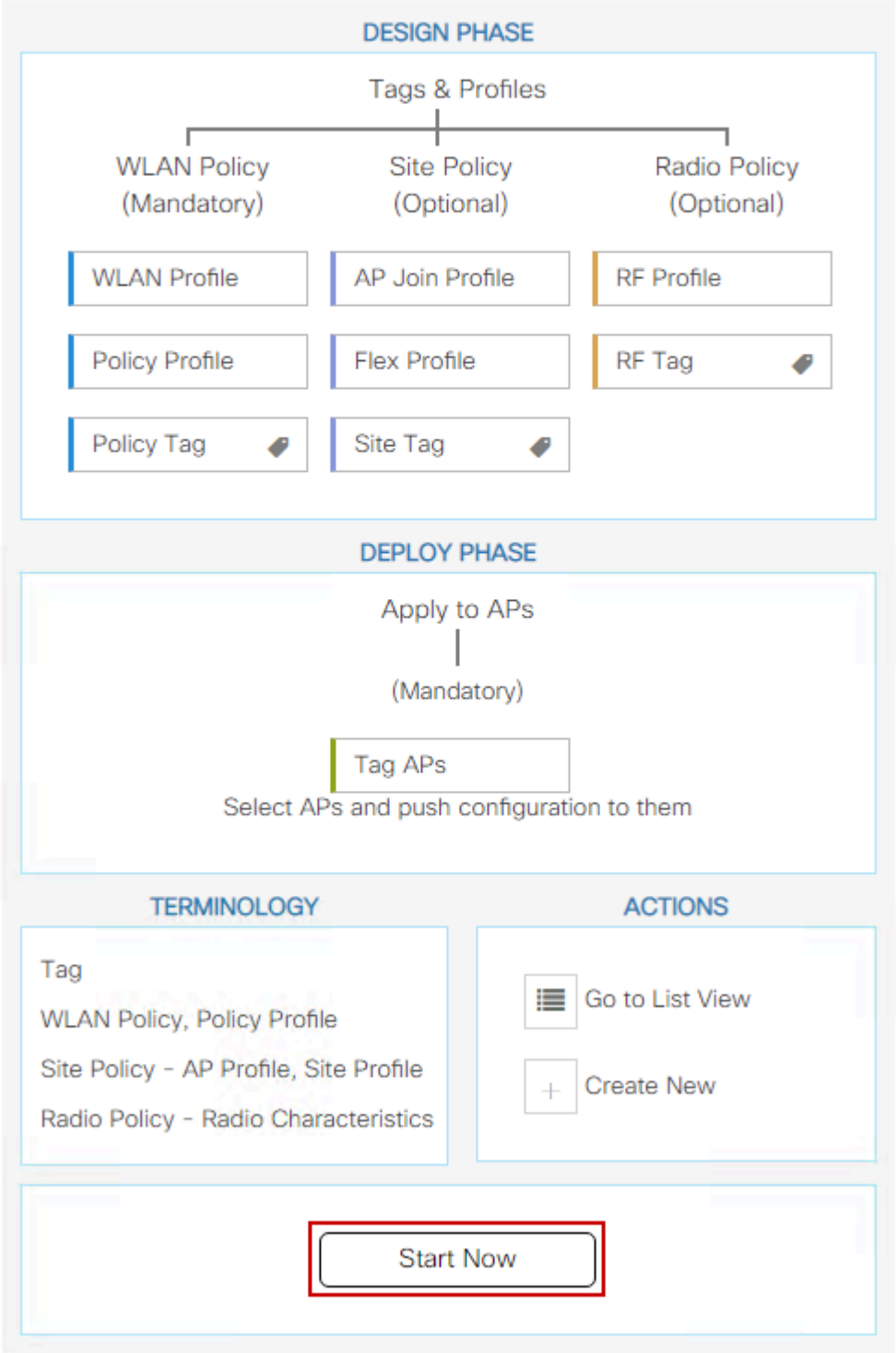


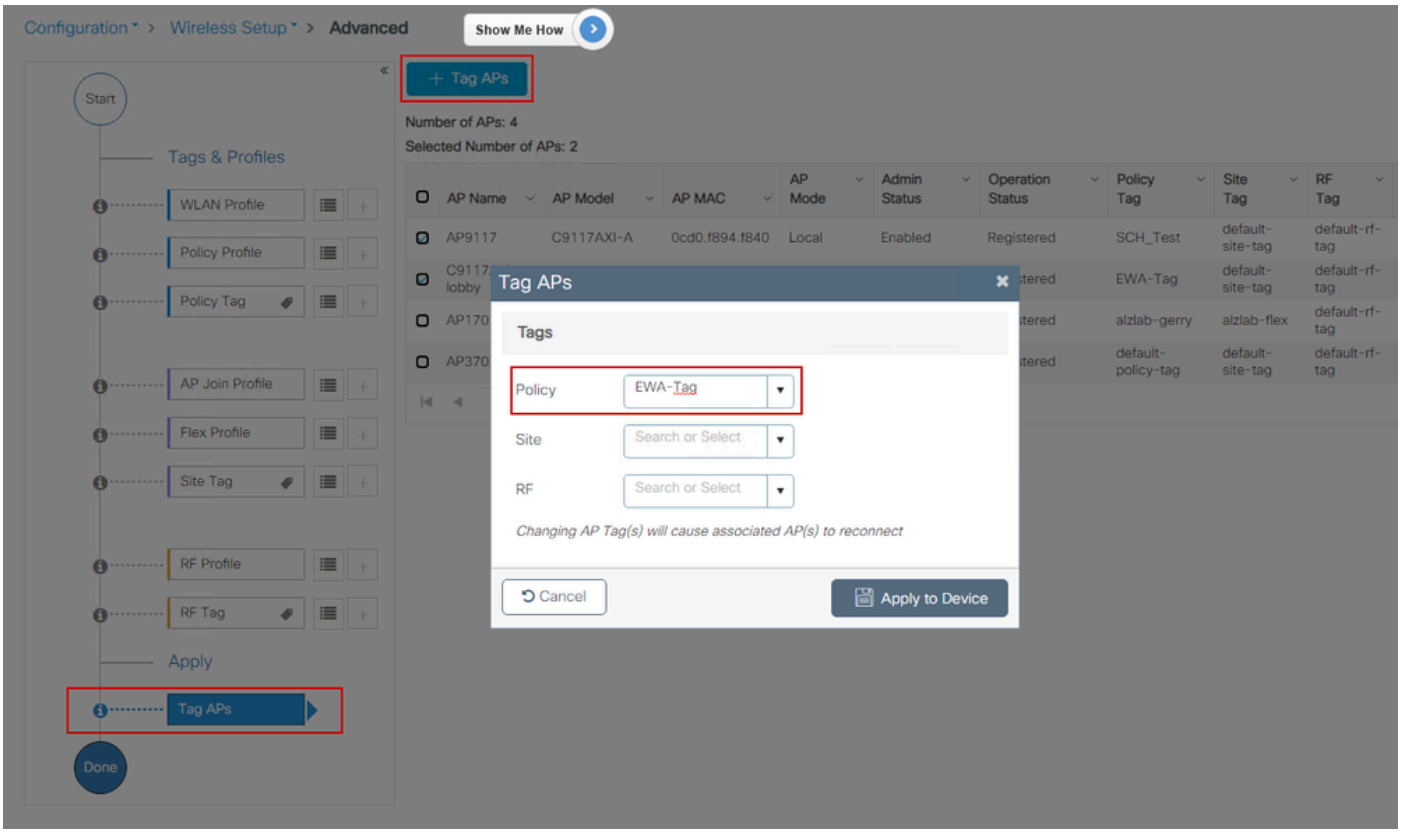
如果需要同时标记多个AP，则有两个可用选项：

选项A。从此处导航到配置>无线设置>高级，选择立即启动以显示配置菜单列表。选择标记AP旁边的列表图标，这会显示处于加入状态的所有AP的列表，检查必要的AP，然后选择+ Tag AP，从下拉菜单中选择创建的策略标记。

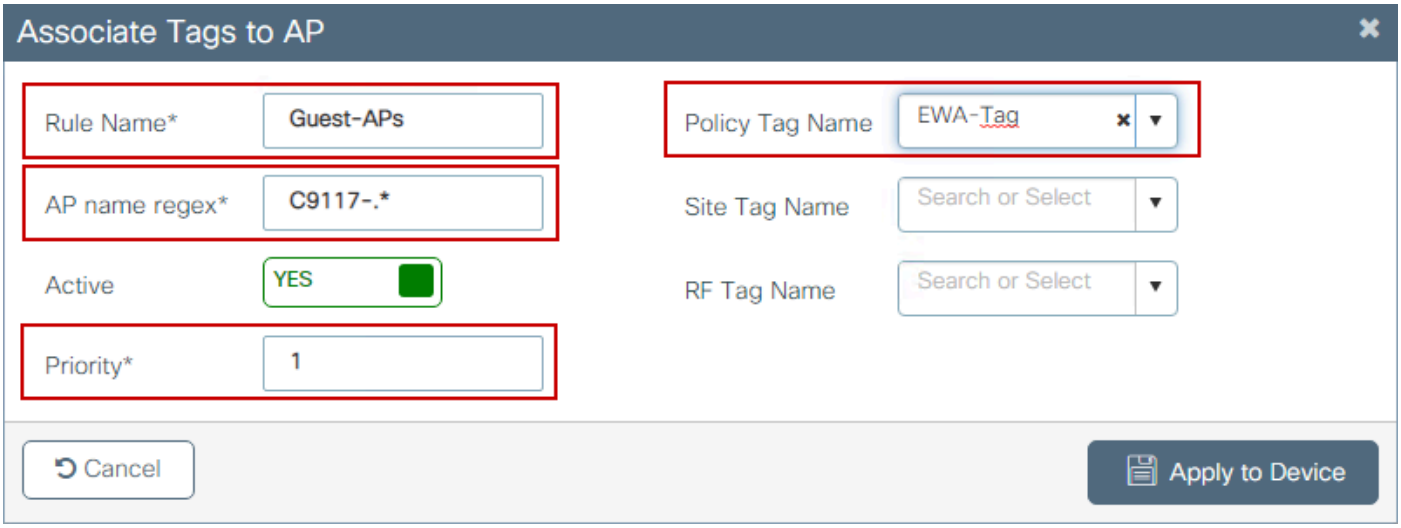
Wireless Setup Flow Overview

This screen allows you to design Wireless LAN Configuration. It involves creating Policies and Tags. Once the design is completed, they can be deployed to the Access Points right here.





选项B.导航至配置>标记和配置文件>标记。然后选择AP选项卡。在Filter（过滤器）选项卡中，选择+ Add。定义规则名称、AP名称regex（此设置允许控制器定义标记的AP）、优先级（较低数字具有更高的优先级）和必要的标记。



验证

参阅本部分来确认配置是否正常运行:

<#root>

9800#

show running-config wlan

```
9800#
show running-config aaa
9800#
show aaa servers

9800#
show ap tag summary

9800#
show ap name


        config general


9800#
show ap name


        tag detail


9800#
show wlan [summary | id | name | all]

9800#
show wireless tag policy detailed


9800#
show wireless profile policy detailed
```

使用show ip http server status验证http服务器状态和可用性：

<#root>

9800#

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 active session modules: ALL
HTTP secure server capability: Present

HTTP secure server status: Enabled

HTTP secure server port: 443

HTTP secure server ciphersuite: rsa-aes-cbc-sha2 rsa-aes-gcm-sha2
dhe-aes-cbc-sha2 dhe-aes-gcm-sha2 ecdhe-rsa-aes-cbc-sha2
ecdhe-rsa-aes-gcm-sha2 ecdhe-ecdsa-aes-gcm-sha2
HTTP secure server TLS version: TLSv1.2 TLSv1.1
HTTP secure server client authentication: Disabled
HTTP secure server PIV authentication: Disabled
HTTP secure server PIV authorization only: Disabled

