9800 WLCでの外部Web認証の設定およびトラブルシューティング

内容

はじめに

このドキュメントでは、Catalyst 9800ワイヤレスLANコントローラ(WLC)での外部Web認証(EWA)の設定およびトラブルシューティング方法について説明します。

前提条件

このドキュメントでは、Webサーバが外部通信を許可するように適切に設定され、WLCがユーザを認証し、クライアントセッションをRUN状態に移行するために必要なすべてのパラメータを送信するようにWebページが適切に設定されていることを前提としています。

注：外部リソースへのアクセスは、アクセスリストの権限によってWLCによって制限されるため、Webページで使用されるすべてのスクリプト、フォント、イメージなどをダウンロードして、Webサーバのローカルのままにしておく必要があります。

ユーザ認証に必要なパラメータは次のとおりです。

- buttonClicked:WLCが認証の試行としてアクションを検出できるようにするには、このパラメータの値を「4」に設定する必要があります。
- redirectUrl：このパラメータの値は、認証が成功したときにクライアントを特定のWebサイトに誘導するためにコントローラによって使用されます。
- err_flag：このパラメータは、不完全な情報や誤ったクレデンシャルなどのエラーを示すために使用されます。認証が成功すると「0」に設定されます。
- username：このパラメータはwebauthパラメータマップでのみ使用されます。パラメータマップがconsentに設定されている場合は無視できます。ワイヤレスクライアントのユーザ名を入力する必要がある
- password：このパラメータはwebauthパラメータマップでのみ使用されます。パラメータマップがconsentに設定されている場合は無視できます。ワイヤレスクライアントパスワードを入力する必要があります。

要件

次の項目に関する知識があることが推奨されます。

- Hyper Text Markup Language(HTML)Web開発
- Cisco IOS®-XEワイヤレス機能
- Webブラウザ開発者ツール

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

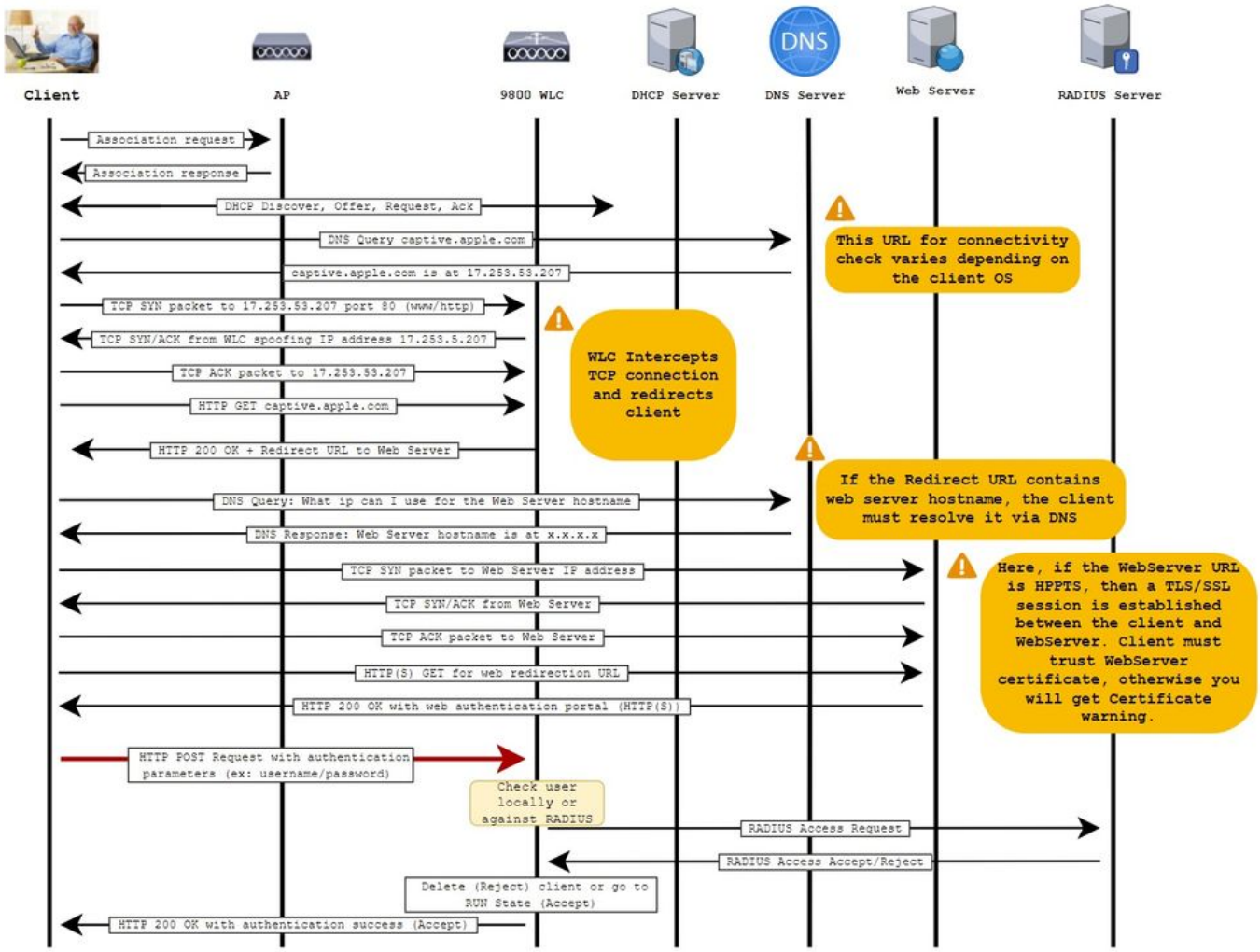
- C9800-CL WLC Cisco IOS®-XEバージョン17.3.3
- インターネットインフォメーションサービス(IIS)機能を備えたMicrosoft Windows Server 2012
- 2802および9117アクセスポイント

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

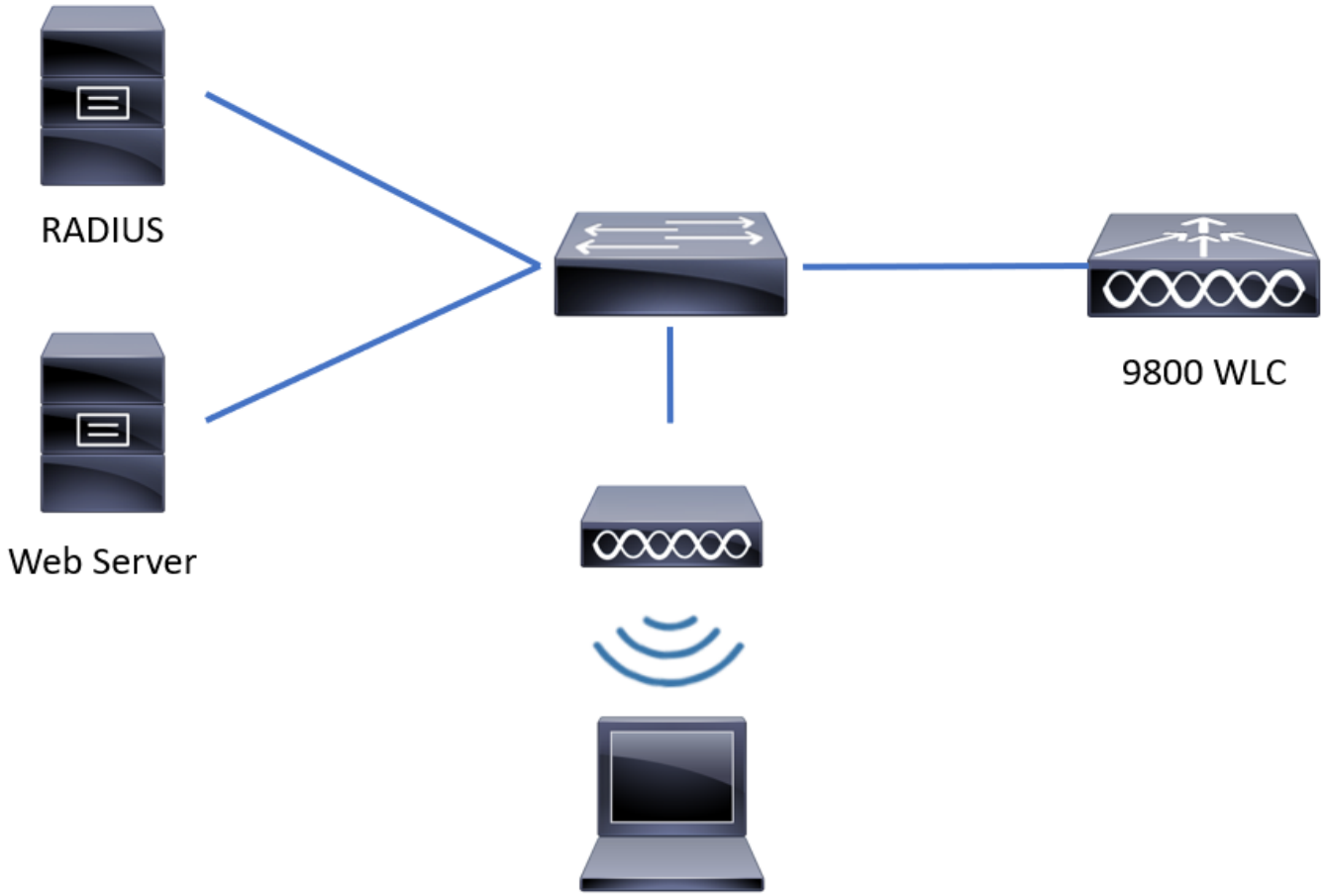
外部Web認証では、専用Webサーバ上のWLCの外部でホストされるWebポータル、またはIdentity Services Engine(ISE)などの多目的サーバを利用して、Webコンポーネントのきめ細かなアクセスと管理を可能にします。クライアントを外部Web認証WLANに正常にオンボードするために必要なハンドシェイクは、イメージでレンダリングされます。この図は、ワイヤレスクライアント、WLC、ドメインネームシステム(DNS)サーバ間のUniform Resource Location(URL)を解決する一連のインタラクション、およびWLCがユーザクレデンシャルをローカルで検証するWebサーバを示しています。このワークフローは、障害状態のトラブルシューティングに役立ちます。

注：クライアントからWLCへのHTTP POSTコールの前に、パラメータマップでセキュアWeb認証が有効になっており、信頼できる認証局によって署名されたトラストポイントがWLCにない場合、セキュリティアラートがブラウザに表示されます。コントローラがクライアントセッションをRUN状態にするためには、クライアントがこの警告をバイパスし、フォームの再送信を受け入れる必要があります。



設定

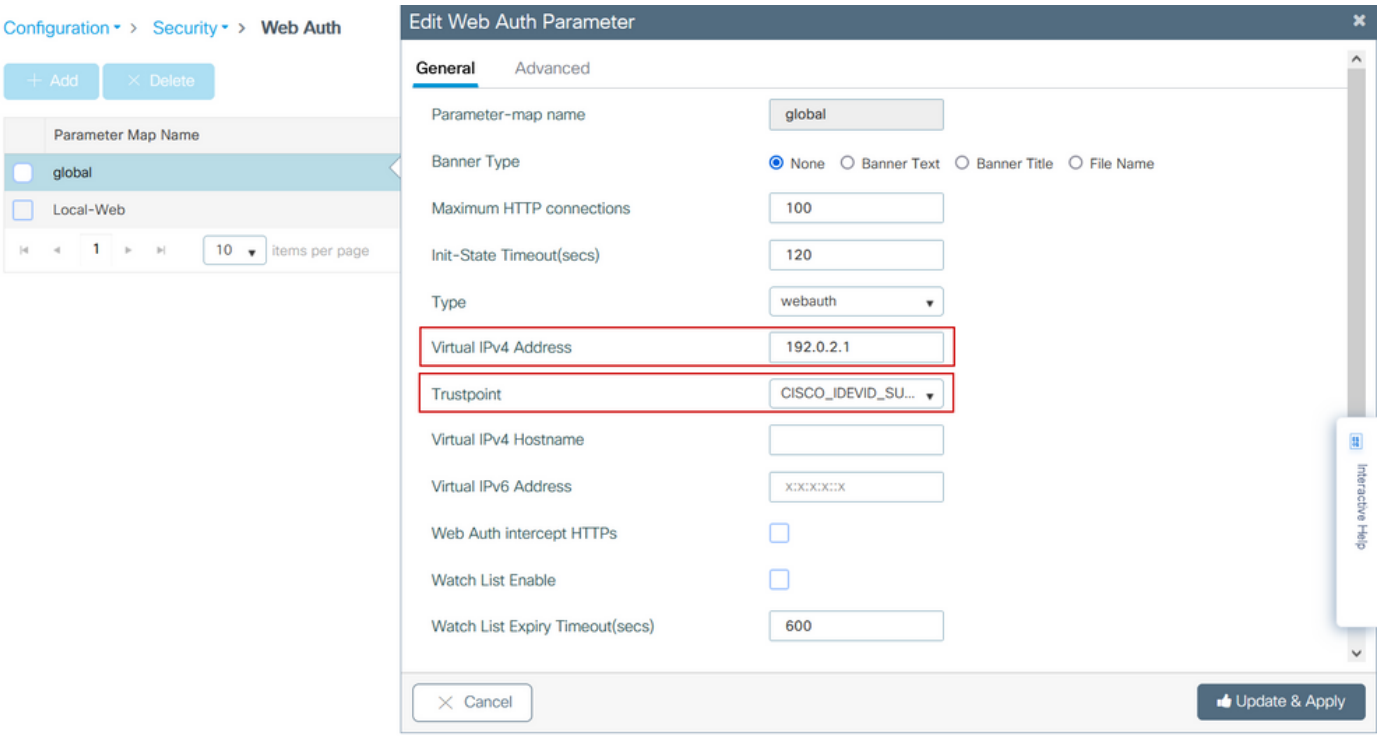
ネットワーク図



Webパラメータの設定

ステップ 1：Configuration > Security > Web Auth の順に移動し、グローバルパラメータマップを選択します。適切なリダイレクション機能を提供するために仮想IPv4アドレスとトラストポイントが設定されていることを確認します。

