

# 9800 WLC에서 외부 웹 인증 구성 및 문제 해결

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[네트워크 다이어그램](#)

[웹 매개 변수 설정 구성](#)

[CLI 구성 요약:](#)

[AAA 설정 구성](#)

[정책 및 태그 구성](#)

[다음을 확인합니다.](#)

[문제 해결](#)

[항상 추적](#)

[조건부 디버깅 및 무선 활성 추적](#)

[포함된 패킷 캡처](#)

[클라이언트측 문제 해결](#)

[HAR 브라우저 문제 해결](#)

[클라이언트측 패킷 캡처](#)

[성공한 시도의 예](#)

---


## 소개

이 문서에서는 Catalyst 9800 WLC(Wireless LAN Controller)에서 외부 웹 인증(EWA)을 구성하고 문제를 해결하는 방법에 대해 설명합니다.

## 사전 요구 사항

이 문서에서는 웹 서버가 외부 통신을 허용하도록 올바르게 구성되어 있고 웹 페이지가 WLC에서 사용자를 인증하고 클라이언트 세션을 RUN 상태로 이동하는 데 필요한 모든 매개변수를 전송하도록 올바르게 구성되어 있다고 가정합니다.

---

 **참고:** 외부 리소스 액세스는 액세스 목록 권한을 통해 WLC에 의해 제한되므로 웹 페이지에서 사용되는 모든 스크립트, 글꼴, 이미지 등을 다운로드하고 웹 서버에 로컬로 유지해야 합니다.

---

사용자 인증에 필요한 매개변수는 다음과 같습니다.

- `buttonClicked`: WLC에서 작업을 인증 시도로 탐지하려면 이 매개변수를 값 "4"로 설정해야 함

니다.

- redirectUrl: 이 매개변수의 값은 인증에 성공하면 클라이언트가 특정 웹 사이트로 이동하도록 컨트롤러에서 사용합니다.
- err\_flag: 이 매개변수는 "0"으로 설정된 인증에 성공했을 때 불완전한 정보 또는 잘못된 자격 증명과 같은 오류를 나타내는 데 사용됩니다.
- username: 이 매개변수는 webauth 매개변수 맵에만 사용되며 매개변수 맵이 consent로 설정된 경우 무시할 수 있습니다. 무선 클라이언트 사용자 이름으로 채워야 합니다.
- password: 이 매개변수는 webauth 매개변수 맵에만 사용되며 매개변수 맵이 consent로 설정된 경우 무시할 수 있습니다. 무선 클라이언트 암호로 채워야 합니다.

## 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- HTML(Hyper Text Markup Language) 웹 개발
- Cisco IOS®-XE 무선 기능
- 웹 브라우저 개발자 도구

## 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.


- C9800-CL WLC Cisco IOS®-XE 버전 17.3.3
- IIS(인터넷 정보 서비스) 기능이 있는 Microsoft Windows Server 2012
- 2802 및 9117 Access Point

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

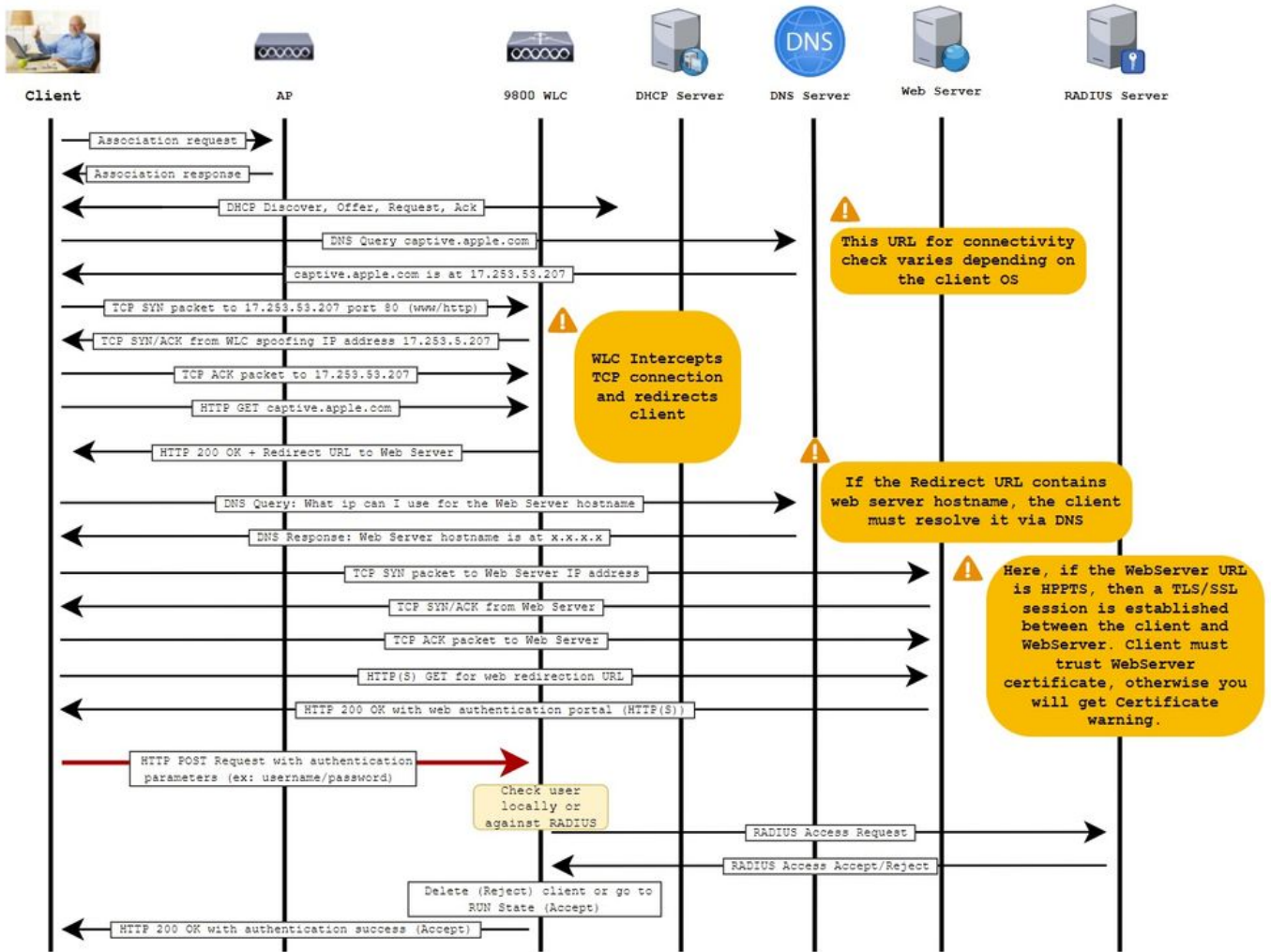
## 배경 정보

외부 웹 인증은 전용 웹 서버 또는 ISE(Identity Services Engine)와 같은 다목적 서버에서 WLC 외부에서 호스팅되는 웹 포털을 활용하므로 웹 구성 요소에 대한 세부적인 액세스 및 관리가 가능합니다. 외부 웹 인증 WLAN에 클라이언트를 성공적으로 온보딩하는 데 관련된 핸드셰이크는 이미지에서 렌더링됩니다. 이 그림에는 URL(Uniform Resource Location)을 확인하는 무선 클라이언트, WLC, DNS(Domain Name System) 서버 및 WLC가 사용자 자격 증명을 로컬로 검증하는 웹 서버 간의 순차적 상호 작용이 나열되어 있습니다. 이 워크플로는 장애 상태를 해결하는 데 유용합니다.