HTTP secure server trustpoint: CISCO_IDEVID_SUDI

HTTP secure server peer validation trustpoint:
HTTP secure server ECDHE curve: secp256r1
HTTP secure server active session modules: ALL

使用以下命令检验ACL是否可连接到客户端会话：

<#root>

9800#

show platform software wireless-client chassis active R0 mac-address

ID : 0xa0000002
MAC address : aaaa.bbbb.cccc
Type : Normal
Global WLAN ID : 4

SSID : EWA-Guest

Client index : 0
Mobility state : Local

Authentication state : L3 Authentication

VLAN ID : 2621
[...]
Disable IPv6 traffic : No

Dynamic policy template : 0x7b 0x73 0x0b 0x1e 0x46 0x2a 0xd7 0x8f 0x23 0xf3 0xfe 0x9e 0x5c 0xb0 0xeb 0x

9800#

show platform software cgacl chassis active F0

Template ID

Group Index

Lookup ID Number of clients

0x7B 0x73 0x0B 0x1E 0x46 0x2A 0xD7 0x8F 0x23 0xF3 0xFE 0x9E 0x5C 0xB0 0xEB 0xF8 0x0000000a

0x0000001a 1

9800#

show platform software cgacl chassis active F0 group-idx

acl

Acl ID Acl Name CGACL Type Protocol Direction Sequence
---
16 IP-Adm-V6-Int-ACL-global Punt IPv6 IN 1

25 WA-sec-172.16.80.8 Security IPv4 IN 2

```
26 WA-v4-int-172.16.80.8 Punt IPv4 IN 1


19 implicit_deny Security IPv4 IN 3
21 implicit_deny_v6 Security IPv6 IN 3
18 preauth_v6 Security IPv6 IN 2
```

故障排除

永远在线跟踪

WLC 9800提供无间断跟踪功能。这可确保持续记录所有客户端连接相关的错误、警告和通知级别消息，并且您可以在发生事故或故障情况后查看其日志。

注意：根据生成的日志量，您可以从几个小时返回几天。

为了查看9800 WLC在默认情况下收集的跟踪，您可以通过SSH/Telnet连接到9800 WLC并阅读以下步骤（确保您将会话记录到文本文件）。

步骤1.检查控制器当前时间，以便您可以在问题发生之前的时间跟踪日志。

<#root>

9800#

show clock

步骤2.根据系统配置的指示，从控制器缓冲区或外部系统日志收集系统日志。这样可以快速查看系统运行状况和错误（如果有）。

<#root>

9800#

show logging

第 3 步： 验证是否已启用调试条件。

<#root>

9800#

show debugging

```
IOSXE Conditional Debug Configs:
Conditional Debug Global State: Stop
```

```
IOSXE Packet Tracing Configs:
Packet Infra debugs:
Ip Address                                              Port
------------------------------------------------------|----------
```

注意：如果看到列出了任何条件，则意味着遇到启用条件（mac地址、IP地址等）的所有进程的跟踪将记录到调试级别。 这会增加日志量。因此，建议在非主动调试时清除所有条件.

步骤4.假设测试的mac地址未作为条件列在步骤3中。收集特定mac地址的始终在线通知级跟踪。

<#root>

9800#

show logging profile wireless filter [mac | ip] [

|

] to-file always-on-

您可以显示会话内容，也可以将文件复制到外部 TFTP 服务器。

<#root>

9800#

more bootflash:always-on-

or
9800#

copy bootflash:always-on-

```
tftp://

/

 /always-on-
```

条件调试和无线电主动跟踪

如果永远在线(always-on)跟踪无法为您提供足够的信息来确定所调查问题的触发器，则可以启用条件调试并捕获无线活动(RA)跟踪，该跟踪为与指定条件（本例中为客户端MAC地址）交互的所有进程提供调试级别跟踪。 要启用条件调试，请阅读以下步骤。

第 1 步： 确保未启用调试条件。

```
<#root>

9800#

clear platform condition all
```

第 2 步： 为要监控的无线客户端 MAC 地址启用调试条件。

这些命令用于开始监控所提供的 MAC 地址，持续 30 分钟（1800 秒）。 您可以选择延长监控时间，最多监控 2085978494 秒。

```
<#root>

9800#
```

debug wireless mac

{monitor-time

}

注意：要一次监控多个客户端，请对每个mac地址运行debug wireless mac命令。

注意：无线客户端活动不会显示在终端会话上，因为所有日志都在内部缓冲，以便稍后查看。

第 3 步： 重现要监控的问题或行为。

第 4 步： 如果在默认或配置的监控时间结束之前，问题再次出现，请停止调试。

<#root>

9800#

no debug wireless mac

监控时间结束或无线网络调试停止后，9800 WLC 会生成一个本地文件，其名称为：

ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log

步骤5.收集mac地址活动的文件。您可以将ra trace .log复制到外部服务器，也可以直接在屏幕上显示输出。

检查 RA 跟踪文件的名称。

<#root>

9800#

dir bootflash: | inc ra_trace

将文件复制到外部服务器：

<#root>

9800#

copy bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log tftp://

/ra-FILENAME.txt

显示内容：

<#root>

9800#

more bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log

第 6 步： 如果根本原因仍然不明显，请收集内部日志，这是更为详细的调试级别日志。您无需再次调试客户端，因为命令提供已收集并内部存储的调试日志。

<#root>

9800#

show logging profile wireless internal filter [mac | ip] [

|

] to-file ra-internal-

.txt

注意：此命令输出会返回所有进程全部日志记录级别的跟踪信息，因此数量非常庞大。请联系Cisco TAC以帮助分析这些跟踪。

<#root>

9800#

copy bootflash:ra-internal-

.txt tftp://

/ra-internal-

.txt

显示内容：

<#root>

9800#

more bootflash:ra-internal-

```
.txt
```

第 7 步： 删除调试条件。

注意：确保您始终在故障排除会话后删除调试条件。

嵌入式数据包捕获

9800控制器可以在本地嗅探数据包；这样可以更轻松地排除故障，因为控制平面数据包处理可视性。

步骤1.定义一个ACL以过滤感兴趣的流量。对于Web身份验证，建议允许来自Web服务器的流量以及来自和流向客户端连接的几个AP的流量。

```
<#root>
9800(config)#
ip access-list extended EWA-pcap

9800(config-ext-nacl)#
permit ip any host

9800(config-ext-nacl)#
permit ip host

 any

9800(config-ext-nacl)#
permit ip any host
```

```
9800(config-ext-nacl)#
permit ip host


        any
```