注：デフォルトでは、リダイレクション処理を開始するためにブラウザはHTTP Webサイトを使用します。HTTPSリダイレクションが必要な場合は、「Web Auth intercept HTTPs」をチェックする必要があります。ただし、CPU使用率が増加するため、この設定は推奨されません。

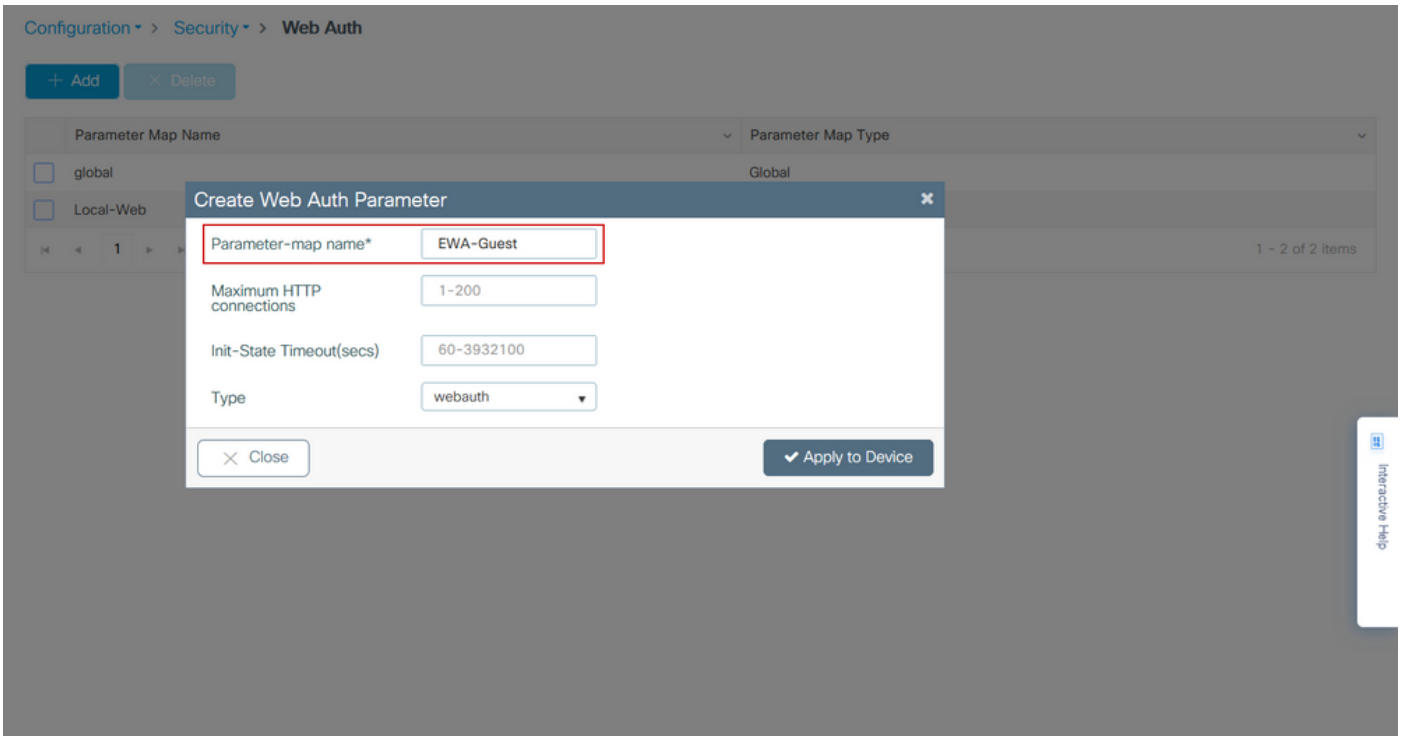


CLI による設定：

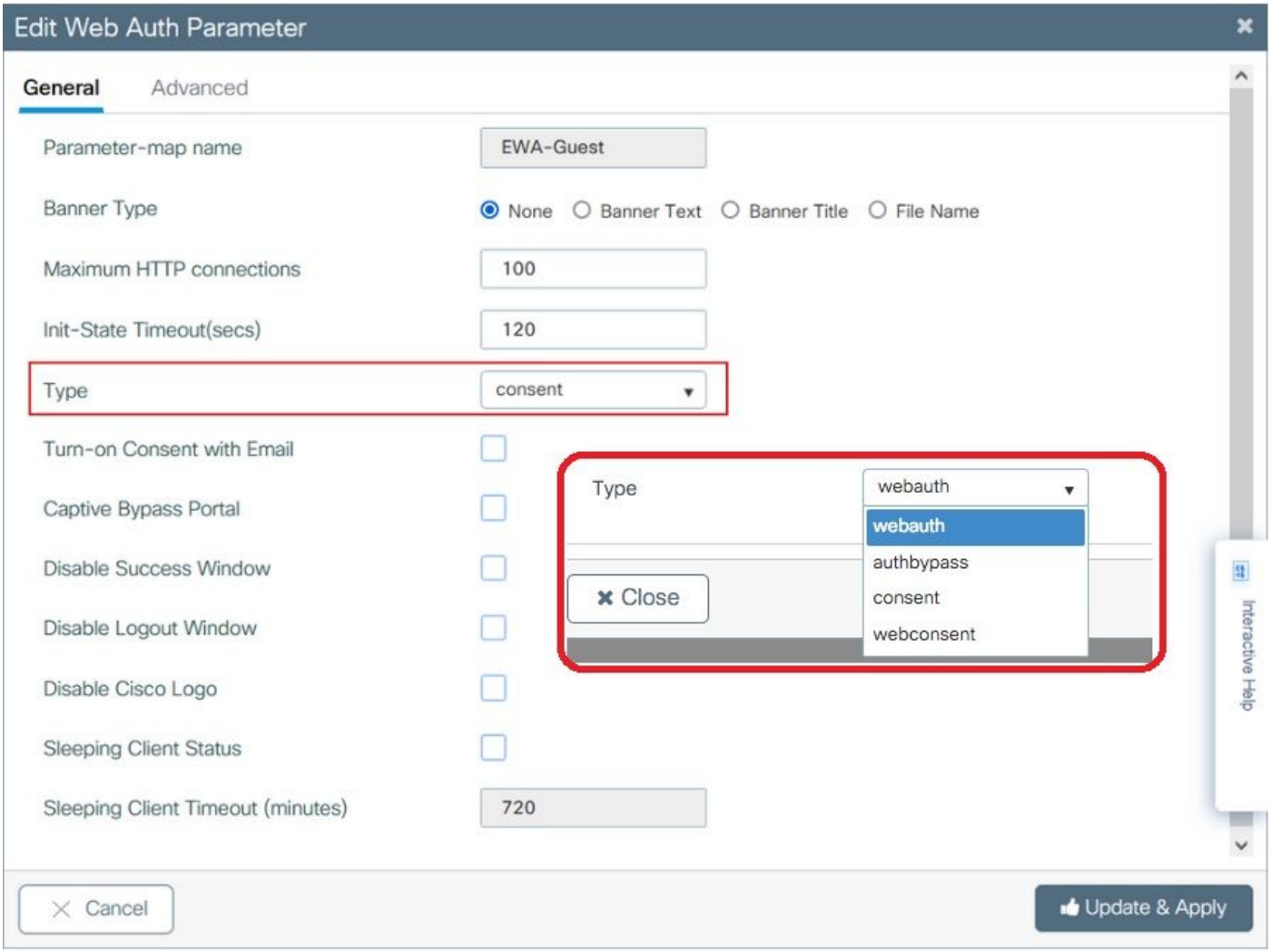
<#root>

```
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
```

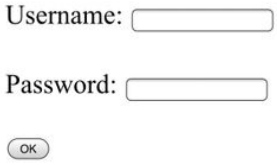
ステップ 2：+ Addを選択し、外部サーバを指す新しいパラメータマップの名前を設定します。オプションで、クライアントが除外されるまでのHTTP認証エラーの最大数と、クライアントがWeb認証状態を維持できる時間（秒単位）を設定します。



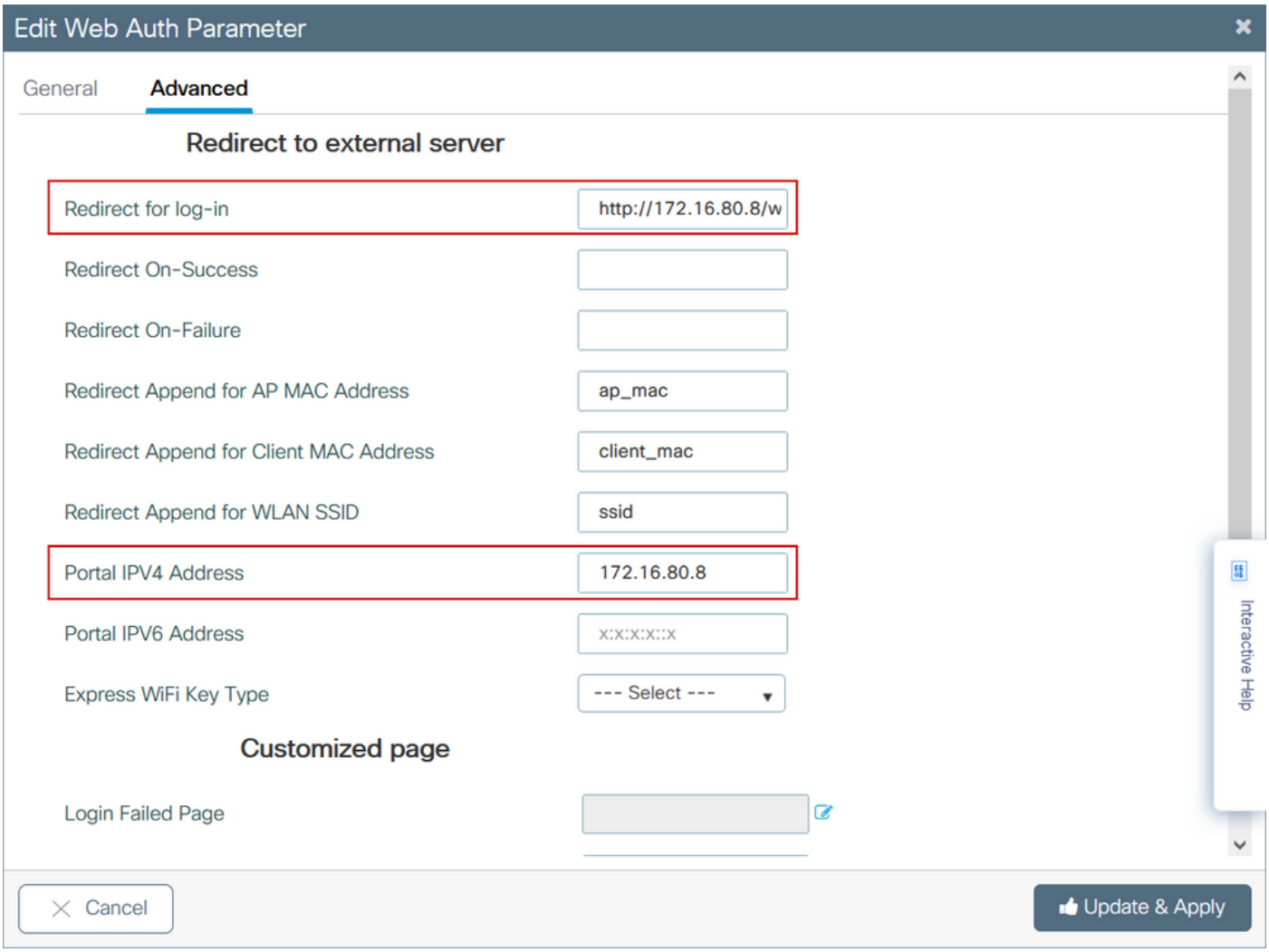
ステップ 3：Generalタブで、新しく作成したパラメータマップを選択し、Typeドロップダウンリストから認証タイプを設定します。