---

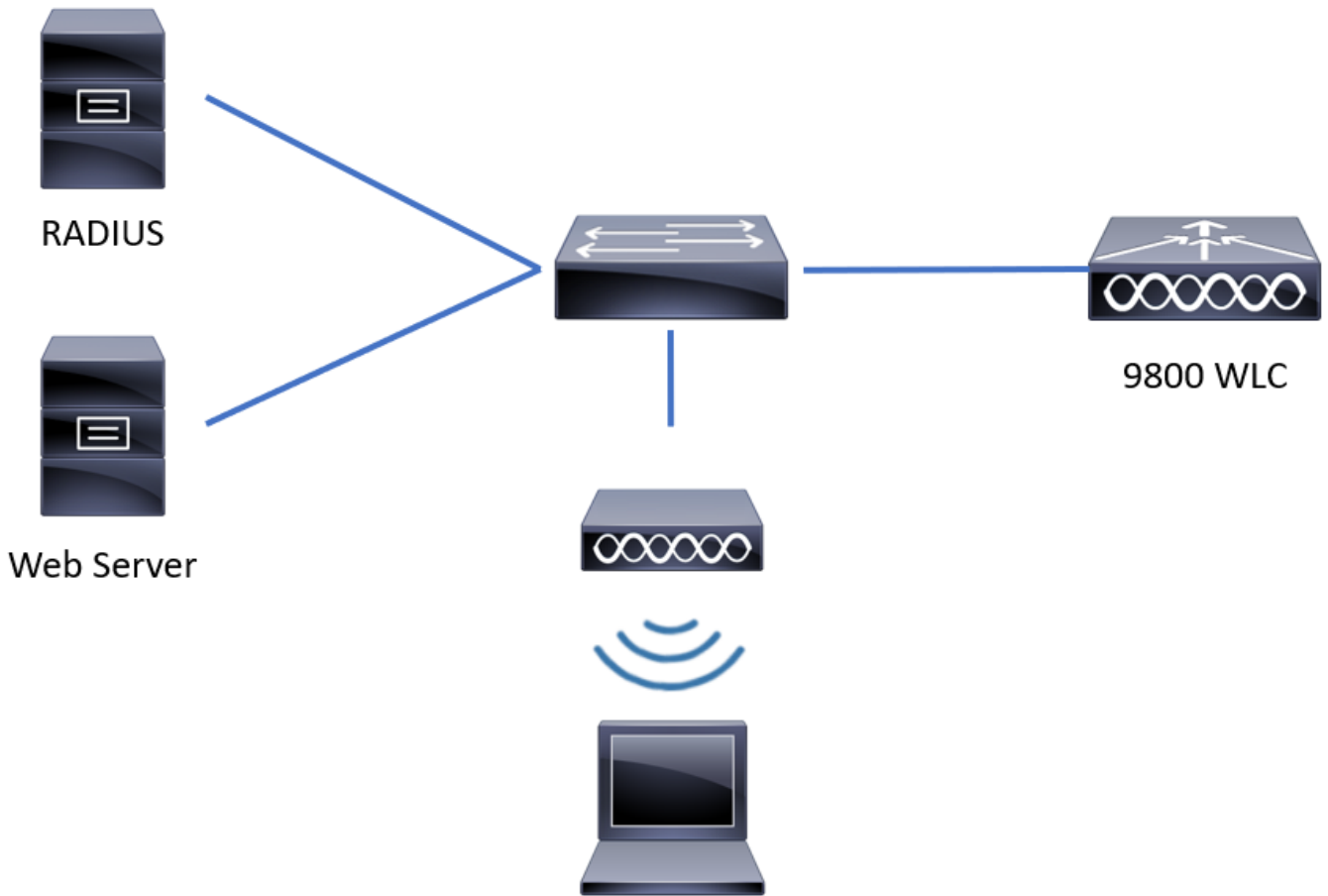
 **참고:** 클라이언트에서 WLC로 HTTP POST를 호출하기 전에 매개변수 맵에서 보안 웹 인증이 활성화되고 WLC에 신뢰할 수 있는 인증 기관에서 서명한 신뢰 지점이 없는 경우 브라우저에 보안 경고가 표시됩니다. 컨트롤러가 클라이언트 세션을 RUN 상태로 전환하려면 클라이언트가 이 경고를 무시하고 양식 재제출을 수락해야 합니다.

---




구성

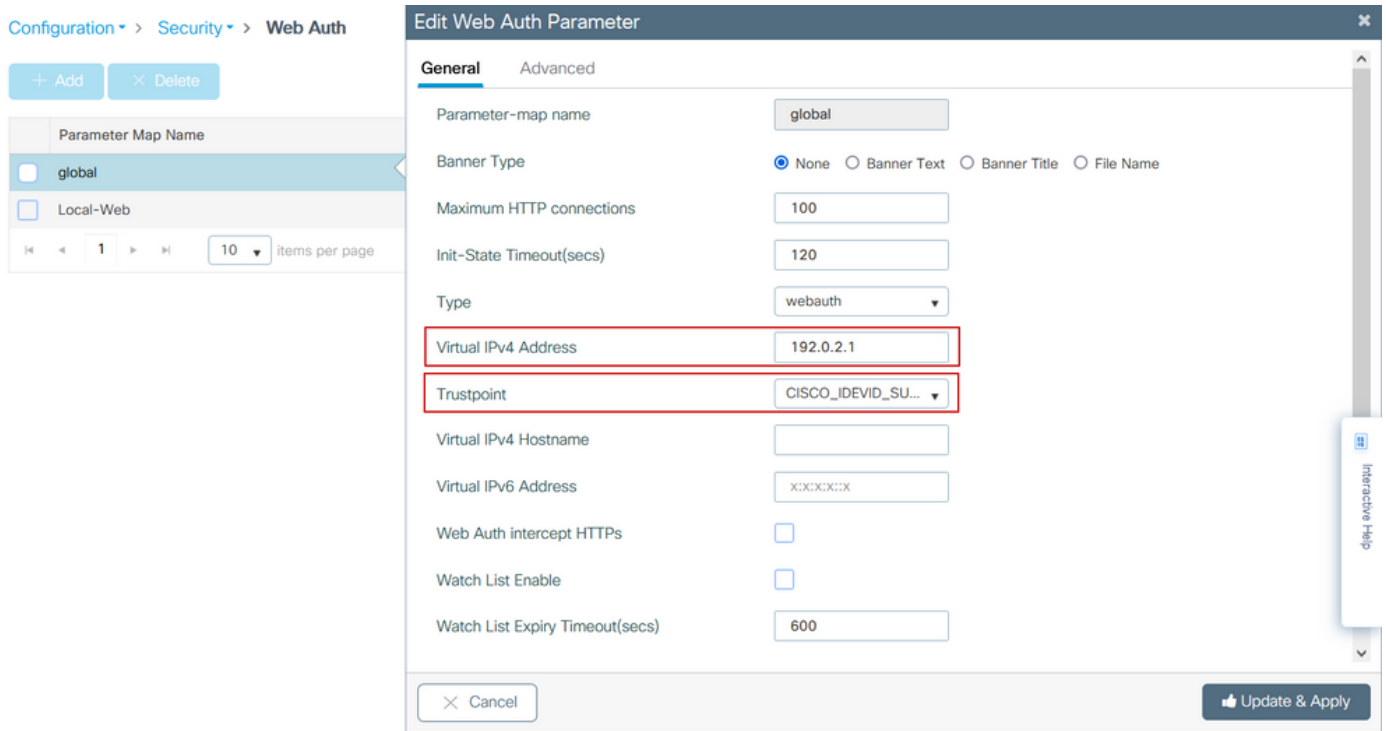
네트워크 다이어그램



## 웹 매개 변수 설정 구성

1단계. Configuration(컨피그레이션) > Security(보안) > Web Auth(웹 인증)로 이동하고 전역 매개변수 맵을 선택합니다. 적절한 리디렉션 기능을 제공하기 위해 가상 IPv4 주소 및 신뢰 지점이 구성되어 있는지 확인합니다.

 참고: 기본적으로 브라우저에서는 HTTP 웹 사이트를 사용하여 리디렉션 프로세스를 시작합니다. HTTPS 리디렉션이 필요한 경우 웹 인증 가로채기 HTTP를 선택해야 합니다. 그러나 이 컨피그레이션은 CPU 사용량을 늘리기 때문에 권장되지 않습니다.