步骤2.定义监控器捕获参数。确保双向启用控制平面流量，接口是指控制器的物理上行链路。

```
<#root>
9800#
monitor capture EWA buffer size


9800#
monitor capture EWA access-list EWA-pcap

9800#
monitor capture EWA control-plane both interface


        both
```

```
<#root>
9800#
show monitor capture EWA

Status Information for Capture EWA
Target Type:
Interface: Control Plane, Direction: BOTH
Interface: TenGigabitEthernet0/1/0, Direction: BOTH
```

```
Status : Inactive
Filter Details:

Access-list: EWA-pcap


Inner Filter Details:
Buffer Details:
Buffer Type: LINEAR (default)

Buffer Size (in MB): 100


Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Packet sampling rate: 0 (no sampling)
```

步骤3.开始监视器捕获并重现问题。

```
<#root>

9800#

monitor capture EWA start


Started capture point : EWA
```

步骤4.停止监控器捕获并将其导出。

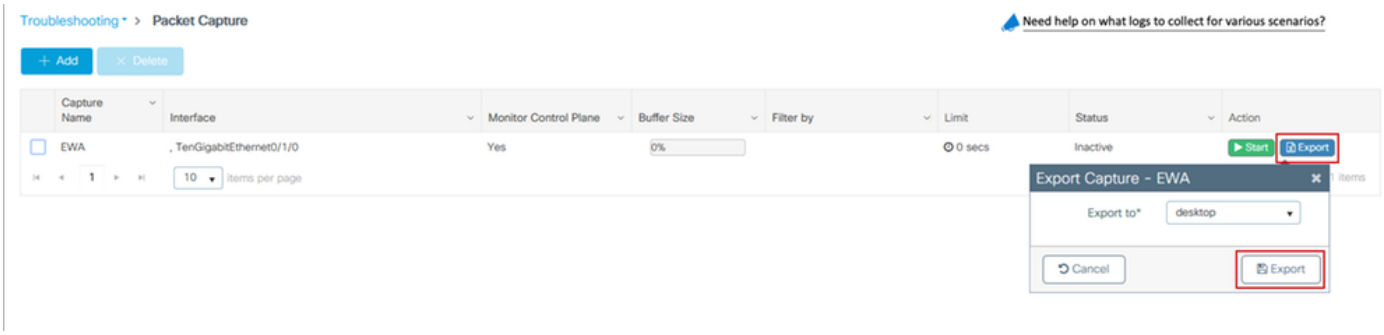
```
<#root>

9800#

monitor capture EWA stop


Stopped capture point : EWA
9800#monitor capture EWA export tftp://<a.b.c.d>/EWA.pcap
```

或者，也可以从GUI下载捕获，导航到故障排除>数据包捕获，并在配置的捕获上选择导出。从下拉菜单中选择desktop（桌面），将捕获通过HTTP下载到所需的文件夹中。



客户端故障排除

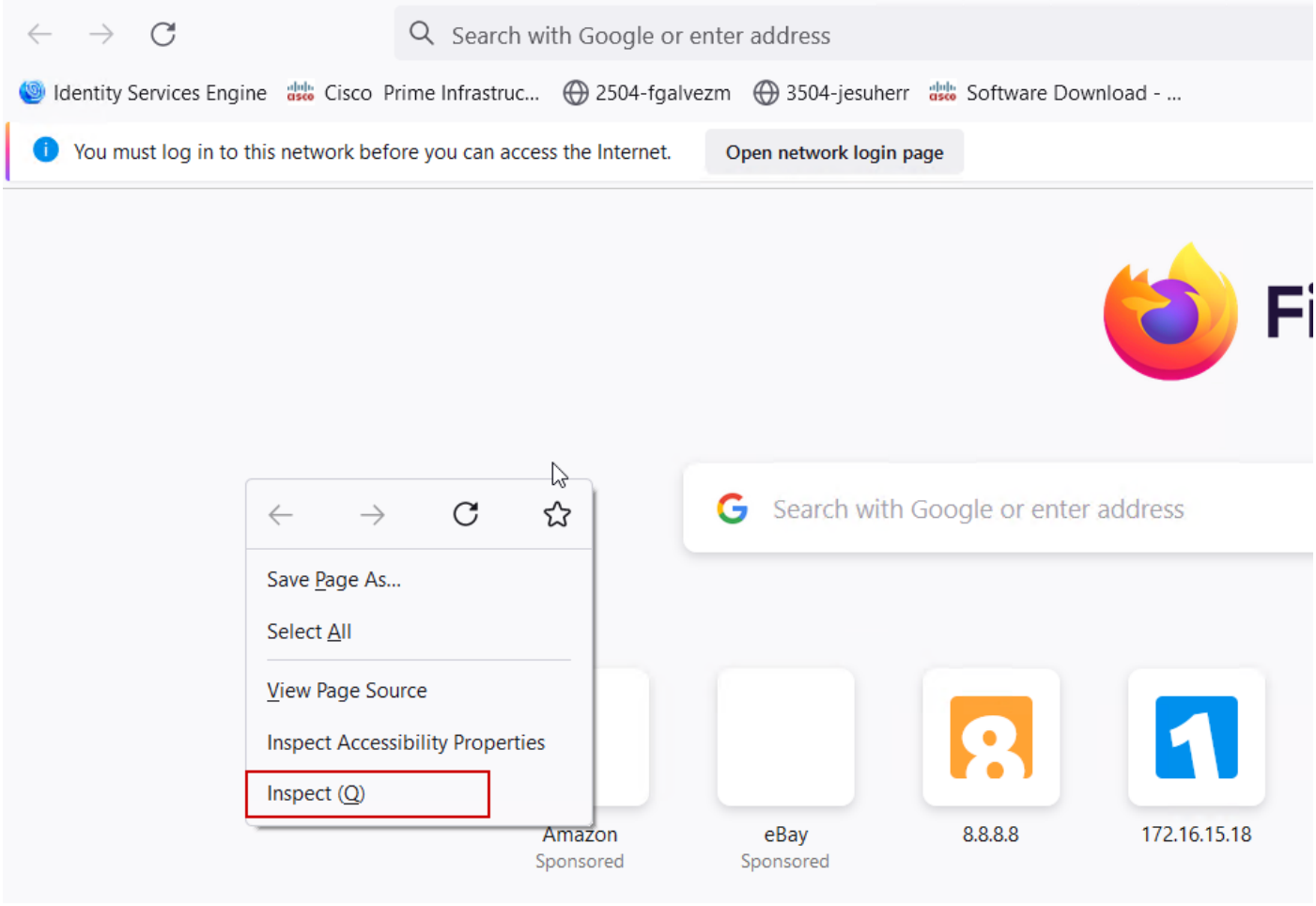
Web身份验证WLAN依赖于客户端行为，在此基础上，客户端行为知识和信息是确定Web身份验证错误行为的根本原因的关键。