- Parameter-map name = WebAuthパラメータマップに割り当てられた名前
- 最大HTTP接続数=クライアントが除外されるまでの認証エラーの数
- Init-State Timeout (seconds) =クライアントがWeb認証の状態を維持できる秒数
- Type = Web認証のタイプ

webauth	認証バイパス	同意	Web承諾
	クライアントが SSIDを使用してIPアドレスを取得し、9800 WLCを使用して MACアドレスが を入力すると、 ネットワーク（存在する場合）が移動されます RUNステートに設定されていない場合、 参加できません。 （Web認証にはフォールバックしません）。		

ステップ 4： Advancedタブで、ログイン用とポータル用のIPアドレス用のリダイレクトを、それぞれ特定のサーバサイトのURLとIPアドレスを使用して設定します。



手順2、3、および4のCLI設定：

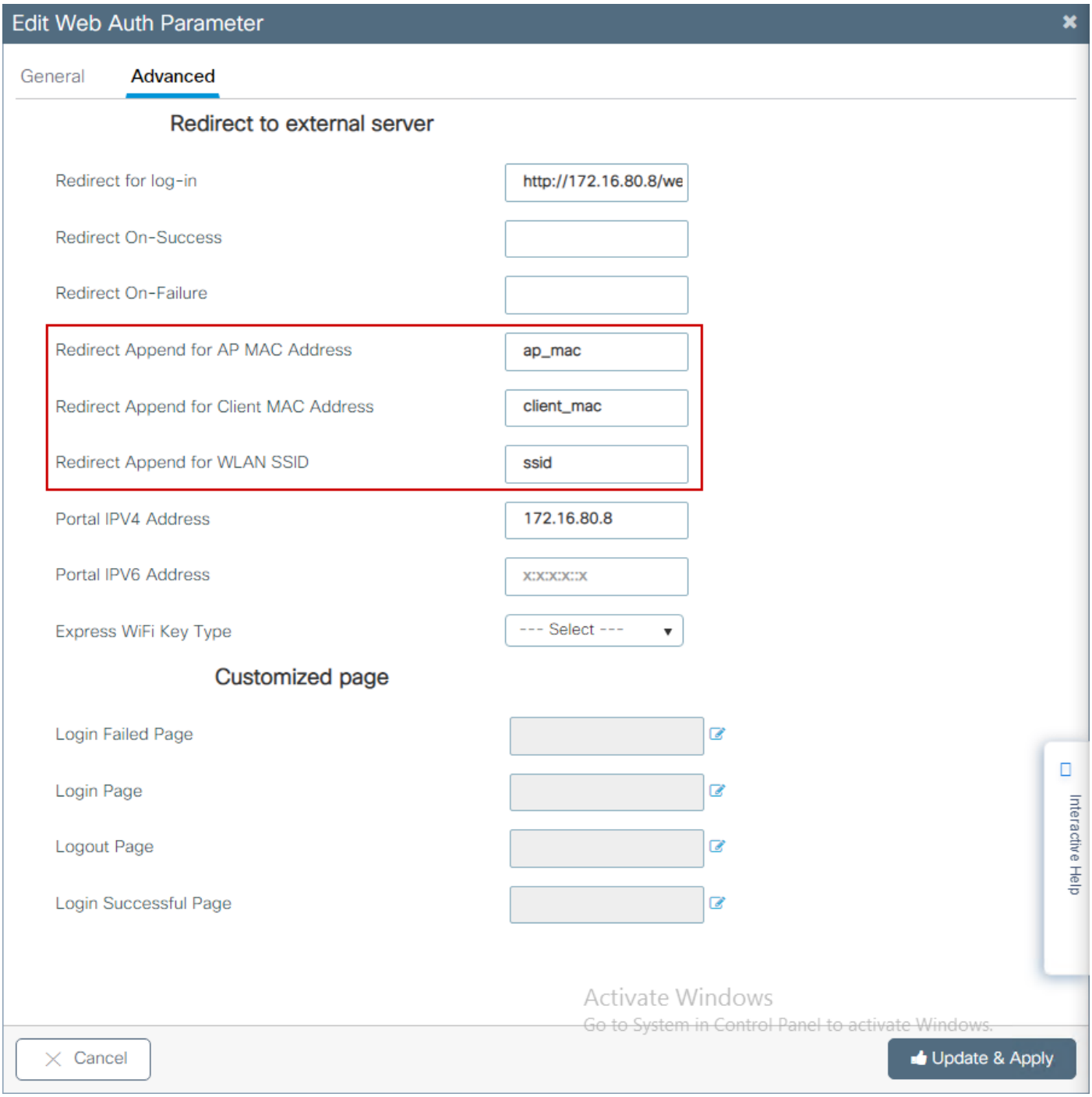
<#root>

```
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
```

ステップ5:（オプション）WLCは、クエリ文字列を使用して追加のパラメータを送信できます。これは、9800をサードパーティ製の外部ポータルと互換性を持たせるために必要な場合がよくあります。「Redirect Append for AP MAC Address」、「Redirect Append for Client MAC Address」、および「Redirect Append for WLAN SSID」フィールドを使用すると、カスタム名を

使用して追加パラメータをリダイレクトACLに追加できます。 新しく作成したパラメータマップを選択し、Advancedタブに移動して、必要なパラメータの名前を設定します。使用可能なパラメータは次のとおりです。

- APのMACアドレス（aa:bb:cc:dd:ee:ff形式）
- クライアントMACアドレス（aa:bb:cc:dd:ee:ff形式）
- SSID名



CLI による設定：

```
<#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
```

この例では、クライアントに送信されるリダイレクションURLは次のようになります。

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

注：Portal IPV4 Address情報を追加すると、ワイヤレスクライアントから外部Web認証サーバへのHTTPおよびHTTPSトラフィックを許可するACLが自動的に追加されるため、追加の事前認証ACLを設定する必要はありません。複数のIPアドレスまたはURLを許可する場合、認証を行う前に特定のURLに一致するIPがフィルタを許可するように設定するしかありません。URLフィルタを使用しない限り、複数のポータルIPアドレスを静的に追加することはできません。

注：グローバルパラメータマップは、仮想IPv4およびIPv6アドレス、Webauthインターセプト HTTP、キャプティブバイパスポータル、ウォッチリストの有効化およびウォッチリストの有効期限タイムアウト設定を定義できる唯一のマップです。

CLI設定の要約：

ローカルWebサーバ

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

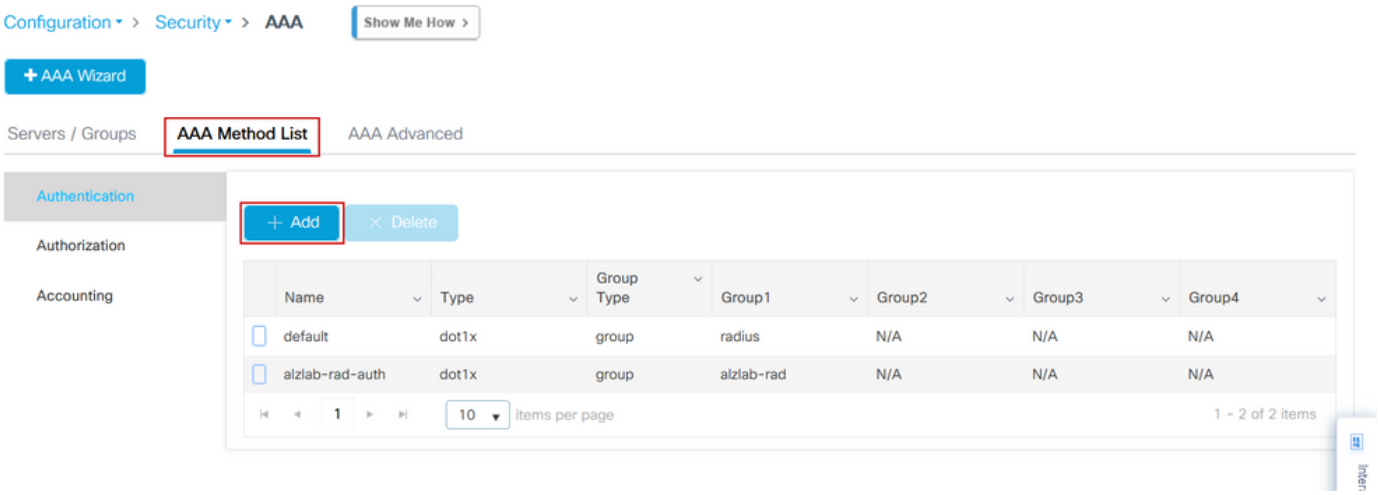
外部Webサーバ

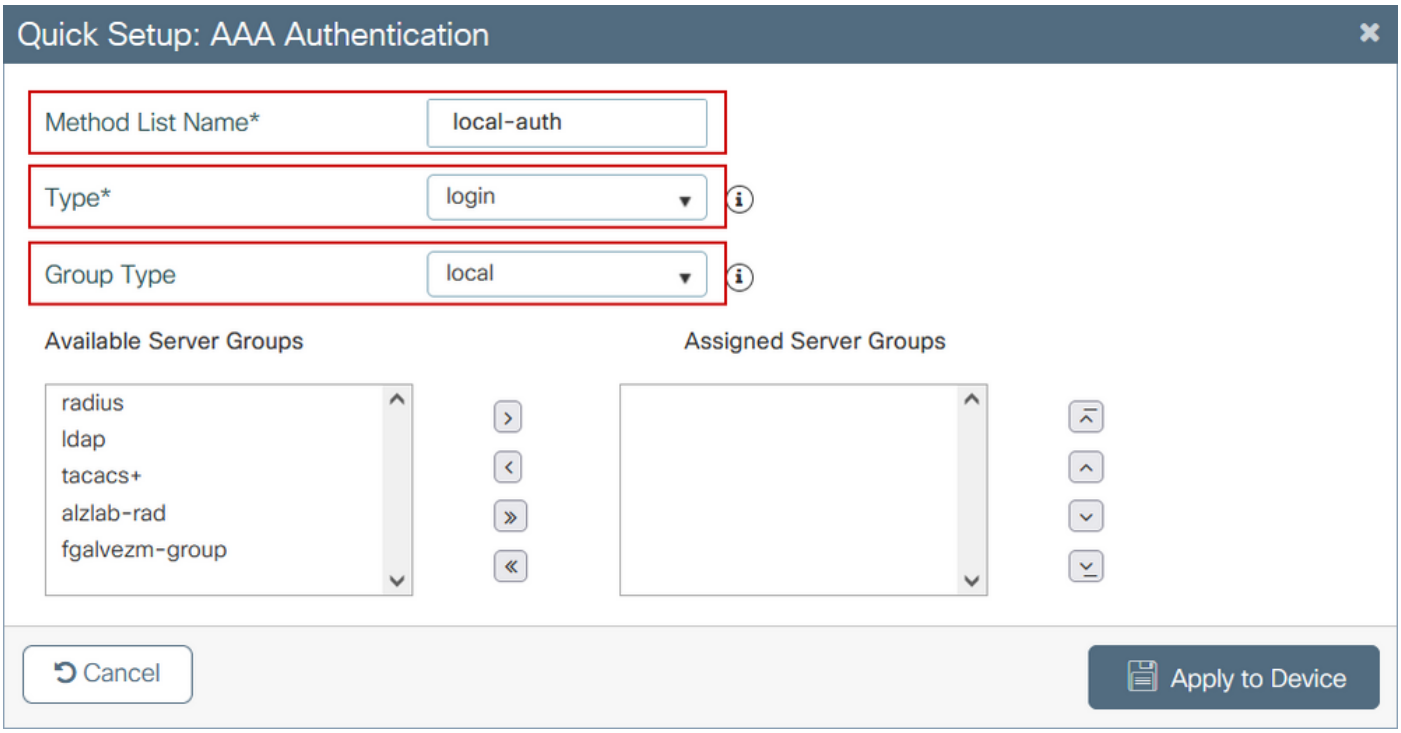
```
parameter-map type webauth <web-parameter-map-name>
 type webauth
 timeout init-state sec 300
 redirect for-login <URL-for-webauth>
 redirect portal ipv4 <external-server's-IP
 max-http-conns 10
```