CLI 구성:

```
<#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
```

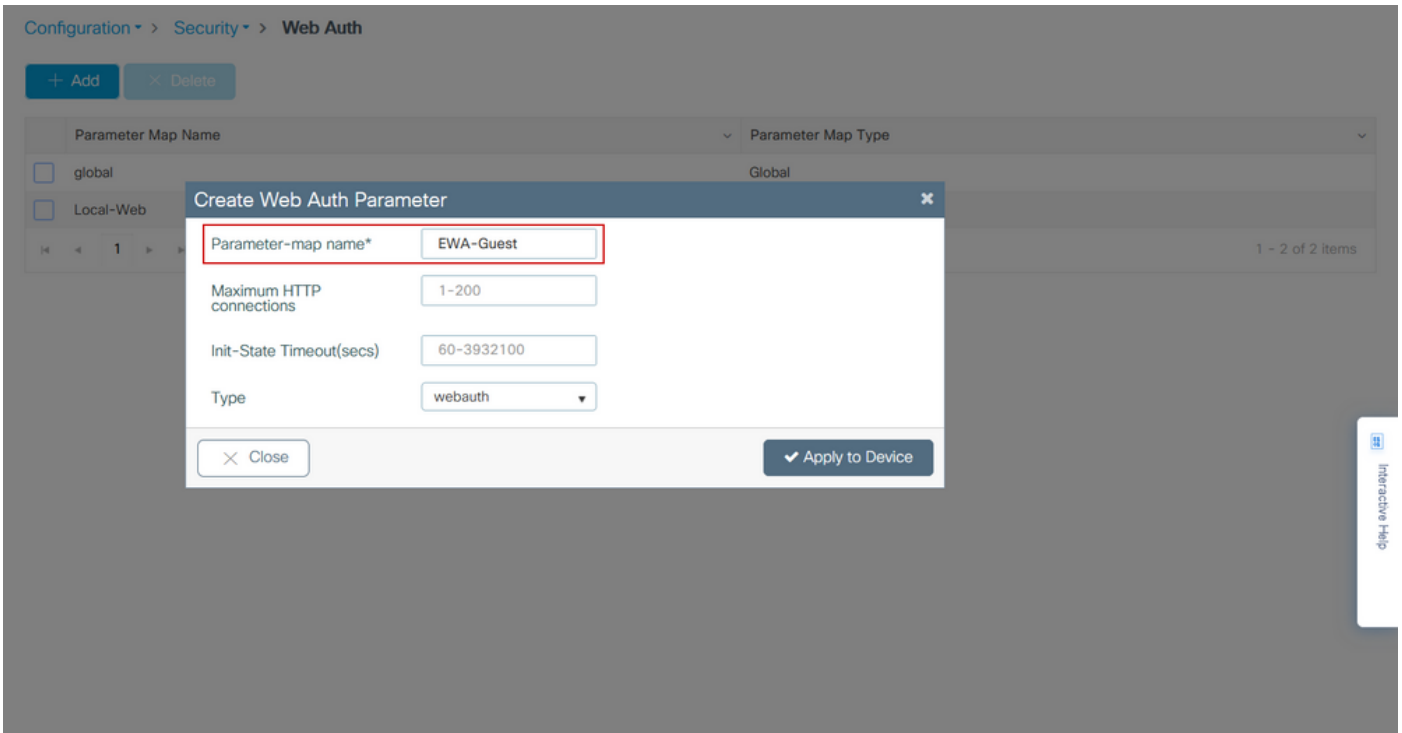
```
9800(config-params-parameter-map)#
```

```
secure-webauth-disable
```

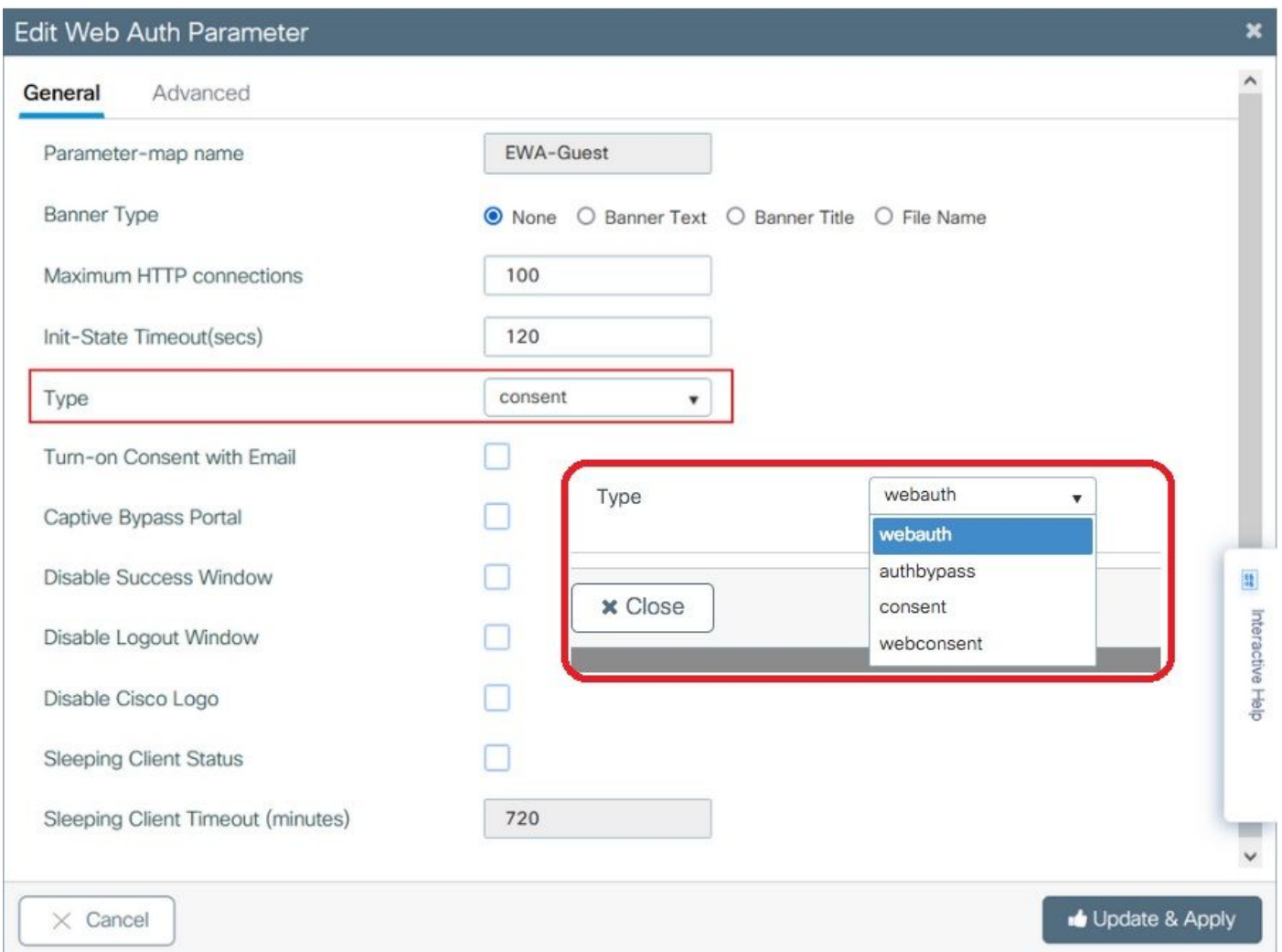
```
9800(config-params-parameter-map)#
```

```
webauth-http-enable
```

2단계. 외부 서버를 가리키는 새 매개변수 맵의 이름을 추가(+ Add)를 선택하고 구성합니다. 선택적으로, 클라이언트가 제외되기 전에 최대 HTTP 인증 실패 횟수 및 클라이언트가 웹 인증 상태를 유지할 수 있는 시간(초)을 구성합니다.



3단계. General(일반) 탭 내에서 새로 생성된 매개변수 맵을 선택하고 Type(유형) 드롭다운 목록에서 인증 유형을 구성합니다.



- Parameter-map name = WebAuth 매개변수 맵에 할당된 이름
- 최대 HTTP 연결 수 = 클라이언트가 제외되기 전 인증 실패 횟수
- Init-State Timeout(초) = 클라이언트가 웹 인증 상태에 있을 수 있는 시간(초)
- 유형 = 웹 인증 유형

웹 인증	authbypass	동의	웹 동의
<p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="OK"/></p>	<p>클라이언트가 SSID를 사용하여 IP 주소를 가져온 다음 9800 WLC MAC 주소가 이(가) 네트워크, 대답이 "예"인 경우 이동 실행 상태가 아닌 경우 참가할 수 없습니다. (웹 인증으로 돌아가지 않음)</p>	<p>banner1  <input checked="" type="radio"/> Accept  <input type="radio"/> Don't Accept</p> <p><input type="button" value="OK"/></p>	<p>banner login  <input checked="" type="radio"/> Accept  <input type="radio"/> Don't Accept</p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="OK"/></p>

4단계. Advanced(고급) 탭에서 특정 서버 사이트 URL 및 IP 주소로 각각 로그인 및 포털 IPV4 주소에 대한 리디렉션을 구성합니다.

