

# Catalyst 9800 Wireless LAN Controller의 무선 디버깅 및 로그 수집 이해

## 목차

---

### [소개](#)

#### [사전 요구 사항](#)

##### [요구 사항](#)

##### [사용되는 구성 요소](#)

#### [배경 정보](#)

##### [9800 WLC 내부의 패킷 흐름](#)

#### [컨트롤 플레인 추적](#)

##### [Syslog](#)

##### [항상 추적](#)

##### [실패 시 추적](#)

##### [조건부 디버깅 및 RadioActive 추적](#)

##### [웹 UI를 통한 방사능 흔적](#)

##### [CLI를 통한 방사성 추적](#)

##### [프로세스별 비조건부 디버깅](#)

#### [데이터 플레인 패킷 추적](#)

#### [임베디드 패킷 캡처](#)

#### [경보 LED 및 중요한 플랫폼 경보](#)

---

## 소개

이 문서에서는 Catalyst 9800 문제 해결에 사용되는 모든 Cisco IOS® XE 기능에 대해 설명하고 개요를 제공합니다.

## 사전 요구 사항

### 요구 사항

- WLC(Wireless LAN Controller)에 대한 기본 지식
- WLC 사용과 관련된 활용 사례 흐름에 대한 기본 지식

### 사용되는 구성 요소

이 문서에서는 9800-CL, 9800-L, 9800-40 및 9800-80 컨트롤러에 대해 설명합니다. 주로 17.3 Cisco IOS® XE 버전을 기반으로 합니다.

## 배경 정보

9800 WLC에서 실행되는 Cisco IOS® XE는 기본적으로 Cisco IOS® 및 데몬으로 구현된 모든 무선 프로세스가 포함된 binOS(Linux Kernel)로 구성됩니다.

모든 프로세스 데몬은 일반 용어인 CP(Control Plane) 아래에 번들로 제공되며 CAPWAP(Control and Provisioning of Access Point), 모빌리티, RRM(Radio Resource Management)을 담당합니다. 9800 WLC를 오가는 NMSP(Rogue Management, Network Mobility Service Protocol)

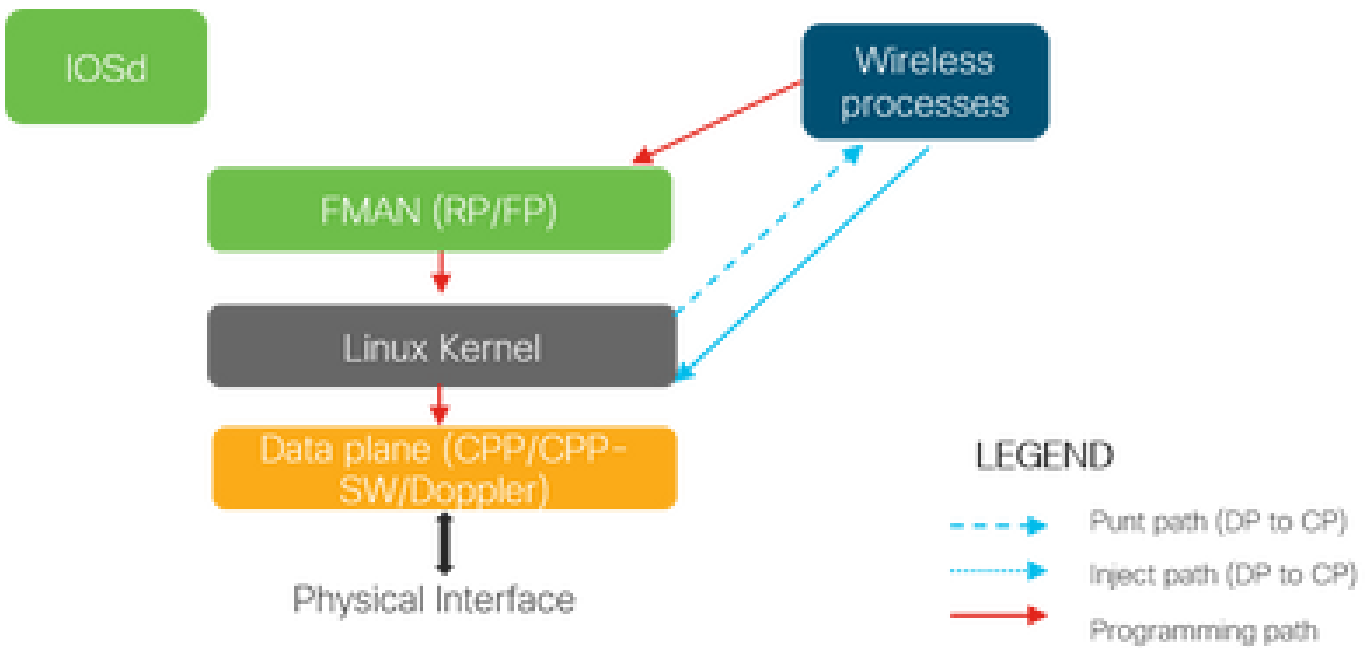
데이터 플레인(DP)은 9800 WLC에서 데이터를 전달하는 구성 요소를 나타냅니다.