AAAの設定

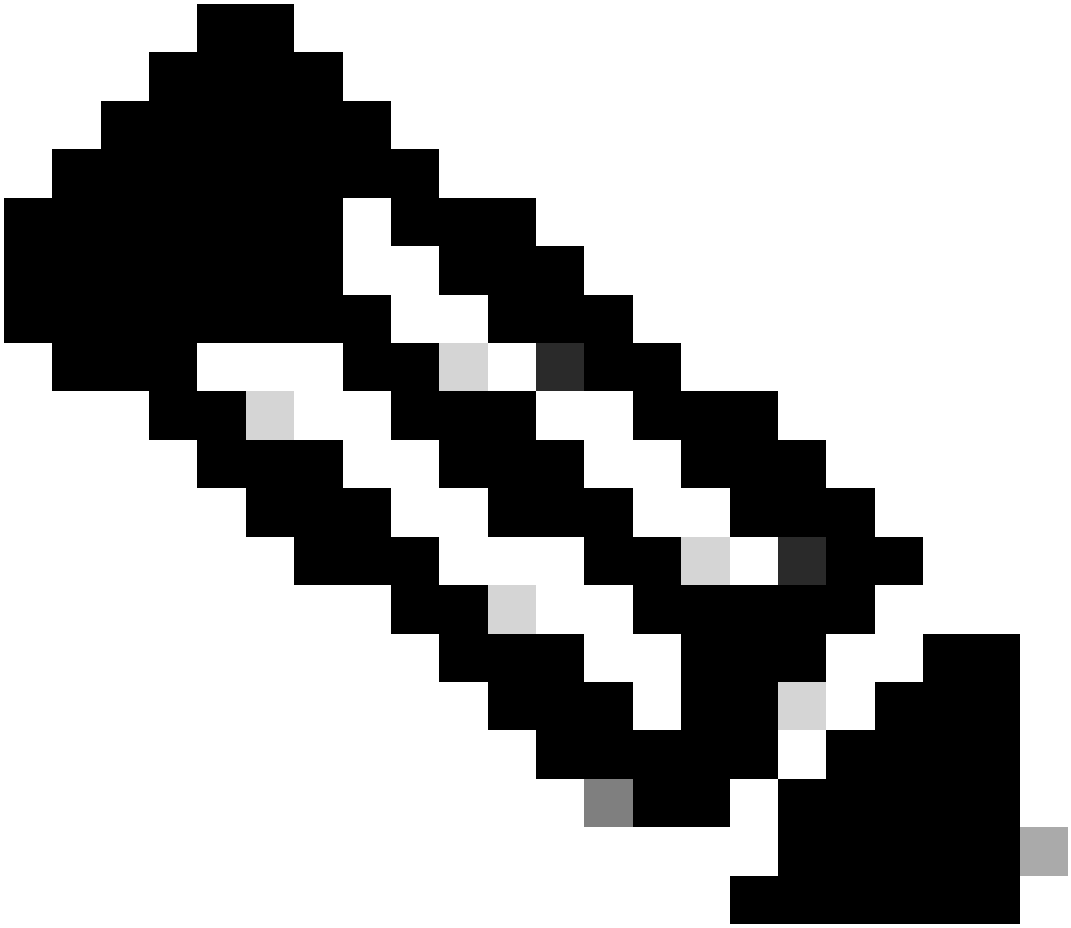
この設定セクションは、WebAuthまたはWebConsentのいずれかの認証タイプ用に設定されたパラメータマップに対してのみ必要です。

ステップ 1：Configuration > Security > AAAの順に移動し、AAA Method Listを選択します。新しい方式リストを設定し、+追加を選択してリストの詳細を入力します。次の図に示すように、タイプが「login」に設定されていることを確認してください。

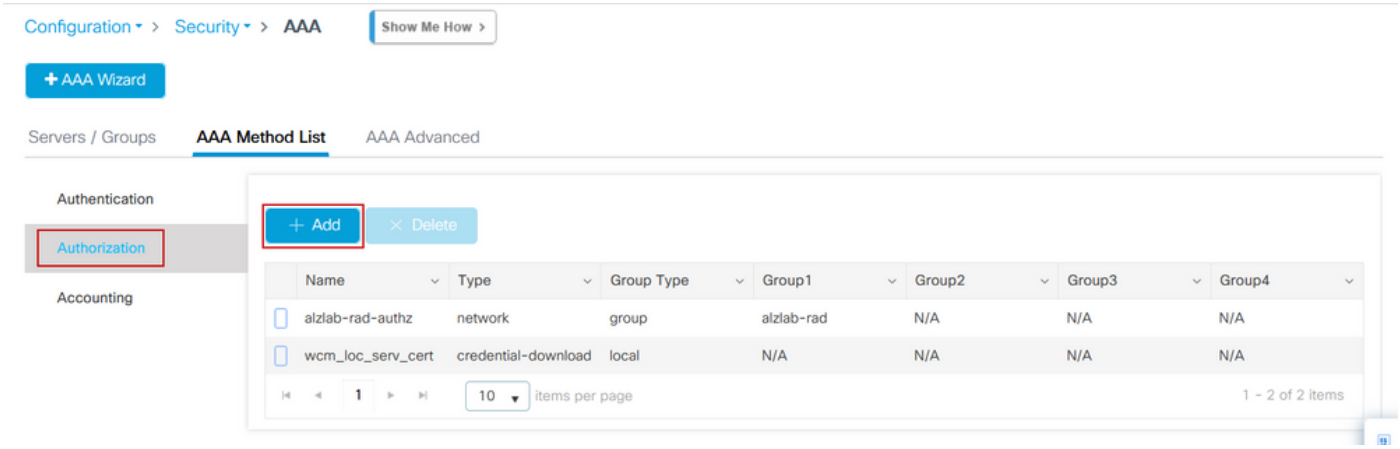


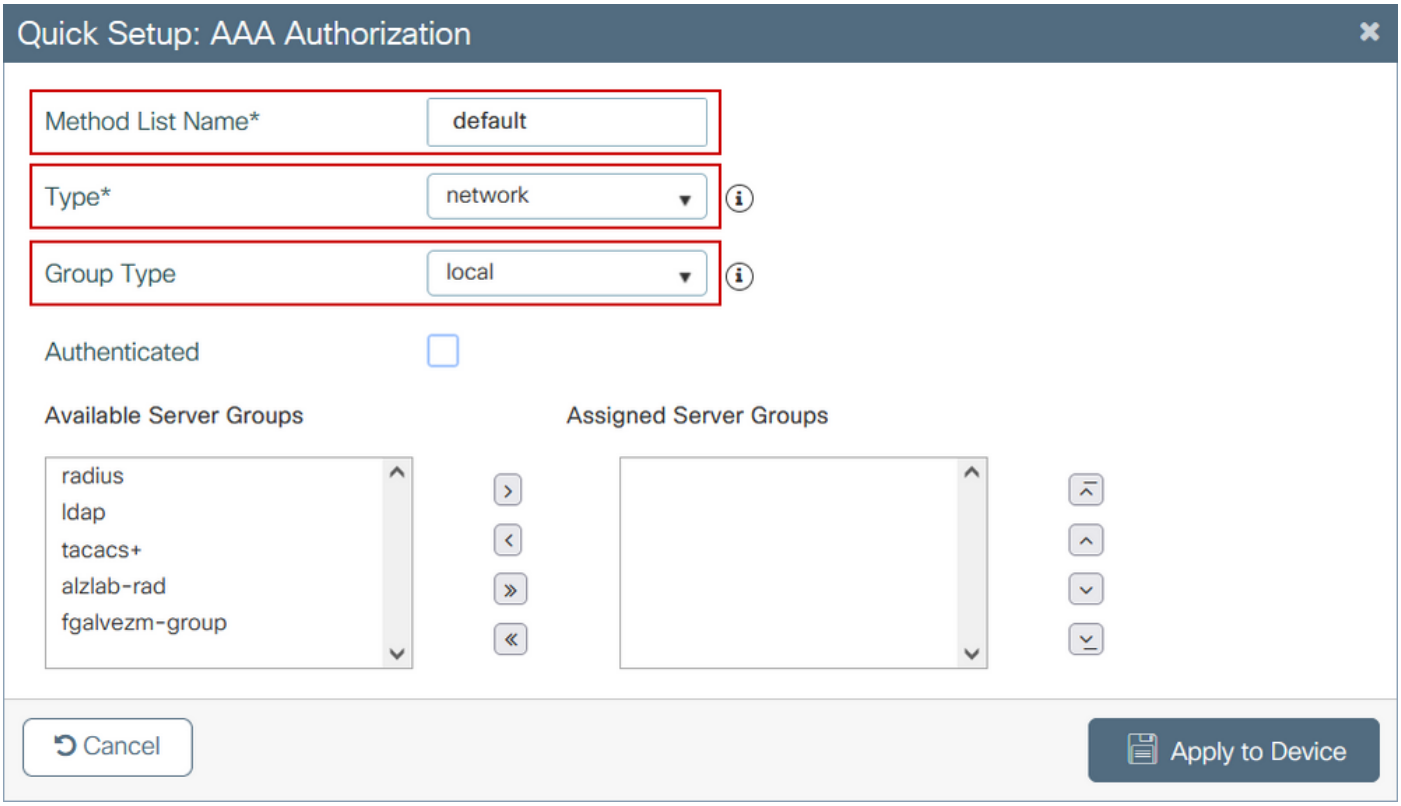


ステップ 2：Authorization を選択し、+ Addを選択して新しい方式リストを作成します。図に示すように、デフォルトの名前をType as networkに設定します。



注：コントローラはWLANレイヤ3セキュリティ設定:中にアドバタイズするため、ローカルログイン方式リストが機能するには、設定「aaa authorization network default local」がデバイスに存在することを確認してください。つまり、ローカルWeb認証を適切に設定するには、defaultという名前の許可方式リストを定義する必要があります。このセクションでは、この特定の認可方式リストを設定します。