Edit Web Auth Parameter
✕

General
Advanced

**Redirect to external server**

Redirect for log-in	<input style="width: 90%;" type="text" value="http://172.16.80.8/w"/>
Redirect On-Success	<input style="width: 90%;" type="text"/>
Redirect On-Failure	<input style="width: 90%;" type="text"/>
Redirect Append for AP MAC Address	<input style="width: 90%;" type="text" value="ap_mac"/>
Redirect Append for Client MAC Address	<input style="width: 90%;" type="text" value="client_mac"/>
Redirect Append for WLAN SSID	<input style="width: 90%;" type="text" value="ssid"/>
Portal IPV4 Address	<input style="width: 90%;" type="text" value="172.16.80.8"/>
Portal IPV6 Address	<input style="width: 90%;" type="text" value="X::X:X::X"/>
Express WiFi Key Type	<input style="width: 90%;" type="text" value="--- Select ---"/>

**Customized page**

Login Failed Page	<input style="width: 90%;" type="text"/>
-------------------	--

✕ Cancel
👍 Update & Apply

Interactive Help

2, 3, 4단계의 CLI 컨피그레이션:

```

<#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

```

5단계. (선택 사항) WLC는 쿼리 문자열을 통해 추가 매개변수를 전송할 수 있습니다. 이는 9800을 서드파티 외부 포털과 호환되도록 하기 위해 필요한 경우가 많습니다. "Redirect Append for AP MAC Address", "Redirect Append for Client MAC Address" 및 "Redirect Append for WLAN SSID" 필드를 사용하면 사용자 지정 이름을 사용하여 리디렉션 ACL에 추가 매개변수를 추가할 수 있습니



다. 새로 생성된 매개변수 맵을 선택하고 Advanced(고급) 탭으로 이동하여 필요한 매개변수의 이름을 구성합니다. 사용 가능한 매개 변수는 다음과 같습니다.

- AP MAC 주소(aa:bb:cc:dd:ee:ff 형식)
- 클라이언트 MAC 주소(aa:bb:cc:dd:ee:ff 형식)
- SSID 이름

**Edit Web Auth Parameter**

General **Advanced**

**Redirect to external server**

Redirect for log-in	<input type="text" value="http://172.16.80.8/we"/>
Redirect On-Success	<input type="text"/>
Redirect On-Failure	<input type="text"/>
Redirect Append for AP MAC Address	<input type="text" value="ap_mac"/>
Redirect Append for Client MAC Address	<input type="text" value="client_mac"/>
Redirect Append for WLAN SSID	<input type="text" value="ssid"/>
Portal IPV4 Address	<input type="text" value="172.16.80.8"/>
Portal IPV6 Address	<input type="text" value="x:x:x:x:x"/>
Express WiFi Key Type	<input type="text" value="--- Select ---"/>

**Customized page**

Login Failed Page	<input type="text"/>	
Login Page	<input type="text"/>	
Logout Page	<input type="text"/>	
Login Successful Page	<input type="text"/>	

Activate Windows  
Go to System in Control Panel to activate Windows.

Interactive Help

CLI 구성:

```
<#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
```

이 예에서 클라이언트로 전송된 리디렉션 URL은 다음과 같은 결과를 초래합니다.


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

---

 참고: 포털 IPv4 주소 정보를 추가하면 무선 클라이언트의 HTTP 및 HTTPS 트래픽을 허용하는 ACL이 자동으로 외부 웹 인증 서버에 추가되므로 추가 사전 인증 ACL을 구성할 필요가 없습니다. 여러 IP 주소 또는 URL을 허용하려는 경우 인증을 수행하기 전에 지정된 URL과 일치하는 IP가 허용되도록 URL 필터를 구성하는 옵션만 있습니다. URL 필터를 사용하지 않는 한 둘 이상의 포털 IP 주소를 정적으로 추가할 수 없습니다.

---

---

 참고: 전역 매개변수 맵은 가상 IPv4 및 IPv6 주소, Webauth 가로채기 HTTP, 종속 바이패스 포털, 감시 목록 활성화 및 감시 목록 만료 시간 제한 설정을 정의할 수 있는 유일한 맵입니다.

---

CLI 구성 요약:

로컬 웹 서버

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

외부 웹 서버

```
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 설정 구성

이 컨피그레이션 섹션은 webauth 또는 webconsent 인증 유형에 대해 구성된 매개변수 맵에만 필요합니다.

1단계. Configuration(컨피그레이션) > Security(보안) > AAA로 이동한 다음 AAA Method List(AAA 메서드 목록)를 선택합니다. 새 방법 목록을 구성하고 + Add(추가)를 선택하고 목록 세부사항을 입력합니다. 이미지에 표시된 대로 Type(유형)이 "login(로그인)"으로 설정되어 있는지 확인합니다.

2단계. Authorization(권한 부여)을 선택한 다음 + Add(추가)를 선택하여 새 메서드 목록을 만듭니다. 이미지에 표시된 대로 Type(유형)을 network(네트워크)로 기본 이름으로 지정합니다.

참고: [WLAN 레이어 3 보안 컨피그레이션](#) 중 컨트롤러에 의해 광고되므로: Local Login Method List(로컬 로그인 방법 목록)가 작동하려면 디바이스에 'aaa authorization network default local' 컨피그레이션이 있는지 확인하십시오. 즉, 로컬 웹 인증을 올바르게 구성하려면 기본적으로 이름을 갖는 권한 부여 방법 목록을 정의해야 합니다. 이 섹션에서는 이 특정 권한 부여 방법 목록을 구성합니다.

Configuration > Security > AAA Show Me How >

+ AAA Wizard

Servers / Groups **AAA Method List** AAA Advanced

Authentication

**Authorization**

Accounting

+ Add × Delete

Name	Type	Group Type	Group1	Group2	Group3	Group4
alzlab-rad-authz	network	group	alzlab-rad	N/A	N/A	N/A
wcm_loc_serv_cert	credential-download	local	N/A	N/A	N/A	N/A

10 items per page 1 - 2 of 2 items

Quick Setup: AAA Authorization ✕

Method List Name\* default

Type\* network ▼ ⓘ

Group Type local ▼ ⓘ

Authenticated

Available Server Groups

radius  
 ldap  
 tacacs+  
 alzlab-rad  
 fgalvezm-group

>  
<  
>>  
<<

Assigned Server Groups

(Empty)

^  
^  
v  
v

↶ Cancel

📄 Apply to Device

1단계와 2단계의 CLI 컨피그레이션:

<#root>

```
9800(config)#
```

```
aaa new-model
```

```
9800(config)#
```

```
aaa authentication login local-auth local
```

```
9800(config)#
```

```
aaa authorization network default local
```

---

**참고:** 외부 RADIUS 인증이 필요한 경우, 9800 WLC의 RADIUS 서버 컨피그레이션(9800 WLC의 [AAA 컨피그레이션](#))과 관련된 다음 [지침을 참조하십시오](#). 인증 방법 목록에 dot1x 대신 유형으로 설정된 "login"이 있는지 확인하십시오.

---

3단계. Configuration(컨피그레이션) > Security(보안) > Guest User(게스트 사용자)로 이동합니다. + Add(추가)를 선택하고 게스트 사용자 어카운트 세부 정보를 구성합니다.

Add Guest User
✕

General

User Name\*

Password\*

👁

Generate password

Confirm Password\*

Description\*

AAA Attribute list

No. of Simultaneous User Logins\*

Enter 0 for unlimited users

Lifetime

Years\*

Months\*

Days\*

Hours\*

Mins\*

↶ Cancel

📄 Apply to Device

## CLI 구성:

```
<#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>
```

4단계(선택 사항) 매개변수 맵 정의 시 두 개의 ACL(Access Control List)이 자동으로 생성됩니다. 이러한 ACL은 웹 서버로의 리디렉션을 트리거할 트래픽과 통과가 허용되는 트래픽을 정의하는 데

사용됩니다. 여러 웹 서버 IP 주소 또는 URL 필터와 같은 특정 요구 사항이 있는 경우 Configuration > Security > ACL select + Add로 이동하여 필요한 규칙을 정의합니다. deny 명령문이 통과 트래픽을 정의하는 동안 permit 명령문이 리디렉션됩니다.

자동으로 생성되는 ACL 규칙은 다음과 같습니다.