9800(9800-40, 9800-80, 9800-CL, 9800-SW,9800-L)의 모든 반복에서 컨트롤 플레인(은 상당히 일반적입니다.

그러나 데이터 평면은 ASR1k와 유사한 하드웨어 QFP(Quantum Flow Processor) 콤플렉스를 사용하는 9800-40 및 9800-80에 따라 다르며, 9800-CL 및 9800-L은 Cisco CPP(Packet Processor)의 소프트웨어 구현을 사용합니다.

9800-SW는 데이터 포워딩을 위해 Catalyst 9k 시리즈 스위치의 도플러 칩셋을 이용하기만 하면 됩니다.

### 9800 WLC 내부의 패킷 흐름



패킷이 물리적 포트에서 9800 WLC로 들어갈 때 제어 트래픽으로 판별되면 해당 제어 평면 프로세스로 보내집니다.

AP 조인의 경우 AP에서 소싱한 모든 capwap 및 dtls 교환이 됩니다. 클라이언트 조인의 경우 클라이언트가 RUN 상태로 전환되기 전까지 클라이언트에서 제공된 모든 트래픽이 PUNT 경로를 따릅니다.

다양한 데몬이 수신 트래픽을 처리하면 9800 WLC에서 클라이언트에 전송할 결과 반환 트래픽 (capwap 응답, dot11, dot1x, dcp 응답)이 데이터 플레인에 다시 주입되어 물리적 포트를 통해 전송됩니다.

AP 가입, 클라이언트 가입, 모빌리티 교환, 데이터 플레인을 처리할 때 데이터 트래픽 포워딩을 처리할 수 있도록 프로그래밍해야 합니다.

이는 이미지에 표시된 프로그래밍 경로에 걸쳐 여러 컴포넌트가 순차적으로 프로그래밍되면서 발생한다.

Cisco IOS® XE는 9800 WLC로 들어오는 순간부터 처리된 트래픽이 시스템을 떠날 때까지 패킷을 추적할 수 있는 다목적 툴 세트를 제공합니다.

다음 섹션에서는 이러한 툴을 CLI(Command Line Interface)에서 호출하는 데 사용되는 명령과 함께 소개합니다.