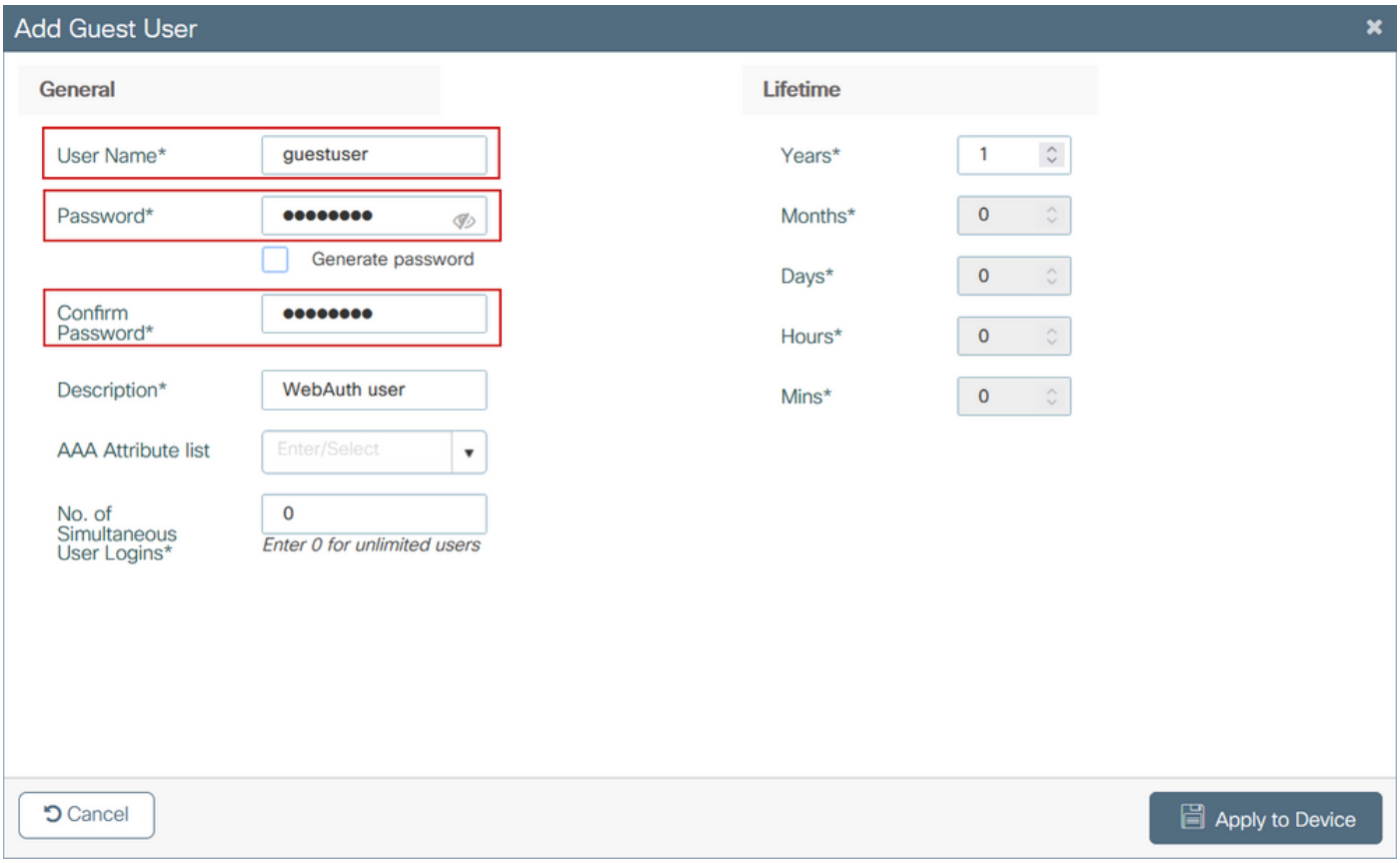
手順1および2のCLI設定：

<#root>

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

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

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

注：外部RADIUS認証が必要な場合は、9800 WLC上のRADIUSサーバの設定に関連する手順「9800 WLC上のAAA Config」をお読みください。認証方式リストにdot1xではなくタイプとして「login」が設定されていることを確認します。

ステップ 3：Configuration > Security > Guest Userの順に移動します。+ Addを選択し、ゲストユーザアカウントの詳細を設定します。



CLI による設定：

<#root>

```
9800(config)#

user-name guestuser

9800(config-user-name)#

description "WebAuth user"

9800(config-user-name)#

password 0 <password>

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 <password>
```

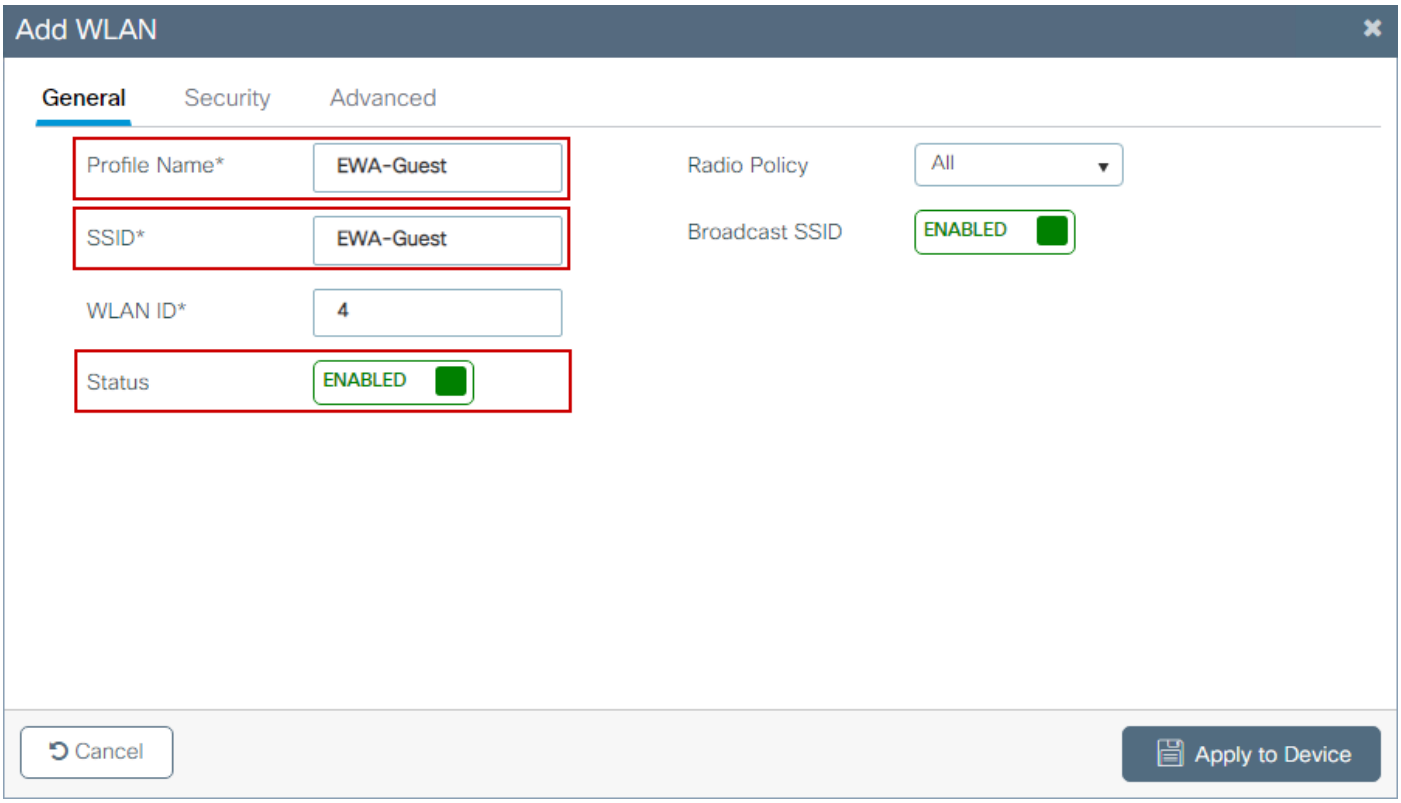
ステップ4: (オプション) パラメータマップを定義すると、複数のアクセスコントロールリスト(ACL)が自動的に作成されます。これらのACLは、どのトラフィックがWebサーバへのリダイレクションをトリガーし、どのトラフィックが通過を許可されるかを定義するために使用されます。複数のWebサーバのIPアドレスやURLフィルタなど、特定の要件が存在する場合は、Configuration > Security > ACLの順に選択し、+ Addを選択して必要なルールを定義します。deny文はトラフィックが通過することを定義しますが、permit文はリダイレクトされます。

自動作成されたACLルールは次のとおりです。

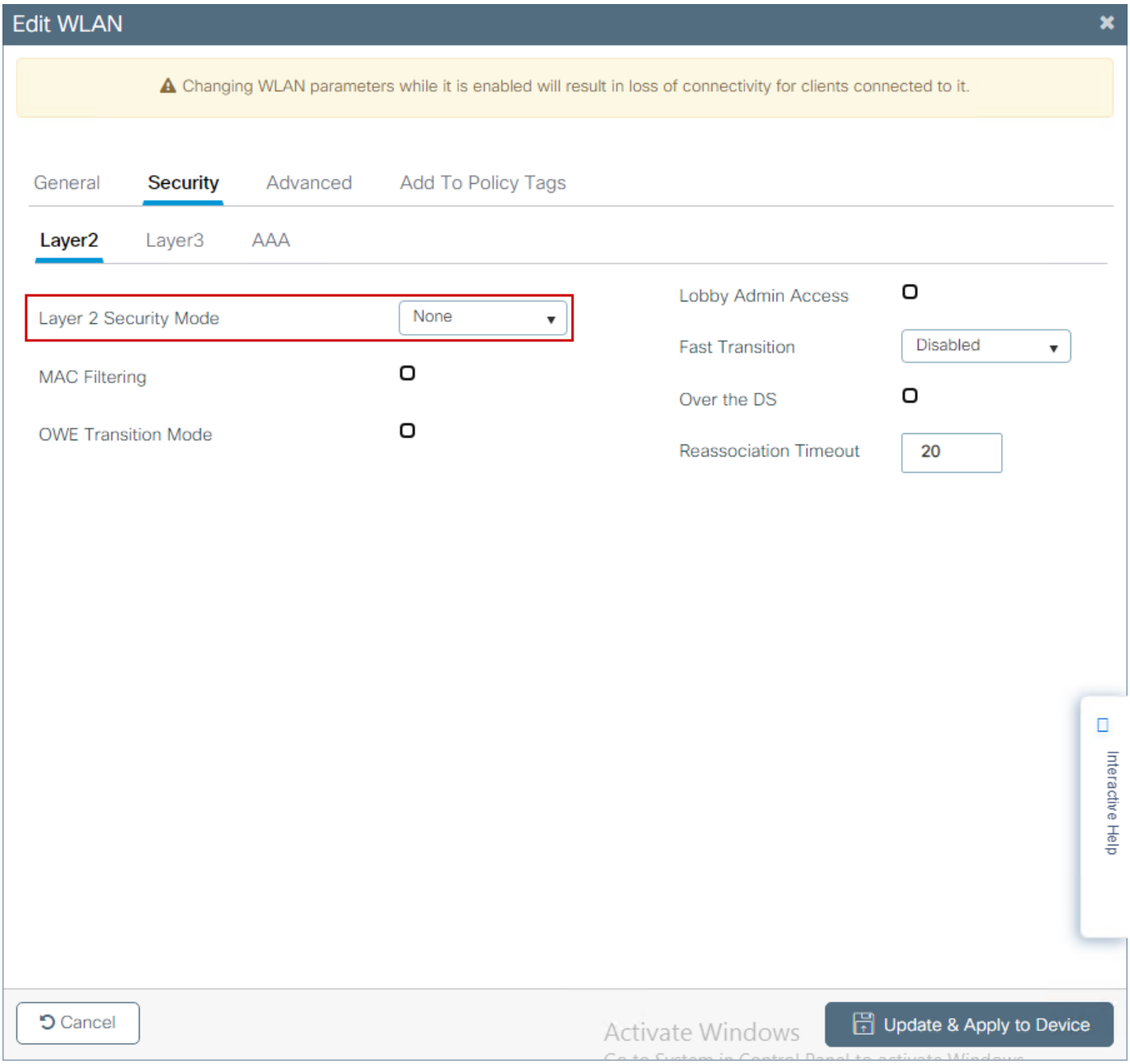
```
<#root>

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
```