```
<#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
```

## 정책 및 태그 구성

1단계. Configuration(컨피그레이션) > Tags & Profiles(태그 및 프로필) > WLANs(WLAN)로 이동하고 + Add(추가)를 선택하여 새 WLAN을 생성합니다. General(일반) 탭에서 프로파일 및 SSID 이름, 상태를 정의합니다.

Add WLAN
✕

---

General
Security
Advanced

Profile Name\*

SSID\*

WLAN ID\*

Status ENABLED

Radio Policy All ▼

Broadcast SSID ENABLED

↶ Cancel

📄
Apply to Device

2단계. Air 암호화 메커니즘에서 Layer 2 인증이 필요하지 않은 경우 Security(보안) 탭을 선택하고 Layer 2 인증을 None(없음)으로 설정합니다. Layer 3(레이어 3) 탭에서 Web Policy(웹 정책) 상자를 선택하고 드롭다운 메뉴에서 매개변수 맵을 선택한 다음 드롭다운 메뉴에서 인증 목록을 선택합니다. 선택적으로, 사용자 지정 ACL이 이전에 정의된 경우 Show Advanced Settings를 선택하고 드롭다운 메뉴에서 적절한 ACL을 선택합니다.



⚠ 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

MAC Filtering

OWE Transition Mode

Lobby Admin Access

Fast Transition

Over the DS

Reassociation Timeout

Interactive Help

Cancel

Activate Windows

Go to System in Control Panel to activate Windows

Update & Apply to Device

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

Web Policy  [Show Advanced Settings >>>](#)

Web Auth Parameter Map EWA-Guest ▼

Authentication List local-auth ▼ ⓘ

*For Local Login Method List to work, please make sure the configuration 'aaa authorization network default local' exists on the device*

↶ Cancel Activate Windows Update & Apply to Device

[Interactive Help](#)

CLI 구성:

```
<#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
```

3단계. Configuration(컨피그레이션) > Tags & Profiles(태그 및 프로파일) > Policy(정책)로 이동하고 + Add(추가)를 선택합니다. 정책 이름 및 상태를 정의합니다. WLAN Switching Policy(WLAN 스위칭 정책) 아래의 Central(중앙) 설정이 Local(로컬) 모드 AP에 대해 Enabled(활성화됨)인지 확인합니다. 이미지에 표시된 대로 Access Policies(액세스 정책) 탭의 VLAN/VLAN Group(VLAN/VLAN 그룹) 드롭다운 메뉴에서 올바른 VLAN을 선택합니다.

## 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

Add Policy Profile
✕

General   **Access Policies**   QOS and AVC   Mobility   Advanced

RADIUS Profiling

HTTP TLV Caching

DHCP TLV Caching

**WLAN Local Profiling**

Global State of Device Classification ⓘ

Local Subscriber Policy Name

**VLAN**

VLAN/VLAN Group

Multicast VLAN

**WLAN ACL**

IPv4 ACL

IPv6 ACL

**URL Filters**

Pre Auth

Post Auth

↶ Cancel

📄 Apply to Device

CLI 구성:

```
<#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
```

4단계. Configuration(컨피그레이션) > Tags & Profiles(태그 및 프로필) > Tags(태그)로 이동하고 Policy(정책) 탭 내에서 + Add(추가)를 선택합니다. 태그 이름을 정의한 다음 WLAN-POLICY Maps(WLAN-POLICY 맵)에서 + Add(추가)를 선택하고 이전에 생성한 WLAN 및 Policy Profile(정책 프로파일)을 추가합니다.

**Add Policy Tag** ✕

Name\*

Description

▼ **WLAN-POLICY Maps: 0**

+ Add ✕ Delete

WLAN Profile	Policy Profile
<span>⏪</span> <span>⏩</span> <span>0</span> <span>▶</span> <span>⏪</span> <span>10</span> items per page <span style="float: right;">No items to display</span>	

Map WLAN and Policy

WLAN Profile\*  Policy Profile\*

✕
✓

---

➤ **RLAN-POLICY Maps: 0**

↶ Cancel
Apply to Device

CLI 구성:

```
<#root>
```

```
9800(config)#
```

```
wireless tag policy EWA-Tag
```

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

```
wlan EWA-Guest policy Guest-Policy
```

5단계. Configuration(컨피그레이션) > Wireless(무선) > Access Points(액세스 포인트)로 이동하고 이 SSID를 브로드캐스트하는 데 사용되는 AP를 선택합니다. Edit AP(AP 수정) 메뉴의 Policy(정책) 드롭다운 메뉴에서 새로 생성된 태그를 선택합니다.

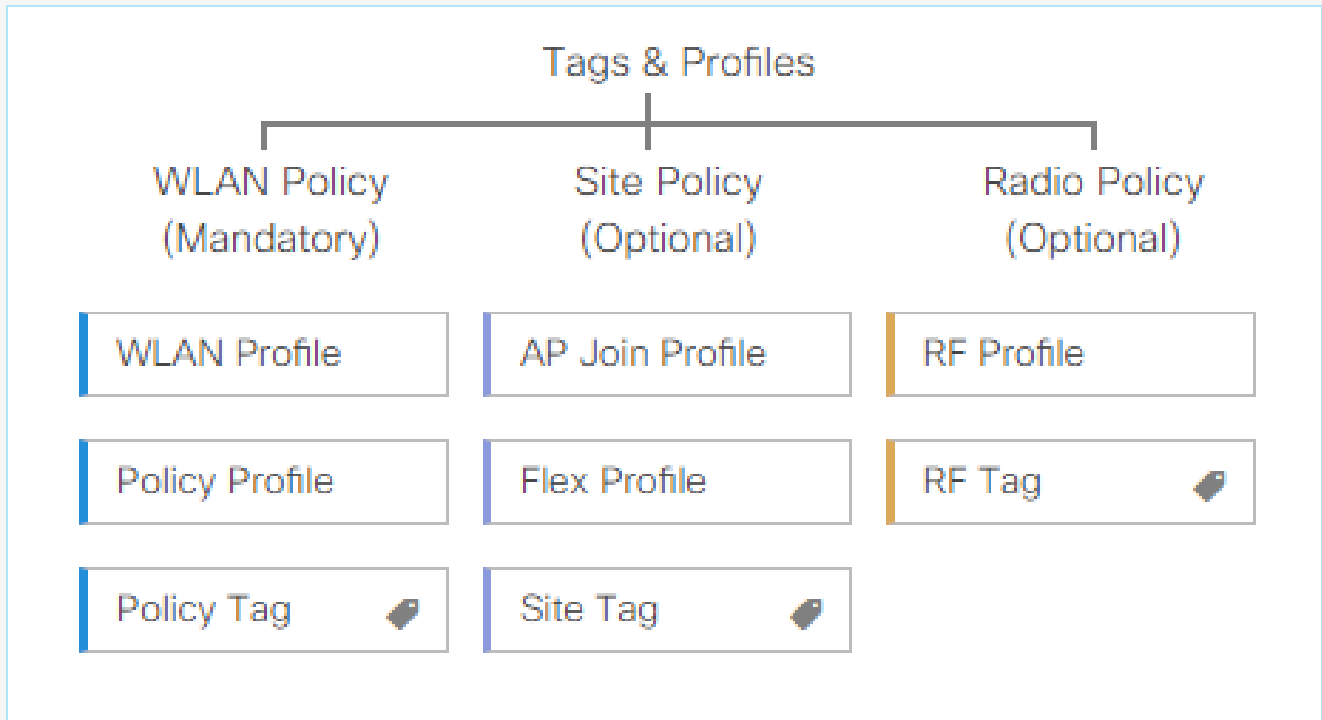
여러 AP에 동시에 태그를 지정해야 하는 경우 두 가지 옵션을 사용할 수 있습니다.

옵션 A. Configuration(구성) > Wireless Setup(무선 설정) > Advanced(고급)로 이동합니다. 여기서 Start Now(지금 시작)를 선택하여 구성 메뉴 목록을 표시합니다. Tag APs(태그 AP) 옆에 있는 목록 아이콘을 선택하면, Join(조인) 상태의 모든 AP 목록이 표시되고, 필요한 AP를 확인한 다음 + Tag APs(태그 AP)를 선택하고, 드롭다운 메뉴에서 생성된 Policy Tag(정책 태그)를 선택합니다.

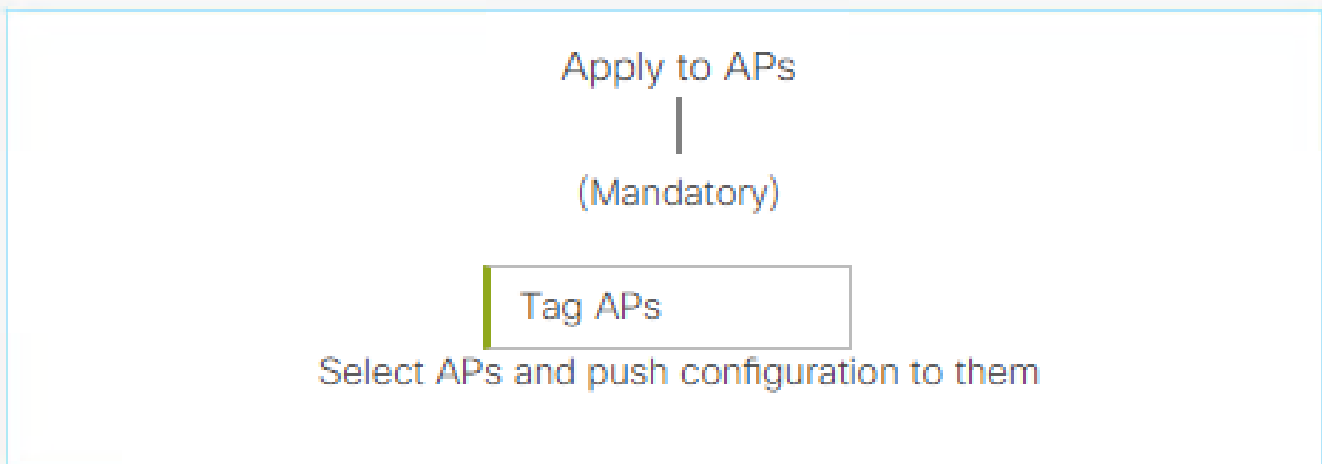
## Wireless Setup Flow Overview

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



### DEPLOY PHASE



### TERMINOLOGY

Tag

WLAN Policy, Policy Profile

Site Policy - AP Profile, Site Profile

Radio Policy - Radio Characteristics

### ACTIONS



Go to List View



Create New



가 태그되었는지 정의할 수 있음), 우선순위(숫자가 낮을수록 우선순위가 높음) 및 필요한 태그를 정의합니다.

### Associate Tags to AP ✕

Rule Name*	Guest-APs	Policy Tag Name	EWA-Tag <span>✕</span> ▼
AP name regex*	C9117-.*	Site Tag Name	Search or Select ▼
Active	YES <input checked="" type="checkbox"/>	RF Tag Name	Search or Select ▼
Priority*	1		

↶ Cancel 📄 Apply to Device

다음을 확인합니다.

구성이 올바르게 작동하는지 확인하려면 이 섹션을 활용하십시오:

```
<#root>
```

```
9800#
```

```
show running-config wlan
```

```
9800#
```

```
show running-config aaa
```

```
9800#
```

```
show aaa servers
```

```
9800#
```

```
show ap tag summary
```

```
9800#
```

```
show ap name <ap-name> config general
```

```
9800#
```

```
show ap name <ap-name> tag detail
```

```
9800#
```

```
show wlan [summary | id | name | all]
```

```
9800#
```

```
show wireless tag policy detailed <policy-tag name>
```

9800#

show wireless profile policy detailed <policy-profile name>

show ip http server status(show ip http server 상태)를 사용하여 http 서버 상태 및 가용성을 확인합니다.

<#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

다음 명령을 사용하여 클라이언트 세션에 대한 ACL plumb를 확인합니다.

<#root>

9800#

show platform software wireless-client chassis active R0 mac-address <Client mac in aaaa.bbbb.cccc format>

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 0xf1

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 0x000000a

0x0000001a 1

9800#

show platform software cgacl chassis active F0 group-idx <group index> acl

Ac1 ID Ac1 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
```

## 문제 해결

### 항상 추적

WLC 9800은 ALWAYS-ON 추적 기능을 제공합니다. 이렇게 하면 모든 클라이언트 연결 관련 오류, 경고 및 알림 수준 메시지가 지속적으로 로깅되며, 사고 또는 장애 발생 후 상황에 대한 로그를 볼 수 있습니다.



참고: 생성된 로그의 볼륨을 기반으로 몇 시간에서 며칠로 돌아갈 수 있습니다.

기본적으로 9800 WLC가 수집한 추적을 보려면 SSH/텔넷을 통해 9800 WLC에 연결하고 다음 단계를 읽을 수 있습니다(세션을 텍스트 파일에 로깅해야 함).

1단계. 문제가 발생했을 때까지의 시간에 로그를 추적할 수 있도록 컨트롤러 현재 시간을 확인합니다.

```
<#root>
```

```
9800#
```

```
show clock
```

2단계. 시스템 컨피그레이션에 따라 컨트롤러 버퍼 또는 외부 syslog에서 syslog를 수집합니다. 이렇게 하면 시스템 상태 및 오류가 있는 경우 이를 빠르게 확인할 수 있습니다.

```
<#root>
```

```
9800#
```

```
show logging
```

3단계. 디버그 조건이 활성화되었는지 확인합니다.

```
<#root>
```

```
9800#
```


```
show debugging
```

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

IOSXE Packet Tracing Configs:

Packet Infra debugs:

Ip Address	Port
----- -----	

 참고: 조건을 나열하면, 활성화된 조건(mac 주소, IP 주소 등)이 발생하는 모든 프로세스의 디버그 레벨에 추적이 로깅됨을 의미합니다. 이로 인해 로그의 볼륨이 증가합니다. 따라서 능동적으로 디버깅하지 않을 때는 모든 조건을 지우는 것이 좋습니다.

4단계. 테스트 중인 mac 주소가 3단계의 조건으로 나열되지 않았다는 가정. 특정 MAC 주소에 대한 always-on 알림 레벨 추적을 수집합니다.

<#root>

9800#

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

세션의 콘텐츠를 표시하거나 파일을 외부 TFTP 서버에 복사할 수 있습니다.

<#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>
```

## 조건부 디버깅 및 무선 활성화 추적

상시 추적 기능에서 조사 중인 문제의 트리거를 확인하기 위해 충분한 정보를 제공하지 않는 경우, 조건부 디버깅을 활성화하고 RA(Radio Active) 추적을 캡처할 수 있습니다. 그러면 지정된 조건(이 경우 클라이언트 mac 주소)과 상호 작용하는 모든 프로세스에 대한 디버그 레벨 추적이 제공됩니다. 조건부 디버깅을 활성화하려면 다음 단계를 읽으십시오.

1단계. 활성화된 디버그 조건이 없는지 확인합니다.

<#root>

9800#

```
clear platform condition all
```

2단계. 모니터링할 무선 클라이언트 mac 주소에 대한 디버그 조건을 활성화합니다.


이 명령은 30분(1,800초) 동안 제공된 MAC 주소를 모니터링하기 시작합니다. 선택적으로 이 시간을 최대 2,085,978,494초까지 늘릴 수 있습니다.

```
<#root>
```


```
9800#
```

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

---

 참고: 한 번에 둘 이상의 클라이언트를 모니터링하려면 mac 주소당 debug wireless mac 명령을 실행합니다.

---

 참고: 모든 로그가 나중에 볼 수 있도록 내부적으로 버퍼링되므로 무선 클라이언트 작업이 터미널 세션에 표시되지 않습니다.

---

3단계. 모니터링할 문제나 동작을 재현합니다.

4단계. 기본 또는 구성된 모니터 시간이 끝나기 전에 문제가 재현되는 경우 디버그를 중지합니다.

```
<#root>
```

```
9800#
```

```
no debug wireless mac <aaaa.bbbb.cccc>
```

모니터링 시간이 경과하거나 무선 디버그가 중단되면 9800 WLC는 다음과 같은 이름의 로컬 파일을 생성합니다.

```
ra_trace_MAC_aaaabbbbcccc_HHMMSS.XXX_timezone_DayWeek_Month_Day_year.log
```

5단계. MAC 주소 활동의 파일을 수집합니다. RA 추적 .log를 외부 서버에 복사하거나 출력을 화면에 직접 표시할 수 있습니다.

RA 추적 파일의 이름을 확인합니다.

```
<#root>
```

```
9800#
```

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

파일을 외부 서버에 복사:

```
<#root>
```

```
9800#
```

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

콘텐츠 표시:

```
<#root>
```

```
9800#
```

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


6단계. 근본 원인이 아직 명확하지 않은 경우 디버그 레벨 로그를 더 자세히 보여주는 내부 로그를 수집합니다. 이 명령은 이미 수집되어 내부적으로 저장된 디버그 로그를 제공하므로 클라이언트를 다시 디버깅할 필요가 없습니다.