## 컨트롤 플레인 추적

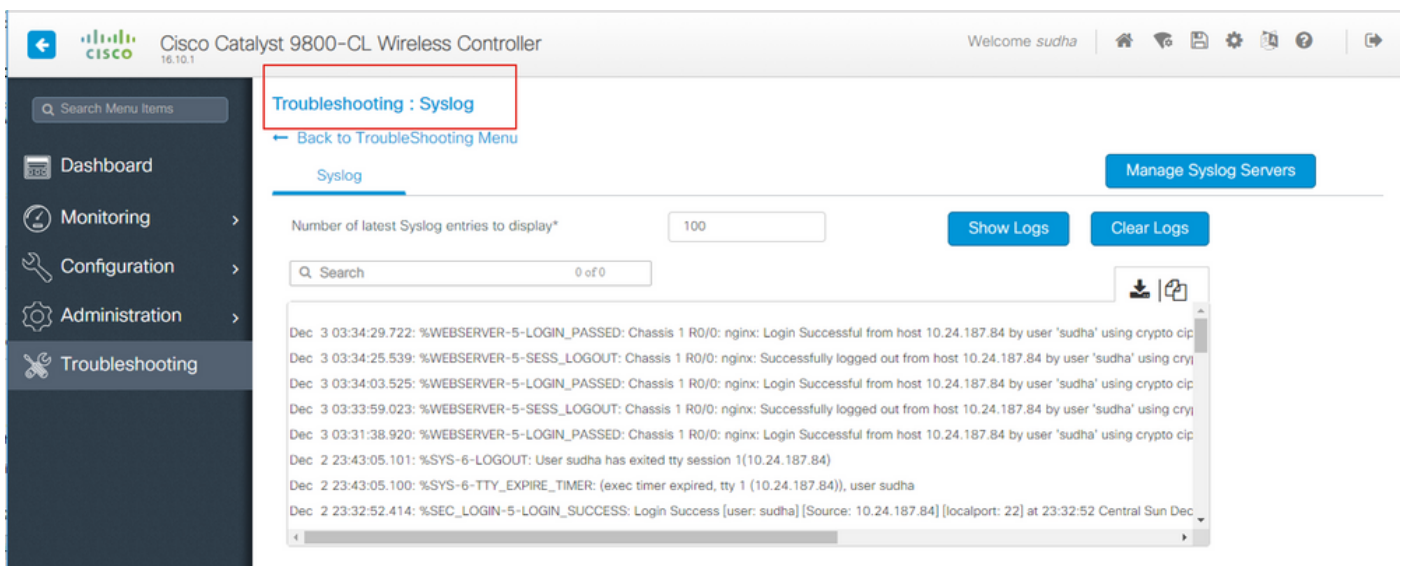
이 섹션에서는 9800 WLC를 위한 패킷이 DP에서 시작된 후 또는 9800 WLC에서 물리적 인터페이스를 전송하기 위해 DP로 제공된 응답 패킷을 주입하기 전에 제어 평면 프로세스에서 수행한 처리를 보는 데 사용할 수 있는 명령 및 툴에 대해 설명합니다

### Syslog

9800 WLC에서 생성된 로그는 시스템의 일반적인 상태를 확인하는 첫 번째 방법입니다.

CPU, 메모리, 버퍼와 같은 시스템 리소스에 대해 사전 정의된 임계값을 위반하는 경우 로그에 보고됩니다.

또한 하위 시스템에서 발생한 모든 오류는 로그에 기록됩니다. 로그를 보려면 Troubleshooting(문제 해결) > Syslog로 이동합니다




또는 CLI 명령을 실행합니다.

```
# show logging
```

이 출력은 일반 로그와 일부 무선 세부 로그를 보여줍니다. 그러나 레거시 Cisco IOS®와는 달리 무선 디버깅은 일반적으로 이 로깅 출력에 적용되지 않습니다.

---

 참고: WLC9800이 이러한 로그를 외부 syslog 서버로 리디렉션하도록 구성된 경우 외부 syslog 서버에서도 로그를 확인해야 합니다.

---

## 항상 추적

WLC9800의 모든 컨트롤 플레인 프로세스는 Notice의 로깅 수준에서 자체 전용 버퍼에 지속적으로 로깅됩니다. 이를 Always-On 추적이라고 합니다.

이는 실패 조건을 재현하도록 강제하지 않고 발생한 실패에 대한 상황 데이터를 얻을 수 있는 고유한 기능입니다.

예를 들어 AireOS에 익숙한 경우 클라이언트 연결 문제 해결을 위해 디버깅을 활성화하고 클라이언트 연결 문제 상태를 재현하여 근본 원인을 파악해야 합니다.

Always-on 추적을 사용하면 이미 캡처된 추적을 다시 살펴보고 일반적인 근본 원인인지 확인할 수 있습니다. 생성된 로그의 양에 따라, 우리는 몇 시간에서 며칠을 되돌아볼 수 있다.

이제 각 개별 프로세스마다 추적이 로깅되지만 클라이언트 mac 또는 AP mac, AP ip 주소와 같은 특정 관심 컨텍스트에 대해 전체적으로 추적이 가능합니다. 이렇게 하려면 명령을 실행합니다

```
# show logging profile wireless filter mac to-file bootflash:
```

기본적으로 이 명령은 로그를 생성 및 디코딩하기 위해 10분 단위로만 되돌아갑니다. 다음으로 시간을 더 거슬러 올라가도록 선택할 수 있습니다.


```
# show logging profile wireless start last
```

```
[minutes|hours|days] filter mac to-file bootflash:
```

프로세스별 로그를 보려면 명령을 실행합니다

```
# show logging process to-file bootflash:
```

---

 참고: 이러한 CLI에는 모듈, 로깅 레벨, 시작 타임스탬프 등을 비롯한 여러 필터링 옵션이 있습니다. 이러한 옵션을 보고 탐색하려면 명령을 실행합니다

---

```
# show logging profile wireless ?  
# show logging process ?
```

## 실패 시 추적

일반적으로 알려진 장애 상태의 빠른 스냅샷을 얻으려면 장애 시 추적 기능을 사용할 수 있습니다. 이는 사전 정의된 장애 조건과 일치시키기 위해 지정된 시점에서 시스템의 모든 추적을 구문 분석하고, 요약 보기와 통계를 제공합니다.