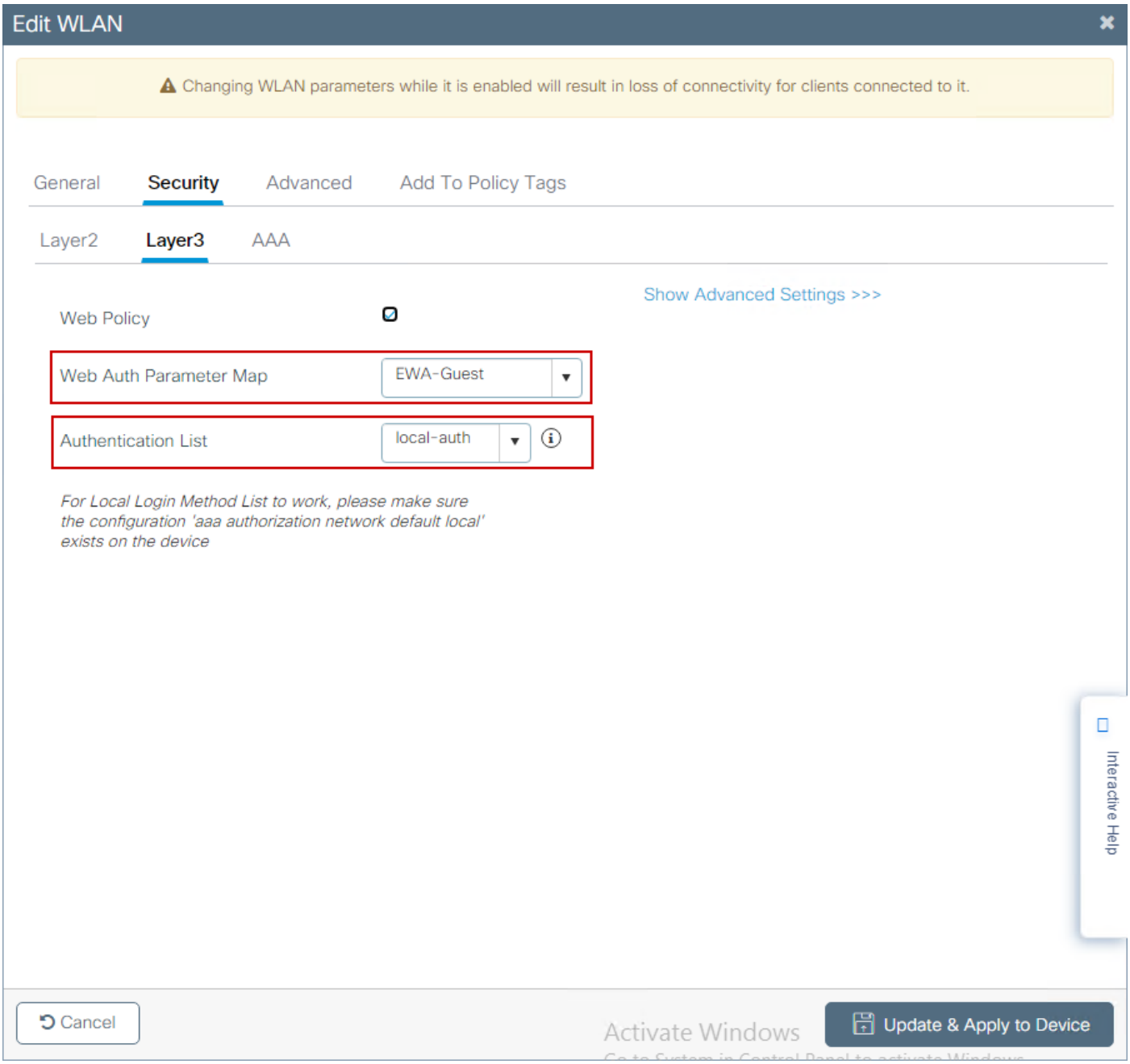
ポリシーとタグの設定

ステップ 1：Configuration > Tags & Profiles > WLANsの順に移動し、+ Addを選択して新しいWLANを作成します。Generalタブで、プロファイル、SSID名、およびStatusを定義します。



ステップ 2：Over-the-Air暗号化メカニズムが不要な場合は、Securityタブを選択して、Layer 2 authenticationをNoneに設定します。Layer 3タブで、Web Policyボックスにチェックマークを入れ、ドロップダウンメニューからパラメータマップを選択し、ドロップダウンメニューから認証リストを選択します。オプションで、カスタムACLが以前に定義されている場合、[詳細設定の表示]を選択し、ドロップダウンメニューから適切なACLを選択します。


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



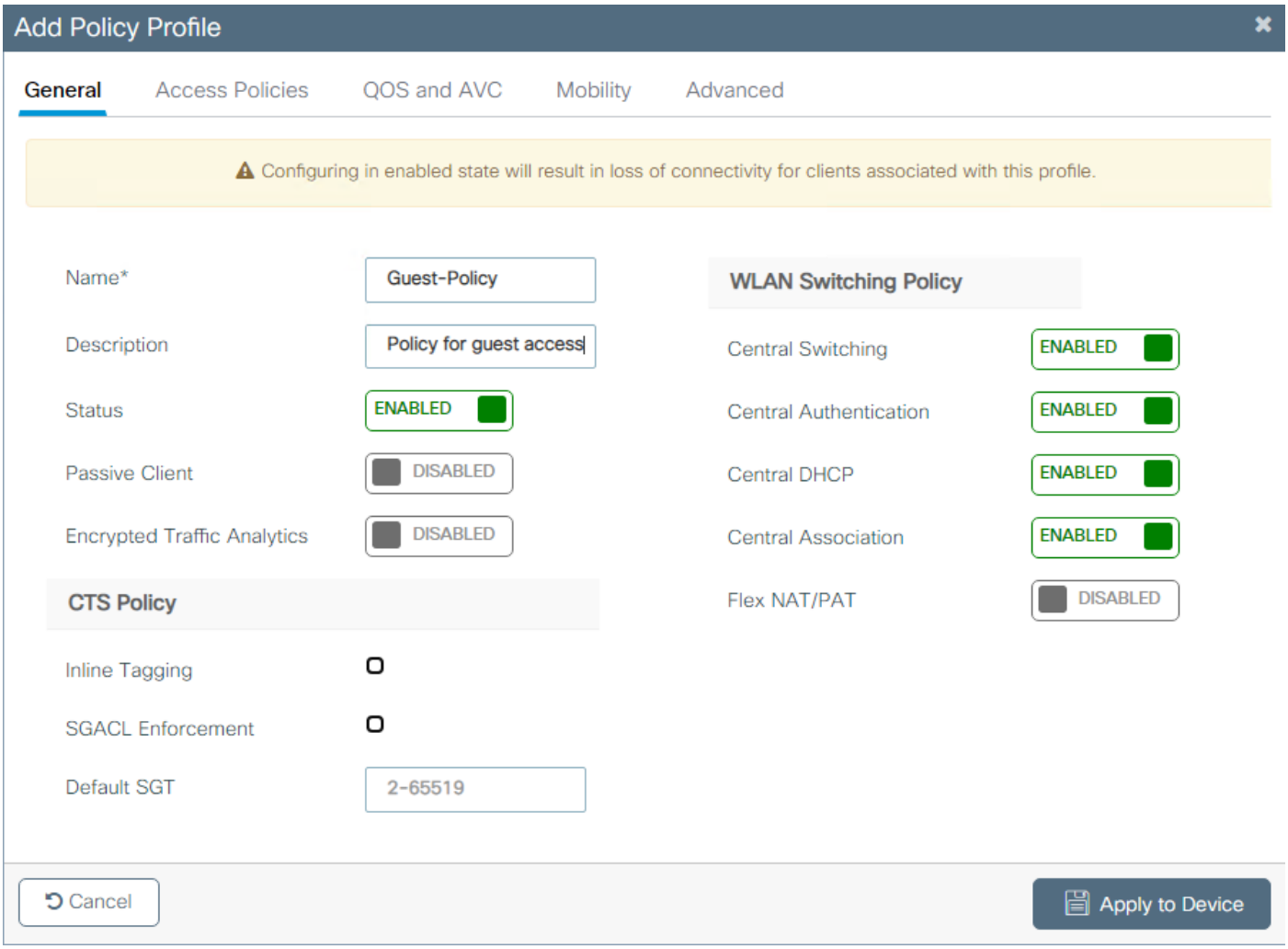
CLI の設定:

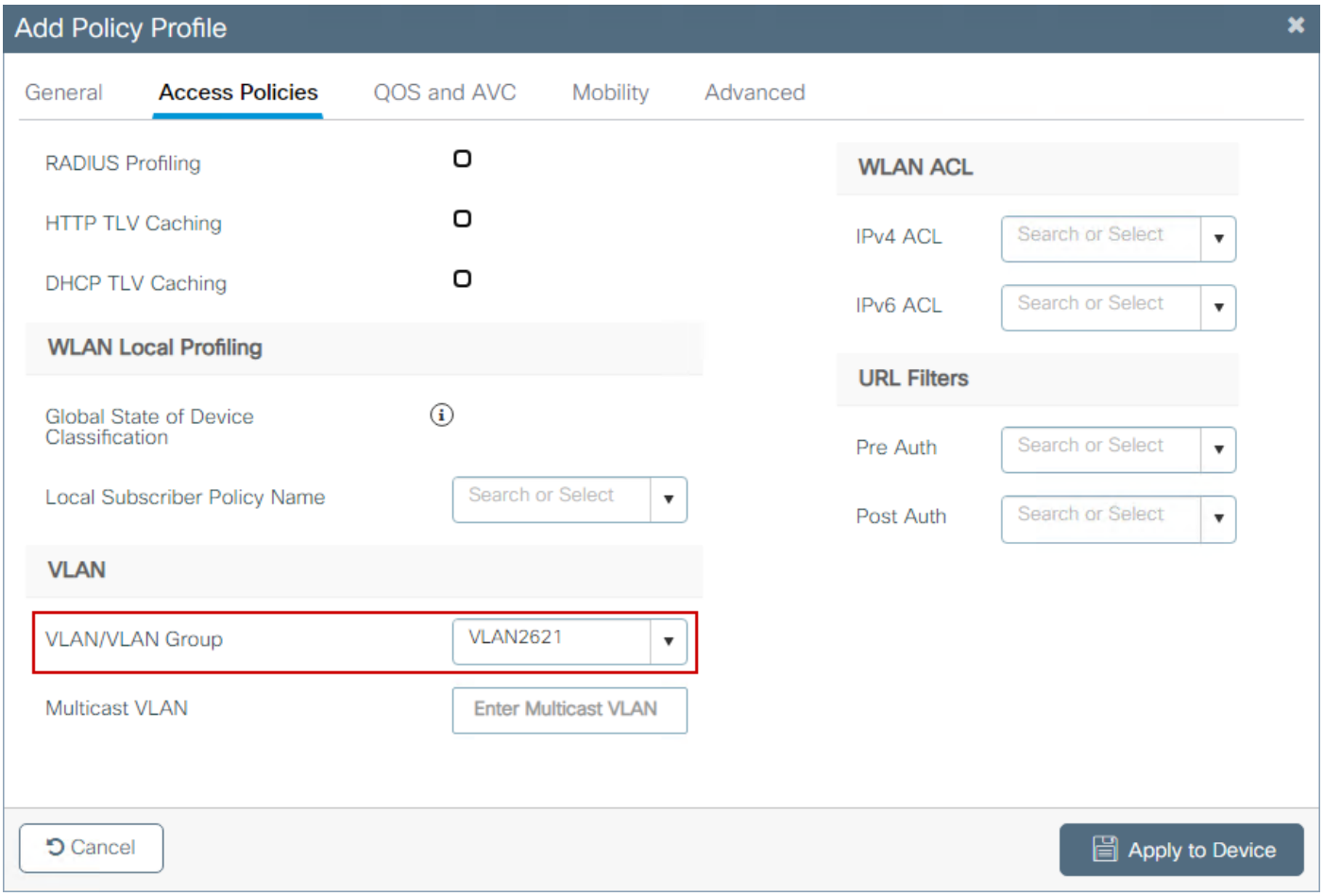
<#root>

```
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
```

ステップ 3：Configuration > Tags & Profiles > Policyの順に移動し、+ Addを選択します。ポリシーの名前とステータスを定義します。WLANスイッチングポリシーの下のCentral設定がローカルモードAPに対して有効になっていることを確認します。Access Policiesタブで、図のように、VLAN/VLAN Groupドロップダウンメニューから正しいVLANを選択します。