HAR浏览器故障排除

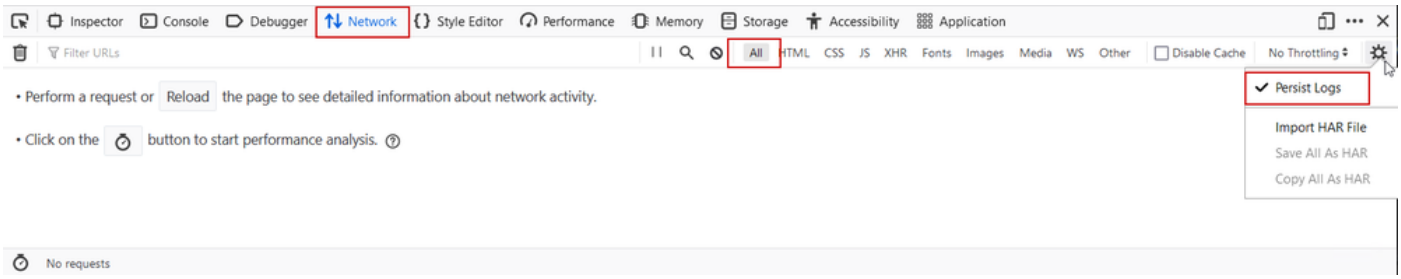
许多现代浏览器（如Mozilla Firefox和Google Chrome）都提供控制台开发人员工具来调试Web应用程序交互。HAR文件是客户端 — 服务器交互的记录，提供HTTP交互的时间表以及请求和响应信息（报头、状态代码、参数等）。

可以从客户端浏览器导出HAR文件，然后在不同的浏览器上导入该文件以进行进一步分析。本文档概述如何从Mozilla Firefox收集HAR文件。

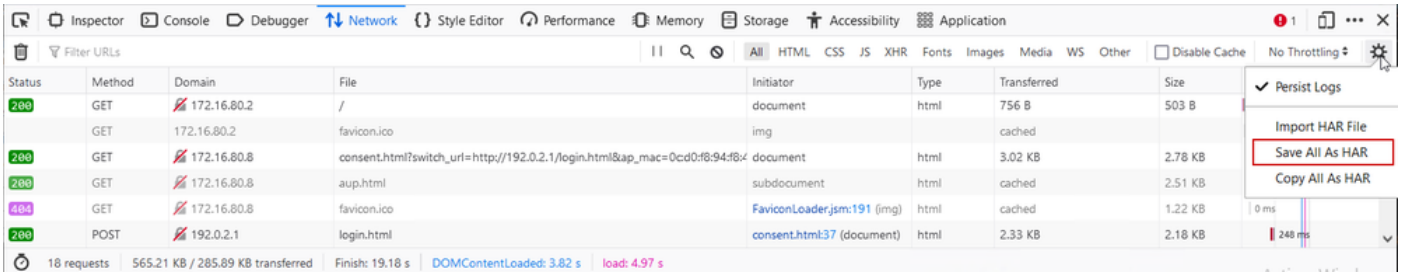
步骤1.使用Ctrl + Shift + I打开Web开发工具，或者右键单击浏览器内容，然后选择Inspect。



第2步：导航到Network，确保选择“All”以捕获所有请求类型。选择齿轮图标并确保Persist Logs旁边有一个箭头，否则只要触发域更改，就会清除日志请求。



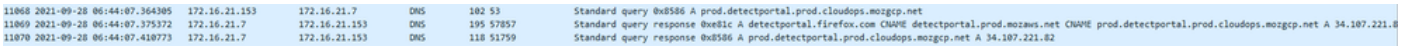
步骤3.重现问题，确保浏览器记录所有请求。一旦问题再次出现，停止网络日志记录，然后选择齿轮图标并选择Save All As HAR。



客户端数据包捕获

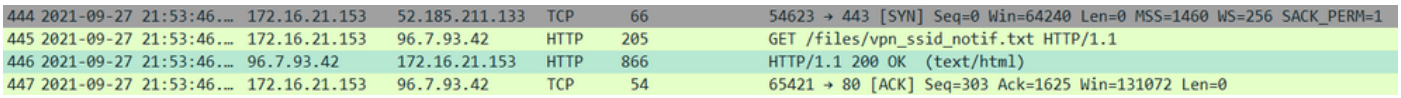
使用Windows或MacOS等操作系统的无线客户端可以嗅探其无线网卡适配器上的数据包。虽然不是空中数据包捕获的直接替代方法，但它们可以大致了解整体的Web身份验证流程。