```
<#root>
```

```
9800#
```

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

---

 참고: 이 명령 출력은 모든 프로세스의 모든 로깅 레벨에 대한 추적을 반환하며 상당히 방대합니다. 이러한 추적을 구문 분석할 수 있도록 Cisco TAC에 문의하십시오.

---

```
<#root>
```

```
9800#
```

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

콘텐츠 표시:


```
<#root>
```

```
9800#
```

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

7단계. 디버그 조건을 제거합니다.

---

 참고: 트러블슈팅 세션 후 항상 디버그 조건을 제거해야 합니다.

---

## 포함된 패킷 캡처

9800 컨트롤러는 기본적으로 패킷을 스니핑할 수 있습니다. 따라서 컨트롤 플레인 패킷 처리 가시성으로 더 쉽게 문제를 해결할 수 있습니다.

1단계. 원하는 트래픽을 필터링할 ACL을 정의합니다. 웹 인증의 경우 클라이언트가 연결된 AP에서 보내고 받는 트래픽은 물론 웹 서버로의 트래픽도 허용하는 것이 좋습니다.

```
<#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
```

2단계. 모니터 캡처 매개변수를 정의합니다. 컨트롤 플레인 트래픽이 양방향으로 활성화되었는지 확인합니다. 인터페이스는 컨트롤러의 물리적 업링크를 가리킵니다.

```
<#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)

3단계. 모니터 캡처를 시작하고 문제를 재현합니다.

<#root>

9800#

monitor capture EWA start

Started capture point : EWA

4단계. 모니터 캡처를 중지하고 내보냅니다.

<#root>

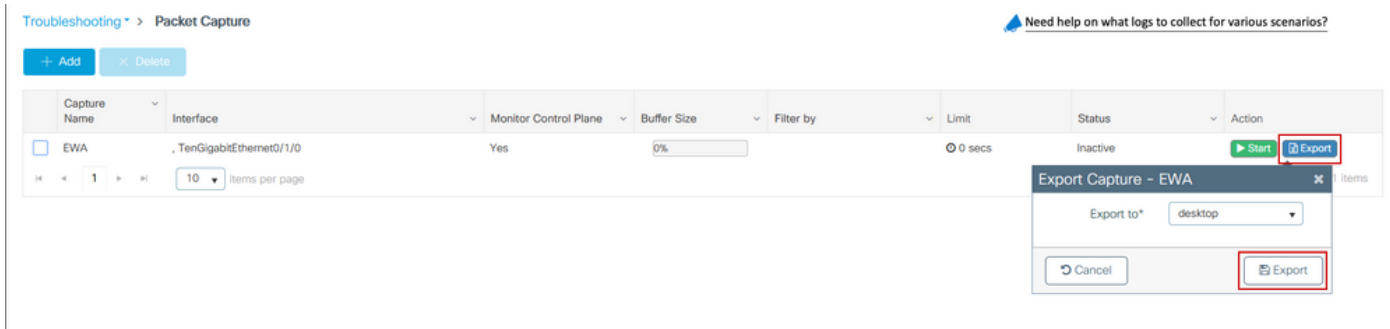
9800#

monitor capture EWA stop

Stopped capture point : EWA

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

또는 GUI에서 캡처를 다운로드하고 Troubleshooting(트러블슈팅) > Packet Capture(패킷 캡처)로 이동한 다음 구성된 캡처에서 Export(내보내기)를 선택합니다. HTTP를 통해 원하는 폴더로 캡처를 다운로드하려면 드롭다운 메뉴에서 Desktop(데스크톱)을 선택합니다.



## 클라이언트측 문제 해결

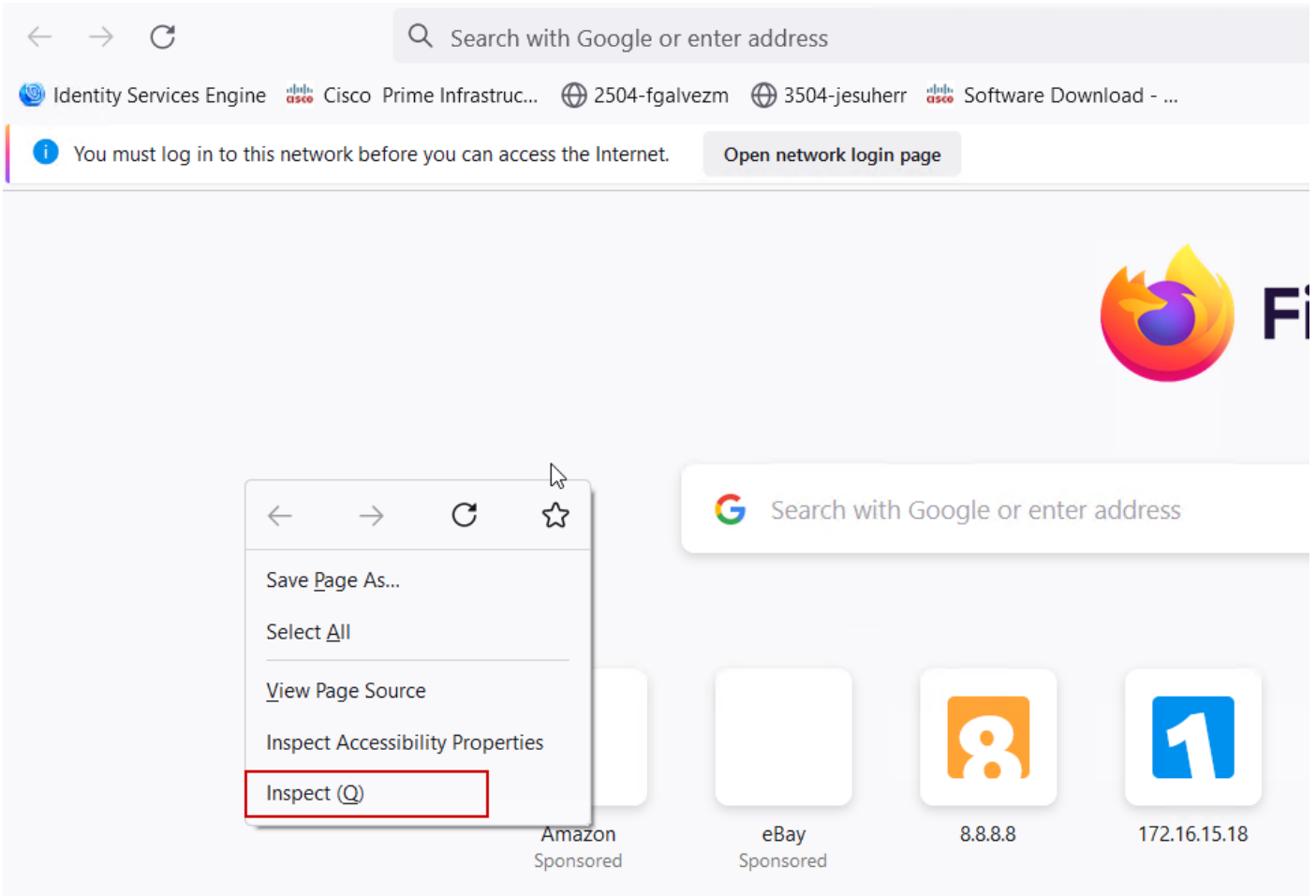
웹 인증 WLAN은 클라이언트 동작에 의존하며, 이를 기반으로 클라이언트 측 동작 지식 및 정보는 웹 인증 오동작의 근본 원인을 식별하는 데 중요합니다.