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

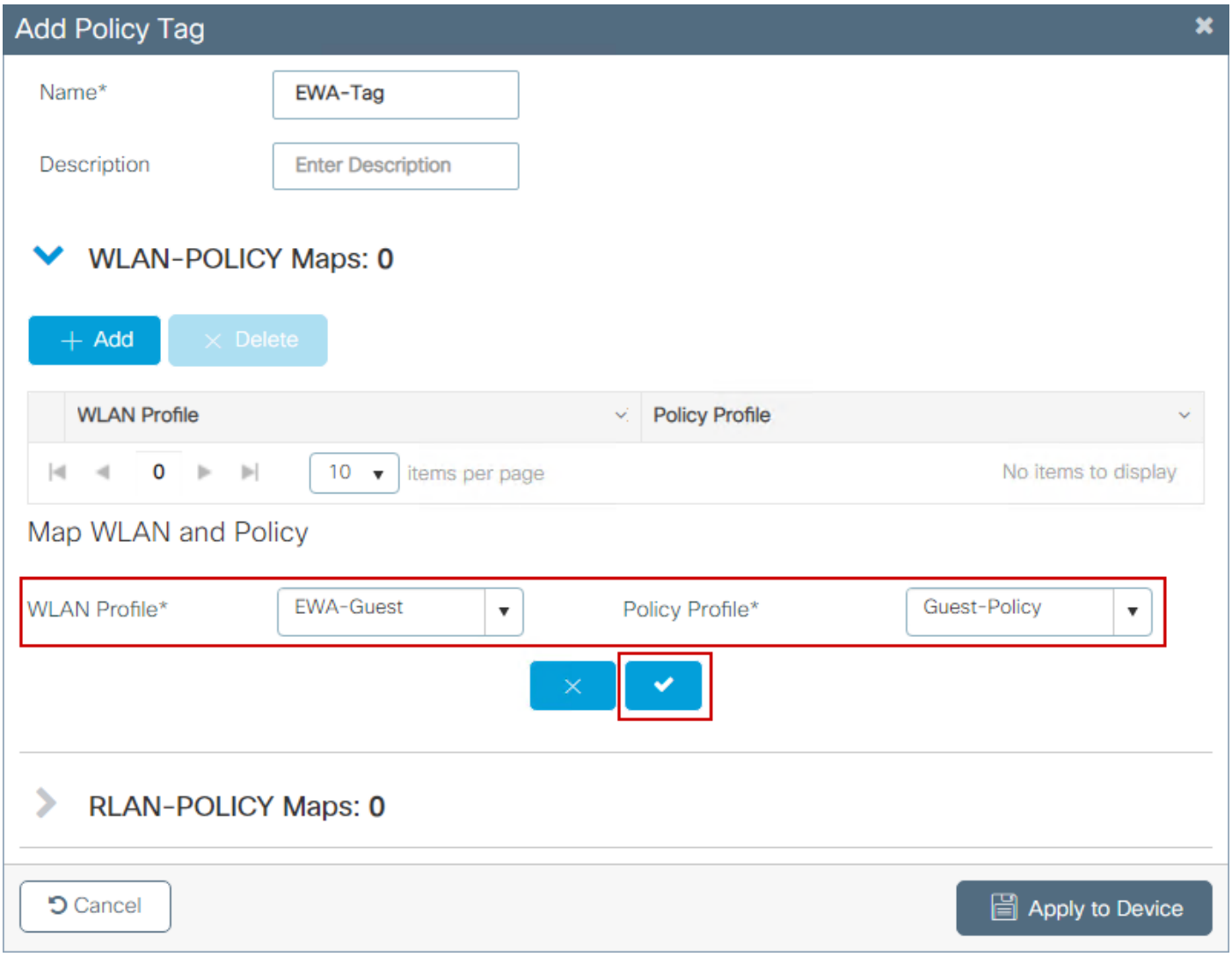


CLI による設定：

<#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
```

ステップ 4：Configuration > Tags & Profiles > Tagsの順に移動し、Policyタブで+ Addを選択します。タグ名を定義して、WLAN-POLICY Mapsで+ Addを選択し、以前に作成したWLANとポリシープロファイルを追加します。

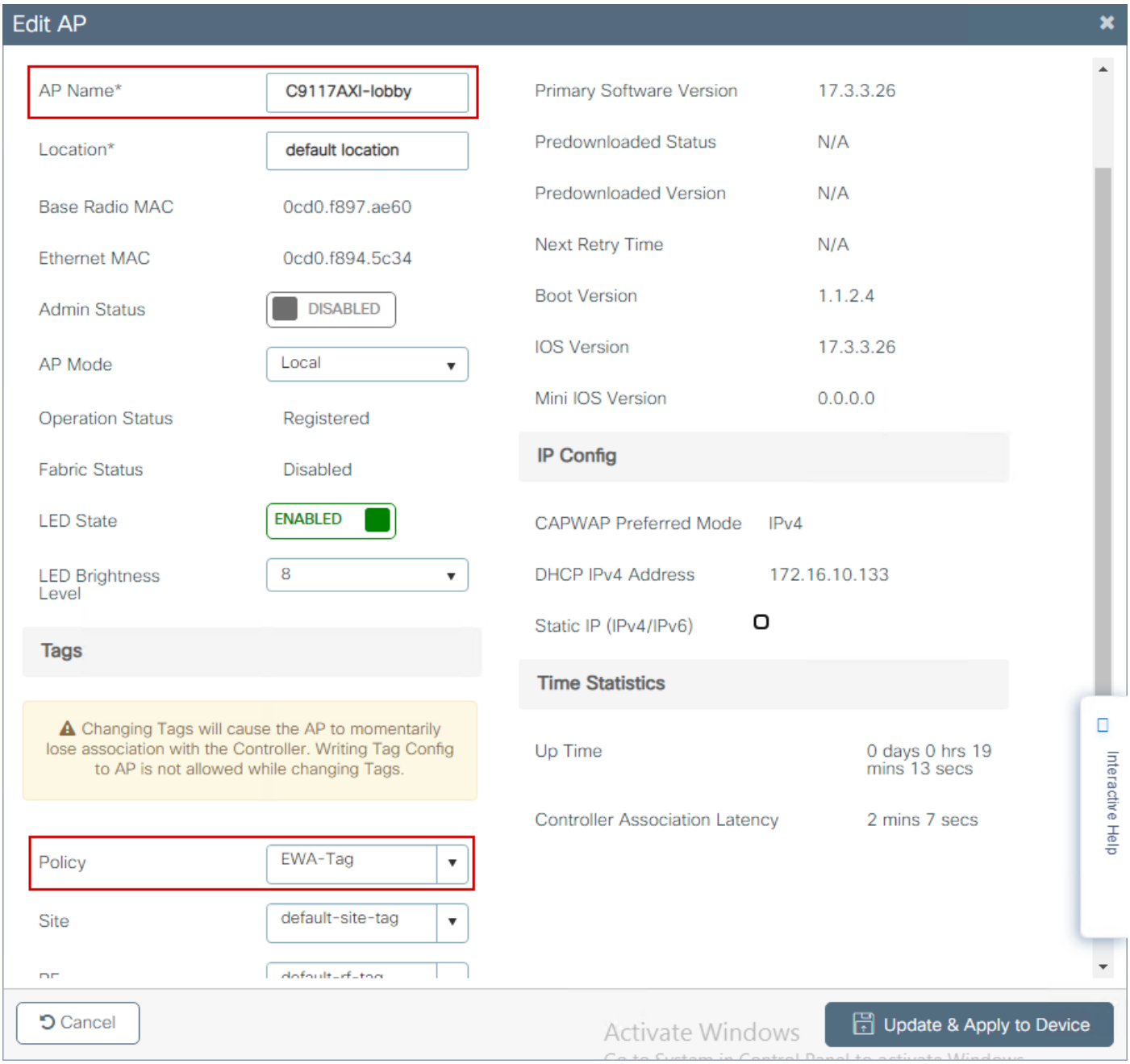


CLI による設定：

<#root>

```
9800(config)#

wireless tag policy EWA-Tag


9800(config-policy-tag)#

wlan EWA-Guest policy Guest-Policy
```

ステップ 5：Configuration > Wireless > Access Pointsの順に移動し、このSSIDのブロードキャストに使用するAPを選択します。Edit APメニューで、Policyドロップダウンメニューから新しく作成したタグを選択します。



複数のAPに同時にタグを付ける必要がある場合は、次の2つのオプションを使用できます。

オプションA. Configuration > Wireless Setup > Advancedの順に移動し、そこからStart Nowを選択して、設定メニューリストを表示します。Tag APsの横にあるリストアイコンを選択します。これにより、Join状態にあるすべてのAPのリストが表示され、必要なAPのチェックボックスをオンにしてから、+ Tag APsを選択します。次に、ドロップダウンメニューから作成したポリシータグを選択します。

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.

DESIGN PHASE

Tags & Profiles

WLAN Policy (Mandatory)	Site Policy (Optional)	Radio Policy (Optional)
WLAN Profile	AP Join Profile	RF Profile
Policy Profile	Flex Profile	RF Tag
Policy Tag	Site Tag	

DEPLOY PHASE

Apply to APs

(Mandatory)

Tag APs

Select APs and push configuration to them

TERMINOLOGY

Tag

WLAN Policy, Policy Profile

Site Policy - AP Profile, Site Profile

Radio Policy - Radio Characteristics

ACTIONS

Go to List View

Create New

```
----------------------------------------------------------------------------------------------------

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 <group index> 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
```

トラブルシュート

常時トレース

WLC 9800には常時トレース機能があります。これにより、クライアント接続に関連するすべてのエラー、警告、および通知レベルのメッセージが継続的に口グに記録され、発生後にインシデントまたは障害状態の口グを表示できます。

注：生成される口グの量に基づいて、数時間から数日に戻ることができます。

9800 WLCがデフォルトで収集したトレースを表示するには、SSH/Telnet経由で9800 WLCに接続し、次の手順を読みます（セッションをテキストファイルに記録していることを確認します）。

ステップ 1：コントローラの現在時刻を確認して、問題が発生した時刻まで口グを追跡できるようにします。

```
<#root>

9800#

show clock
```

ステップ 2：システム設定に従って、コントローラのバッファまたは外部syslogからsyslogを収集します。これにより、システムの健全性とエラー（ある場合）のクイックビューが提供されます。

<#root>

9800#

show logging

ステップ 3：デバッグ条件が有効になっているかどうかを確認します。

<#root>

9800#

show debugging

```
IOSXE Conditional Debug Configs:
Conditional Debug Global State: Stop
IOSXE Packet Tracing Configs:
Packet Infra debugs:
Ip Address                                              Port
------------------------------------------------------|----------
```

注：条件が一覧表示されている場合は、有効な条件（MACアドレス、IPアドレスなど）に遭遇するすべてのプロセスについて、トレースがデバッグレベルで記録されていることを意味します。これにより、ログの量が増加します。そのため、アクティブにデバッグを行っていない場合は、すべての条件をクリアすることを推奨します。

ステップ 4：テスト対象のMACアドレスがステップ3の条件としてリストされていないことが前提です。特定のMACアドレスのAlways-On Notice Level(AToS)トレースを収集します。

<#root>

9800#

show logging profile wireless filter [mac | ip] [<aaaa.bbbb.cccc> | <a.b.c.d>] to-file always-on-<FILENA

セッションで内容を表示するか、ファイルを外部 TFTP サーバーにコピーできます。

<#root>

9800#

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

or
9800#

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

条件付きデバッグとラジオアクティブトレース

常時オン状態のトレースでは、調査中の問題のトリガーを判別するために十分な情報が得られない場合は、条件付きデバッグを有効にして、無線アクティブ(RA)トレースをキャプチャできます。これにより、指定された条件（この場合はクライアントMACアドレス）と対話するすべてのプロセスにデバッグレベルのトレースが提供されます。条件付きデバッグを有効にするには、次の手順を参照してください。

ステップ 1：デバッグ条件が有効になっていないことを確認します。

<#root>

```
9800#

clear platform condition all
```

ステップ 2：監視するワイヤレスクライアントのMACアドレスのデバッグ条件を有効にします。

次のコマンドは、指定された MAC アドレスの 30 分間（1800 秒）のモニターを開始します。必要に応じて、この時間を最大 2085978494 秒まで増やすことができます。

<#root>

```
9800#

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

注：複数のクライアントを同時にモニタするには、MACアドレスごとにdebug wireless macコマンドを実行します。

注：後で表示できるように、すべてのログは内部でバッファリングされているため、ワイヤレスクライアントアクティビティはターミナルセッションに表示されません。

ステップ 3：監視する問題または動作を再現します。

ステップ 4：デフォルトまたは設定されたモニタ時間が経過する前に問題が再現した場合は、デバッグを停止します。

<#root>

9800#

no debug wireless mac <aaaa.bbbb.cccc>

モニター時間が経過するか、debug wireless が停止すると、9800 WLC では次の名前のローカルファイルが生成されます。

ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log

ステップ 5： MAC アドレスアクティビティのファイルを収集します。 ra trace.log を外部サーバーにコピーするか、出力を画面に直接表示できます。

RA トレースファイルの名前を確認します。

<#root>

9800#

dir bootflash: | inc ra_trace

ファイルを外部サーバーにコピーします。

<#root>

9800#

copy bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log tftp://<a.b.c.d>

内容を表示します。

<#root>

9800#

more bootflash:ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log

手順 6：根本原因がまだ明らかでない場合は、デバッグレベルのログのより詳細なビューである内部ログを収集します。このコマンドは、すでに収集されて内部で保存されているデバッグログを提供するため、クライアントを再度デバッグする必要はありません。

<#root>

9800#

show logging profile wireless internal filter [mac | ip] [<aaaa.bbbb.cccc> | <a.b.c.d>] to-file ra-inter

注：このコマンド出力は、すべてのプロセスのすべてのログレベルに関するトレースを返し、非常に大量です。これらのトレースの解析については、Cisco TACにお問い合わせください。