DNS请求：



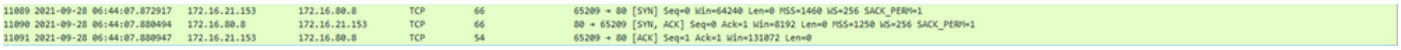
```
11068 2021-09-28 06:44:07.364305   172.16.21.153       172.16.21.7         DNS       102 53          Standard query 0x8586 A prod.detectportal.prod.cloudops.mozgcp.net
11069 2021-09-28 06:44:07.375372   172.16.21.7         172.16.21.153       DNS       195 57857       Standard query response 0xe81c A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.8
11070 2021-09-28 06:44:07.410773   172.16.21.7         172.16.21.153       DNS       118 51759       Standard query response 0x8586 A prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.82
```

用于重定向的初始TCP握手和HTTP GET:



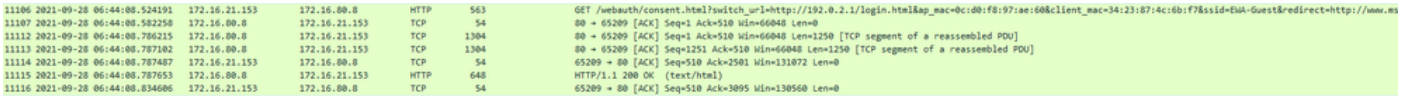
```
444 2021-09-27 21:53:46.… 172.16.21.153   52.185.211.133   TCP      66   54623 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
445 2021-09-27 21:53:46.… 172.16.21.153   96.7.93.42       HTTP     205  GET /files/vpn_ssid_notif.txt HTTP/1.1
446 2021-09-27 21:53:46.… 96.7.93.42      172.16.21.153    HTTP     866  HTTP/1.1 200 OK  (text/html)
447 2021-09-27 21:53:46.… 172.16.21.153   96.7.93.42       TCP      54   65421 → 80 [ACK] Seq=303 Ack=1625 Win=131072 Len=0
```

与外部服务器的TCP握手：



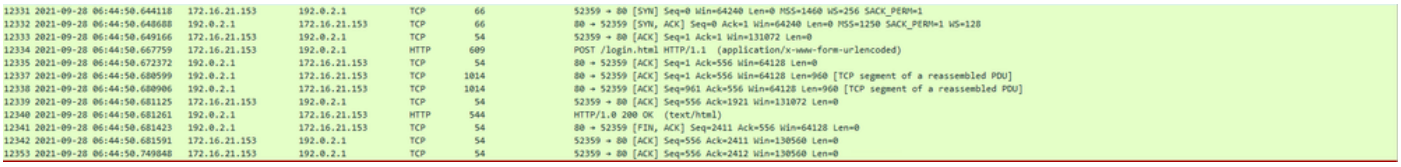
```
11089 2021-09-28 06:44:07.872917   172.16.21.153       172.16.80.8         TCP       66              65209 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
11090 2021-09-28 06:44:07.880494   172.16.80.8         172.16.21.153       TCP       66              80 → 65209 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1250 WS=256 SACK_PERM=1
11091 2021-09-28 06:44:07.880947   172.16.21.153       172.16.80.8         TCP       54              65209 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
```

HTTP GET到外部服务器（强制网络门户请求）：



```
11106 2021-09-28 06:44:08.524191   172.16.21.153       172.16.80.8         HTTP      563             GET /webauth/consent.html?switch_url=http://192.0.2.1/login.html&ap_mac=0c:d0:f8:97:ae:60&client_mac=34:23:87:4c:6b:f7&ssid=EWA-Guest&redirect=http://www.ms
11107 2021-09-28 06:44:08.582258   172.16.80.8         172.16.21.153       TCP       54              80 → 65209 [ACK] Seq=1 Ack=510 Win=66048 Len=0
11112 2021-09-28 06:44:08.786215   172.16.80.8         172.16.21.153       TCP       1304            80 → 65209 [ACK] Seq=1 Ack=510 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
11113 2021-09-28 06:44:08.787102   172.16.80.8         172.16.21.153       TCP       1304            80 → 65209 [ACK] Seq=1251 Ack=510 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
11114 2021-09-28 06:44:08.787487   172.16.21.153       172.16.80.8         TCP       54              65209 → 80 [ACK] Seq=510 Ack=2501 Win=131072 Len=0
11115 2021-09-28 06:44:08.787653   172.16.80.8         172.16.21.153       HTTP      648             HTTP/1.1 200 OK  (text/html)
11116 2021-09-28 06:44:08.834606   172.16.21.153       172.16.80.8         TCP       54              65209 → 80 [ACK] Seq=510 Ack=3095 Win=130560 Len=0
```

HTTP POST到虚拟IP进行身份验证：



```
12331 2021-09-28 06:44:50.644118   172.16.21.153       192.0.2.1           TCP       66              52359 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
12332 2021-09-28 06:44:50.648688   192.0.2.1           172.16.21.153       TCP       66              80 → 52359 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1250 SACK_PERM=1 WS=128
12333 2021-09-28 06:44:50.649166   172.16.21.153       192.0.2.1           TCP       54              52359 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
12334 2021-09-28 06:44:50.667759   172.16.21.153       192.0.2.1           HTTP      609             POST /login.html HTTP/1.1  (application/x-www-form-urlencoded)
12335 2021-09-28 06:44:50.672372   192.0.2.1           172.16.21.153       TCP       54              80 → 52359 [ACK] Seq=1 Ack=556 Win=64128 Len=0
12337 2021-09-28 06:44:50.680599   192.0.2.1           172.16.21.153       TCP       1014            80 → 52359 [ACK] Seq=1 Ack=556 Win=64128 Len=960 [TCP segment of a reassembled PDU]
12338 2021-09-28 06:44:50.680906   192.0.2.1           172.16.21.153       TCP       1014            80 → 52359 [ACK] Seq=961 Ack=556 Win=64128 Len=960 [TCP segment of a reassembled PDU]
12339 2021-09-28 06:44:50.681125   172.16.21.153       192.0.2.1           TCP       54              52359 → 80 [ACK] Seq=556 Ack=1921 Win=131072 Len=0
12340 2021-09-28 06:44:50.681261   192.0.2.1           172.16.21.153       HTTP      544             HTTP/1.0 200 OK  (text/html)
12341 2021-09-28 06:44:50.681423   192.0.2.1           172.16.21.153       TCP       54              80 → 52359 [FIN, ACK] Seq=2411 Ack=556 Win=64128 Len=0
12342 2021-09-28 06:44:50.681591   172.16.21.153       192.0.2.1           TCP       54              52359 → 80 [ACK] Seq=556 Ack=2411 Win=130560 Len=0
12353 2021-09-28 06:44:50.749048   172.16.21.153       192.0.2.1           TCP       54              52359 → 80 [ACK] Seq=556 Ack=2412 Win=130560 Len=0
```