## HAR 브라우저 문제 해결

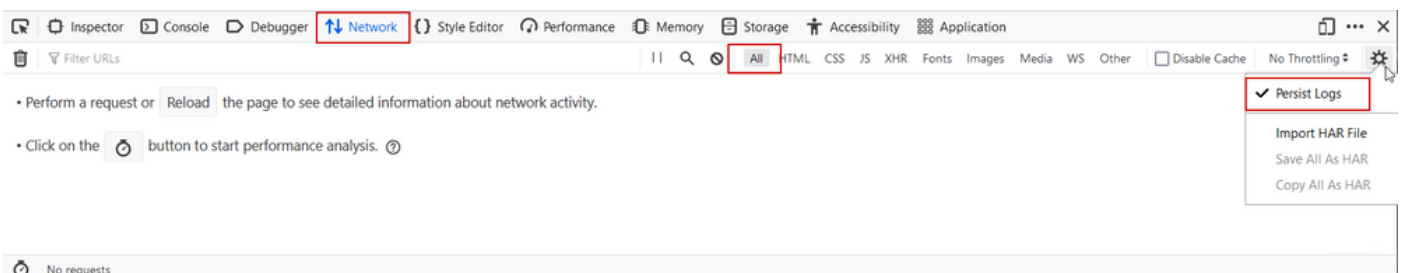
Mozilla Firefox 및 Google Chrome과 같은 많은 최신 브라우저는 웹 애플리케이션 상호 작용을 디버깅하는 콘솔 개발자 도구를 제공합니다. HAR 파일은 클라이언트-서버 상호 작용의 레코드이며 요청 및 응답 정보(헤더, 상태 코드, 매개변수 등)와 함께 HTTP 상호 작용의 타임라인을 제공합니다.

HAR 파일은 클라이언트 브라우저에서 내보내고 추가 분석을 위해 다른 브라우저에서 가져올 수 있습니다. 이 문서에서는 Mozilla Firefox에서 HAR 파일을 수집하는 방법을 간략하게 설명합니다.

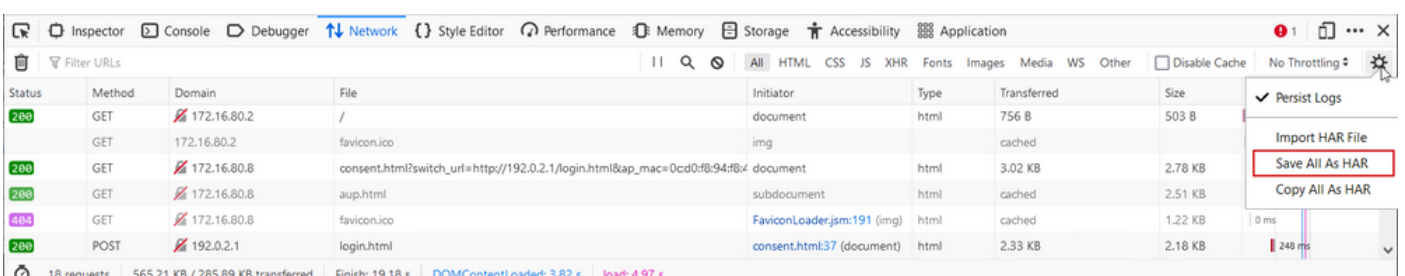
1단계. Ctrl + Shift + I로 Web Developer Tools를 열거나, 브라우저 내용 내에서 마우스 오른쪽 버튼을 클릭하고 Inspect를 선택합니다.



2단계. Network(네트워크)로 이동하여 모든 요청 유형을 캡처하려면 "All(모두)"이 선택되었는지 확인합니다. 기어 아이콘을 선택하고 Persist Logs(로그 유지) 옆에 화살표가 있는지 확인합니다. 그렇지 않으면 도메인 변경이 트리거될 때마다 로그 요청이 지워집니다.



3단계. 문제를 재현하고 브라우저가 모든 요청을 기록하는지 확인합니다. 문제가 재현되면 네트워크 로깅을 중지한 다음 톱니바퀴 아이콘에서 선택하고 Save All As HAR(모두 HAR로 저장)을 선택합니다.



## 클라이언트측 패킷 캡처

Windows 또는 MacOS와 같은 OS가 있는 무선 클라이언트는 무선 카드 어댑터에서 패킷을 스니핑할 수 있습니다. OTA(over-the-air) 패킷 캡처를 직접 대체하는 것은 아니지만 전반적인 웹 인증 흐름을 한눈에 파악할 수 있습니다.

### DNS 요청:

11868	2021-09-28	06:44:07.364395	172.16.21.153	172.16.21.7	DNS	182 53	Standard query 0x8586 A prod.detectportal.prod.cloudops.mozgcp.net
11869	2021-09-28	06:44:07.375372	172.16.21.7	172.16.21.153	DNS	195 57857	Standard query response 0x861c A detectportal.firefox.com CNWE detectportal.prod.mozaws.net CNWE prod.detectportal.prod.cloudops.mozgcp.net A 34.187.221.8
11870	2021-09-28	06:44:07.418773	172.16.21.7	172.16.21.153	DNS	118 51759	Standard query response 0x8586 A prod.detectportal.prod.cloudops.mozgcp.net A 34.187.221.82

### 리디렉션을 위한 초기 TCP 핸드셰이크 및 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

### 외부 서버와의 TCP 핸드셰이크:

11889	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
11890	2021-09-28	06:44:07.888494	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
11891	2021-09-28	06:44:07.888947	172.16.21.153	172.16.80.8	TCP	54	65209 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0

### 외부 서버에 대한 HTTP GET(중속 포털 요청):

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&mac=0cd0f897ae60&client_mac=34:23:b7:4c:6b:f7&ssid=EWa-Guest&redirect=http://www.m
11107	2021-09-28	06:44:08.522258	172.16.80.8	172.16.21.153	TCP	54	80 → 65209 [ACK] Seq=1 Ack=510 Win=66048 Len=0 [TCP segment of a reassembled PDU]
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	06:44:08.787182	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	06:44:08.834686	172.16.21.153	172.16.80.8	TCP	54	65209 → 80 [ACK] Seq=510 Ack=3095 Win=130560 Len=0

### 인증을 위해 가상 IP에 대한 HTTP POST:

12331	2021-09-28	06:44:58.644118	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	06:44:58.648688	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	06:44:58.649166	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	06:44:58.667759	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	06:44:58.672372	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	06:44:58.680599	192.0.2.1	172.16.21.153	TCP	1814	80 → 52359 [ACK] Seq=1 Ack=556 Win=64128 Len=908 [TCP segment of a reassembled PDU]
12338	2021-09-28	06:44:58.680906	192.0.2.1	172.16.21.153	TCP	1814	80 → 52359 [ACK] Seq=961 Ack=556 Win=64128 Len=908 [TCP segment of a reassembled PDU]
12339	2021-09-28	06:44:58.681125	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	06:44:58.681261	192.0.2.1	172.16.21.153	HTTP	544	HTTP/1.0 200 OK (text/html)
12341	2021-09-28	06:44:58.681423	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	06:44:58.681591	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	06:44:58.749848	172.16.21.153	192.0.2.1	TCP	54	52359 → 80 [ACK] Seq=556 Ack=2412 Win=130560 Len=0

## 성공한 시도의 예

이는 Radio Active 추적 관점에서 성공한 연결 시도의 출력입니다. 레이어 3 웹 인증 SSID에 연결하는 클라이언트의 클라이언트 세션 단계를 식별하기 위한 참조로 사용합니다.

### 802.11 인증 및 연결:

<#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.

계층 2 인증을 건너뛰었습니다.

<#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 플러그 앤:

<#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-ac1] [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-ac1] [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 학습 프로세스:

<#root>

2021/09/28 12:59:51.799515 {wncd\_x\_R0-0}{1}: [client-orch-state] [26328]: (note): MAC: 3423.874c.6bf7 Client

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 capwap  
[...]

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\_9000000b]

2021/09/28 12:59:52.810794 {wncd\_x\_R0-0}{1}: [auth-mgr-feat\_wireless] [26328]: (info): [0000.0000.0000:unknown]

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

## 레이어 3 인증 및 리디렉션 프로세스:

<#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.6bf7]

2021/09/28 12:59:55.324565 {wncd\_x\_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap\_9000000b[3423.874c.6bf7]

HTTP GET request

2021/09/28 12:59:55.324588 {wncd\_x\_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap\_9000000b[3423.874c.6bf7]

[...]

2021/09/28 13:01:29.859434 {wncd\_x\_R0-0}{1}: [webauth-httpd] [26328]: (info): capwap\_9000000b[3423.874c.6bf7]

POST rcvd when in LOGIN state

2021/09/28 13:01:29.859636 {wncd\_x\_R0-0}{1}: [webauth-ac1] [26328]: (info): capwap\_9000000b[3423.874c.6bf7]

2021/09/28 13:01:29.860335 {wncd\_x\_R0-0}{1}: [webauth-ac1] [26328]: (info): capwap\_9000000b[3423.874c.6bf7]

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

실행 상태로 전환:

<#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

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.