<#root>

9800#

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

内容を表示します。

<#root>

9800#

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

手順 7：デバッグ条件を削除します。

注：トラブルシューティングセッションの後は、必ずデバッグ条件を削除してください。

組み込みパケットキャプチャ

9800コントローラはパケットをネイティブにスニファできるため、コントロールプレーンのパケット処理の可視性などのトラブルシューティングが容易になります。

ステップ 1：対象のトラフィックをフィルタリングするACLを定義します。Web認証では、Webサーバとの間のトラフィックだけでなく、クライアントが接続されているAP間のトラフィックも許可することを推奨します。

<#root>

9800(config)#

ip access-list extended EWA-pcap

9800(config-ext-nacl)#

permit ip any host <web server IP>

9800(config-ext-nacl)#

permit ip host <web server IP> any

```
9800(config-ext-nacl)#
permit ip any host <AP IP>

9800(config-ext-nacl)#
permit ip host <AP IP> any
```

ステップ 2：モニタキャプチャパラメータを定義します。コントロールプレーントラフィックが両方向で有効になっていること、インターフェイスがコントローラの物理アップリンクを参照していることを確認します。

<#root>

```
9800#
monitor capture EWA buffer size <buffer size in MB>

9800#
monitor capture EWA access-list EWA-pcap

9800#
monitor capture EWA control-plane both interface <uplink 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)
```

ステップ 3：モニタのキャプチャを開始し、問題を再現します。

<#root>

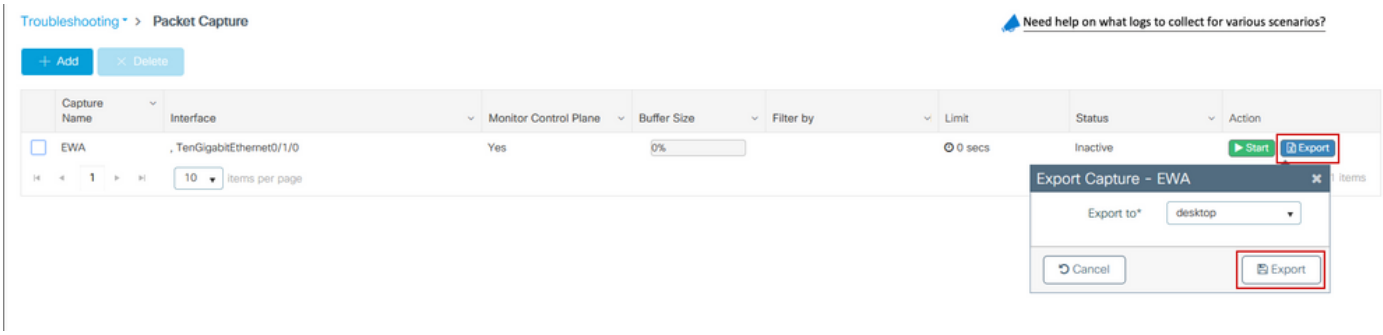
```
9800#

monitor capture EWA start

Started capture point : EWA
```

ステップ 4：モニタのキャプチャを停止し、エクスポートします。

<#root>

```
9800#

monitor capture EWA stop

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

または、GUIからキャプチャをダウンロードし、Troubleshooting > Packet Captureの順に選択し、設定したキャプチャでExportを選択します。ドロップダウンメニューからデスクトップを選択して、HTTPを介してキャプチャを目的のフォルダにダウンロードします。



クライアント側のトラブルシューティング

Web認証WLANはクライアントの動作に依存しますが、これに基づいて、クライアント側の動作の知識と情報がWeb認証の誤動作の根本原因を特定する鍵となります。

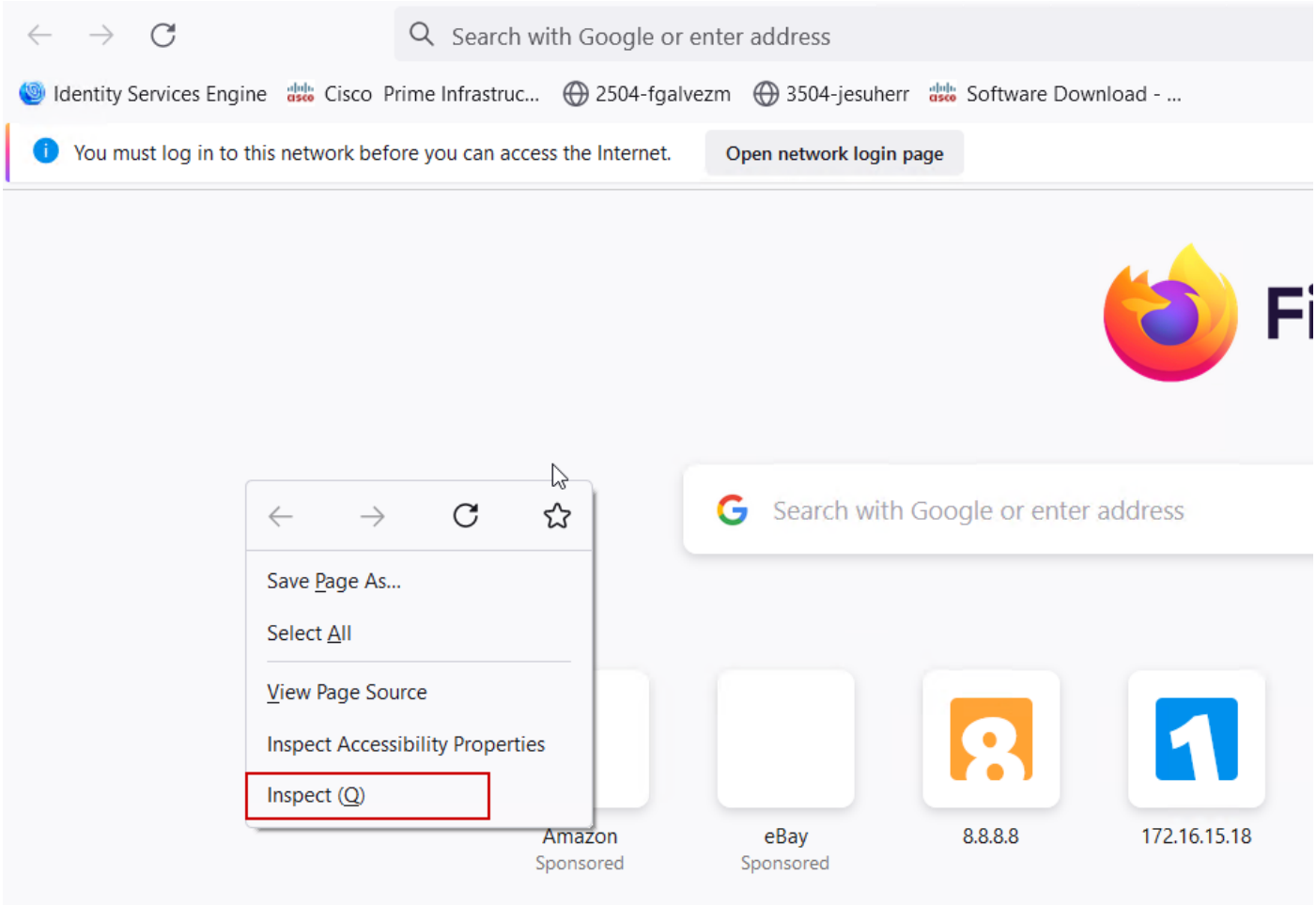
HARブラウザのトラブルシューティング

Mozilla FirefoxやGoogle Chromeなど、多くの最新ブラウザには、Webアプリケーションのインタ

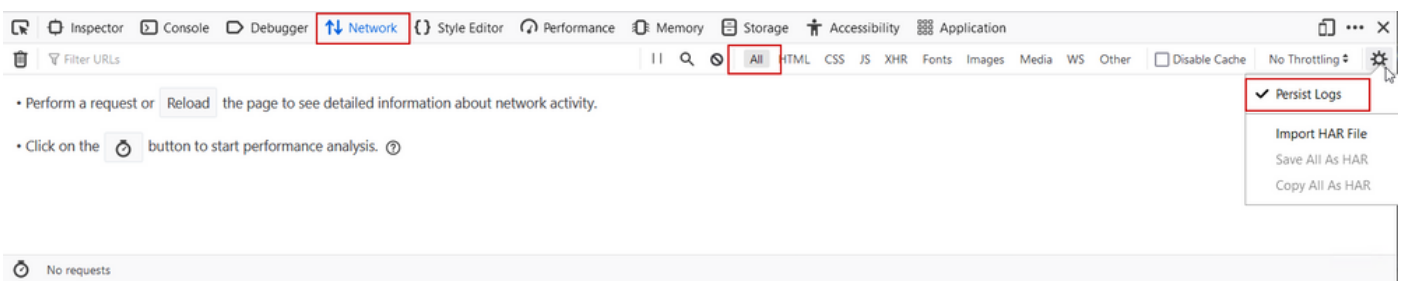
ラクションをデバッグするためのコンソール開発者ツールが用意されています。HARファイルは、クライアントとサーバ間のインタラクションの記録であり、HTTPインタラクションのタイムラインと、要求および応答情報（ヘッダー、ステータスコード、パラメータなど）を提供します。

HARファイルはクライアントブラウザからエクスポートし、別のブラウザにインポートして詳細な分析を行うことができます。このドキュメントでは、Mozilla FirefoxからHARファイルを収集する方法について説明します。

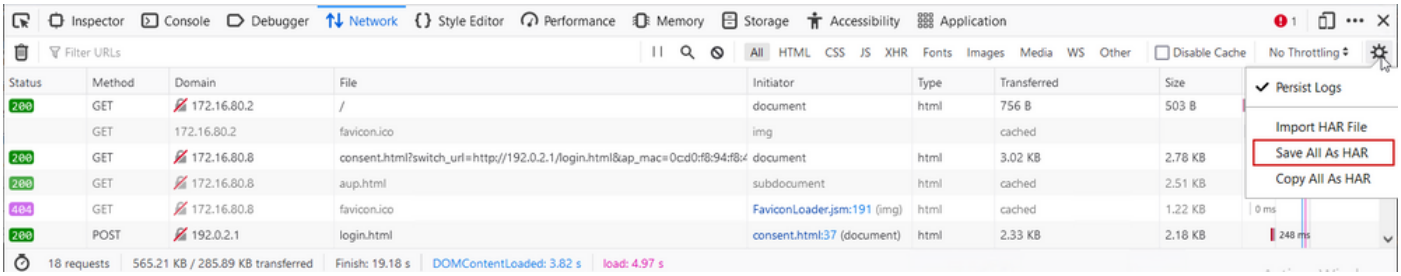
ステップ 1：Ctrl + Shift + Iキーを押しながらWeb Developer Toolsを開くか、ブラウザのコンテンツ内を右クリックしてInspectを選択します。



ステップ 2：Networkに移動し、すべての要求タイプをキャプチャするために「All」が選択されていることを確認します。歯車アイコンを選択し、Persist Logsの横に矢印があることを確認します。矢印が表示されていない場合は、ドメインの変更がトリガーされるたびにログ要求がクリアされます。



ステップ 3：問題を再現し、ブラウザがすべての要求を記録することを確認します。問題が再現されたら、ネットワークロギングを停止し、ギアアイコンでを選択して、「Save All As HAR」を選択します。



クライアント側のパケットキャプチャ

WindowsやMacOSなどのOSを搭載したワイヤレスクライアントは、ワイヤレスカードアダプタでパケットをスニファできます。Over-the-Airパケットキャプチャは直接置き換わるものではありませんが、全体的なWeb認証フローを一目で確認できます。

DNS要求：

```
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
```

リダイレクトのための初期TCPハンドシェイクと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
```

外部サーバとのTCPハンドシェイク：

```
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
```

外部サーバへのHTTP GET（キャプティブポータル要求）:

```
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
```

認証用の仮想IPへのHTTP POST:

```
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 06:44:50.749848  172.16.21.153  192.0.2.1      TCP   54   52359 → 80 [ACK] Seq=556 Ack=2412 Win=130560 Len=0
```

成功した試行の例

これは、無線アクティブトレースの観点から接続試行が成功した場合の出力です。この出力を参考にして、レイヤ3 Web認証SSIDに接続するクライアントのクライアントセッション段階を識別してください。

802.11認証および関連付け：

<#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.

レイヤ2認証がスキップされました：

<#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の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学習プロセス：

<#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 cap
[...]
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_900000
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

レイヤ3認証とリダイレクションプロセス：

<#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_900000
```

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

RUN状態への移行：

<#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

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。