成功尝试的示例

这是从无线活动跟踪角度进行的成功连接尝试的输出，使用此参考来确定连接到第3层Web身份验证SSID的客户端的客户端会话阶段。

802.11身份验证和关联：

<#root>

```
2021/09/28 12:59:51.781967 {wncd_x_R0-0}{1}: [client-orch-sm] [26328]: (note): MAC: 3423.874c.6bf7 Asso
2021/09/28 12:59:51.782009 {wncd_x_R0-0}{1}: [client-orch-sm] [26328]: (debug): MAC: 3423.874c.6bf7
```

Received Dot11 association request.

```
 Processing started,
```

SSID: EWA-Guest, Policy profile: Guest-Policy

```
, AP Name: C9117AXI-lobby, Ap Mac Address: 0cd0.f897.ae60 BSSID MAC0000.0000.0000 wlan ID: 4RSSI: -39,
2021/09/28 12:59:51.782152 {wncd_x_R0-0}{1}: [client-orch-state] [26328]: (note): MAC: 3423.874c.6bf7 C
2021/09/28 12:59:51.782357 {wncd_x_R0-0}{1}: [dot11-validate] [26328]: (info): MAC: 3423.874c.6bf7 WiFi
2021/09/28 12:59:51.782480 {wncd_x_R0-0}{1}: [dot11] [26328]: (debug): MAC: 3423.874c.6bf7 dot11 send a
```

Sending association response with resp_status_code: 0

```
2021/09/28 12:59:51.782483 {wncd_x_R0-0}{1}: [dot11] [26328]: (debug): MAC: 3423.874c.6bf7 Dot11 Capabi
2021/09/28 12:59:51.782509 {wncd_x_R0-0}{1}: [dot11-frame] [26328]: (info): MAC: 3423.874c.6bf7 WiFi di
2021/09/28 12:59:51.782519 {wncd_x_R0-0}{1}: [dot11] [26328]: (info): MAC: 3423.874c.6bf7 dot11 send as
2021/09/28 12:59:51.782611 {wncd_x_R0-0}{1}: [dot11] [26328]: (note): MAC: 3423.874c.6bf7
```

Association success. AID 1

```
, Roaming = False, WGB = False, 11r = False, 11w = False
2021/09/28 12:59:51.782626 {wncd_x_R0-0}{1}: [dot11] [26328]: (info): MAC: 3423.874c.6bf7 DOT11 state t
2021/09/28 12:59:51.782676 {wncd_x_R0-0}{1}: [client-orch-sm] [26328]: (debug): MAC: 3423.874c.6bf7
```

Station Dot11 association is successful.

已跳过第2层身份验证：

<#root>

```
2021/09/28 12:59:51.782727 {wncd_x_R0-0}{1}: [client-orch-sm] [26328]: (debug): MAC: 3423.874c.6bf7 Sta
2021/09/28 12:59:51.782745 {wncd_x_R0-0}{1}: [client-orch-state] [26328]: (note): MAC: 3423.874c.6bf7 C
2021/09/28 12:59:51.782785 {wncd_x_R0-0}{1}: [client-auth] [26328]: (note): MAC: 3423.874c.6bf7
```

L2 Authentication initiated. method WEBAUTH

```
, Policy VLAN 2621,AAA override = 0
2021/09/28 12:59:51.782803 {wncd_x_R0-0}{1}: [sanet-shim-translate] [26328]: (ERR): 3423.874c.6bf7 wlan
[...]
2021/09/28 12:59:51.787912 {wncd_x_R0-0}{1}: [client-auth] [26328]: (info): MAC: 3423.874c.6bf7 Client
2021/09/28 12:59:51.787953 {wncd_x_R0-0}{1}: [client-auth] [26328]: (info): MAC: 3423.874c.6bf7 Client
2021/09/28 12:59:51.787966 {wncd_x_R0-0}{1}: [client-orch-sm] [26328]: (debug): MAC: 3423.874c.6bf7
```

L2 Authentication of station is successful., L3 Authentication : 1

ACL plumb:

<#root>

```
2021/09/28 12:59:51.785227 {wncd_x_R0-0}{1}: [webauth-sm] [26328]: (info): [ 0.0.0.0]Starting Webauth,
2021/09/28 12:59:51.785307 {wncd_x_R0-0}{1}: [auth-mgr-feat_wireless] [26328]: (info): [0000.0000.0000:
2021/09/28 12:59:51.785378 {wncd_x_R0-0}{1}: [webauth-acl] [26328]: (info): capwap_9000000b[3423.874c.6
```

Applying IPv4 intercept ACL via SVM, name: WA-v4-int-172.16.80.8

```
, priority: 50, IIF-ID: 0
2021/09/28 12:59:51.785738 {wncd_x_R0-0}{1}: [epm-redirect] [26328]: (info): [0000.0000.0000:unknown]
```

URL-Redirect-ACL = WA-v4-int-172.16.80.8

```
2021/09/28 12:59:51.786324 {wncd_x_R0-0}{1}: [webauth-acl] [26328]: (info): capwap_9000000b[3423.874c.6
```

Applying IPv6 intercept ACL via SVM, name: IP-Adm-V6-Int-ACL-global, priority: 52

```
, IIF-ID: 0
2021/09/28 12:59:51.786598 {wncd_x_R0-0}{1}: [epm-redirect] [26328]: (info): [0000.0000.0000:unknown]
```

URL-Redirect-ACL = IP-Adm-V6-Int-ACL-global

```
2021/09/28 12:59:51.787904 {wncd_x_R0-0}{1}: [client-auth] [26328]: (info): MAC: 3423.874c.6bf7 Client
```

IP学习过程：

```
<#root>

2021/09/28 12:59:51.799515 {wncd_x_R0-0}{1}: [client-orch-state] [26328]: (note): MAC: 3423.874c.6bf7 C
2021/09/28 12:59:51.799716 {wncd_x_R0-0}{1}: [client-iplearn] [26328]: (info): MAC: 3423.874c.6bf7

IP-learn state transition: S_IPLEARN_INIT -> S_IPLEARN_IN_PROGRESS


2021/09/28 12:59:51.802213 {wncd_x_R0-0}{1}: [client-auth] [26328]: (info): MAC: 3423.874c.6bf7 Client
2021/09/28 12:59:51.916777 {wncd_x_R0-0}{1}: [sisf-packet] [26328]: (debug): RX: ARP from interface capw
[...]
2021/09/28 12:59:52.810136 {wncd_x_R0-0}{1}: [client-iplearn] [26328]: (note): MAC: 3423.874c.6bf7

Client IP learn successful. Method: ARP IP: 172.16.21.153


2021/09/28 12:59:52.810185 {wncd_x_R0-0}{1}: [epm] [26328]: (info): [0000.0000.0000:unknown] HDL = 0x0
2021/09/28 12:59:52.810404 {wncd_x_R0-0}{1}: [auth-mgr] [26328]: (info): [3423.874c.6bf7:capwap_9000000
2021/09/28 12:59:52.810794 {wncd_x_R0-0}{1}: [auth-mgr-feat_wireless] [26328]: (info): [0000.0000.0000:
2021/09/28 12:59:52.810863 {wncd_x_R0-0}{1}: [client-iplearn] [26328]: (info): MAC: 3423.874c.6bf7

IP-learn state transition: S_IPLEARN_IN_PROGRESS -> S_IPLEARN_COMPLETE
```

第3层身份验证和重定向过程：

```
<#root>

2021/09/28 12:59:52.811141 {wncd_x_R0-0}{1}: [client-auth] [26328]: (note): MAC: 3423.874c.6bf7

L3 Authentication initiated. LWA


2021/09/28 12:59:52.811154 {wncd_x_R0-0}{1}: [client-auth] [26328]: (info): MAC: 3423.874c.6bf7 Client
2021/09/28 12:59:55.324550 {wncd_x_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap_9000000b[3423.874c
2021/09/28 12:59:55.324565 {wncd_x_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap_9000000b[3423.874c

HTTP GET request


2021/09/28 12:59:55.324588 {wncd_x_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap_9000000b[3423.874c
[...]
2021/09/28 13:01:29.859434 {wncd_x_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap_9000000b[3423.874c

POST rcvd when in LOGIN state


2021/09/28 13:01:29.859636 {wncd_x_R0-0}{1}: [webauth-acl] [26328]: (info): capwap_9000000b[3423.874c.6
2021/09/28 13:01:29.860335 {wncd_x_R0-0}{1}: [webauth-acl] [26328]: (info): capwap_9000000b[3423.874c.6
2021/09/28 13:01:29.861092 {wncd_x_R0-0}{1}: [auth-mgr] [26328]: (info): [3423.874c.6bf7:capwap_9000000

Authc success from WebAuth, Auth event success
```

2021/09/28 13:01:29.861151 {wncd_x_R0-0}{1}: [ewlc-infra-evq] [26328]: (note): Authentication Success.
2021/09/28 13:01:29.862867 {wncd_x_R0-0}{1}: [client-auth] [26328]: (note): MAC: 3423.874c.6bf7

L3 Authentication Successful.

ACL:[]
2021/09/28 13:01:29.862871 {wncd_x_R0-0}{1}: [client-auth] [26328]: (info): MAC: 3423.874c.6bf7

Client auth-interface state transition: S_AUTHIF_WEBAUTH_PENDING -> S_AUTHIF_WEBAUTH_DONE

转换到运行状态：

<#root>

2021/09/28 13:01:29.863176 {wncd_x_R0-0}{1}: [client-auth] [26328]: (note): MAC: 3423.874c.6bf7 ADD MOB
2021/09/28 13:01:29.863272 {wncd_x_R0-0}{1}: [errmsg] [26328]: (info): %CLIENT_ORCH_LOG-6-CLIENT_ADDED_

Username entry (3423.874C.6BF7) joined with ssid (EWA-Guest) for device with MAC: 3423.874c.6bf7

2021/09/28 13:01:29.863334 {wncd_x_R0-0}{1}: [aaa-attr-inf] [26328]: (info): [Applied attribute :bsn-v
2021/09/28 13:01:29.863336 {wncd_x_R0-0}{1}: [aaa-attr-inf] [26328]: (info): [Applied attribute : time
2021/09/28 13:01:29.863343 {wncd_x_R0-0}{1}: [aaa-attr-inf] [26328]: (info): [Applied attribute : url-
2021/09/28 13:01:29.863387 {wncd_x_R0-0}{1}: [ewlc-qos-client] [26328]: (info): MAC: 3423.874c.6bf7 Cli
2021/09/28 13:01:29.863409 {wncd_x_R0-0}{1}: [rog-proxy-capwap] [26328]: (debug):

Managed client RUN state notification

: 3423.874c.6bf7
2021/09/28 13:01:29.863451 {wncd_x_R0-0}{1}: [client-orch-state] [26328]: (note): MAC: 3423.874c.6bf7

Client state transition: S_CO_L3_AUTH_IN_PROGRESS -> S_CO_RUN

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）。