요약 보기를 가져오려면 명령을 실행합니다

```
# show logging profile wireless trace-on-failure summary
```

미리 정의된 실패 조건 및 이러한 조건에 해당하는 통계를 보려면 명령을 실행합니다

```
# show wireless stats trace-on-failure
```

실패를 알게 되면 실패의 컨텍스트에 특정한 추적을 수집하려면 명령을 실행합니다

```
# show logging profile wireless filter uuid to-file bootflash:tof-FILENAME.txt
```

이러한 파일은 터미널 세션에서 보거나 명령을 사용하여 오프라인 분석을 위해 내보낼 수 있습니다


```
# more bootflash:tof-FILENAME.txt  
OR  
# copy bootflash:tof-FILENAME.txt { tftp: | ftp: | scp: | https: } tof-FILENAME.txt
```


## 조건부 디버깅 및 RadioActive 추적

조건부 디버깅을 사용하면 원하는 조건의 특정 기능에 대한 디버그 레벨 로깅을 활성화할 수 있습니다.

RadioActive 추적은 관심 상태의 스레드와 프로세스 간에 디버그 정보를 조건부로 인쇄하는 기능을 추가하여 한 단계 더 나아갑니다.

즉, 기본 아키텍처가 완전히 추상화됩니다.

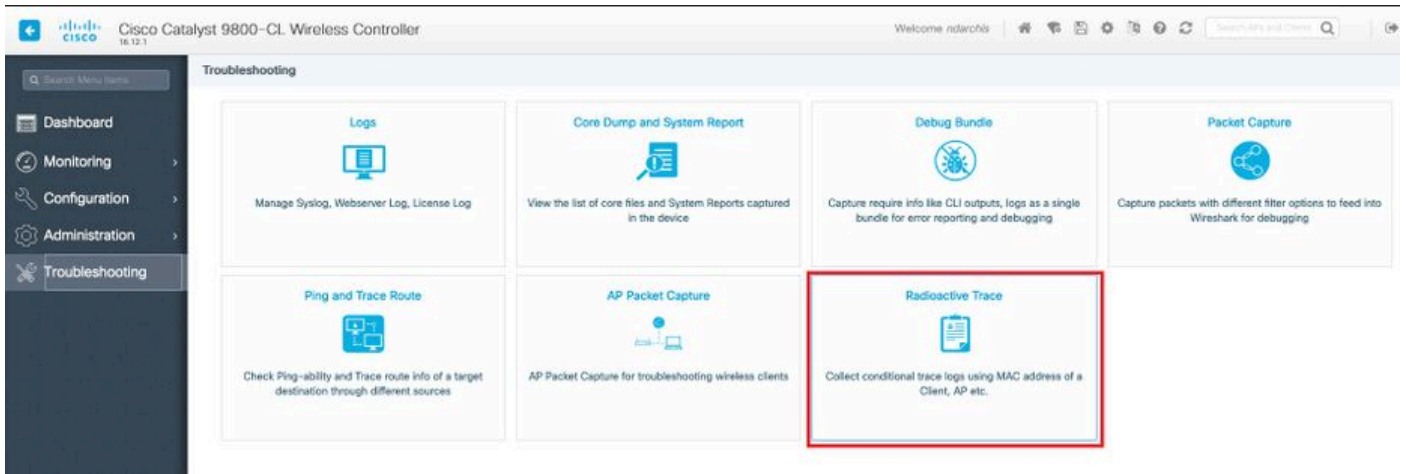
 참고: 16.12에서는 AP 라디오 및 이더넷 mac 주소로 AP 가입, 클라이언트 mac 주소로 클라이언트 가입, 모빌리티 피어 IP 및 CMX 연결과 관련된 모빌리티 문제를 관심 조건으로 트러블 슈팅하기 위해 방사성 추적이 구현됩니다.

 참고: MAC 주소와 IP 주소를 조건으로 사용하면 서로 다른 프로세스에서 동일한 네트워크 엔티티(AP 또는 클라이언트 또는 모빌리티 피어)에 대해 서로 다른 식별자를 인식하므로 서로 다른 출력이 제공됩니다.

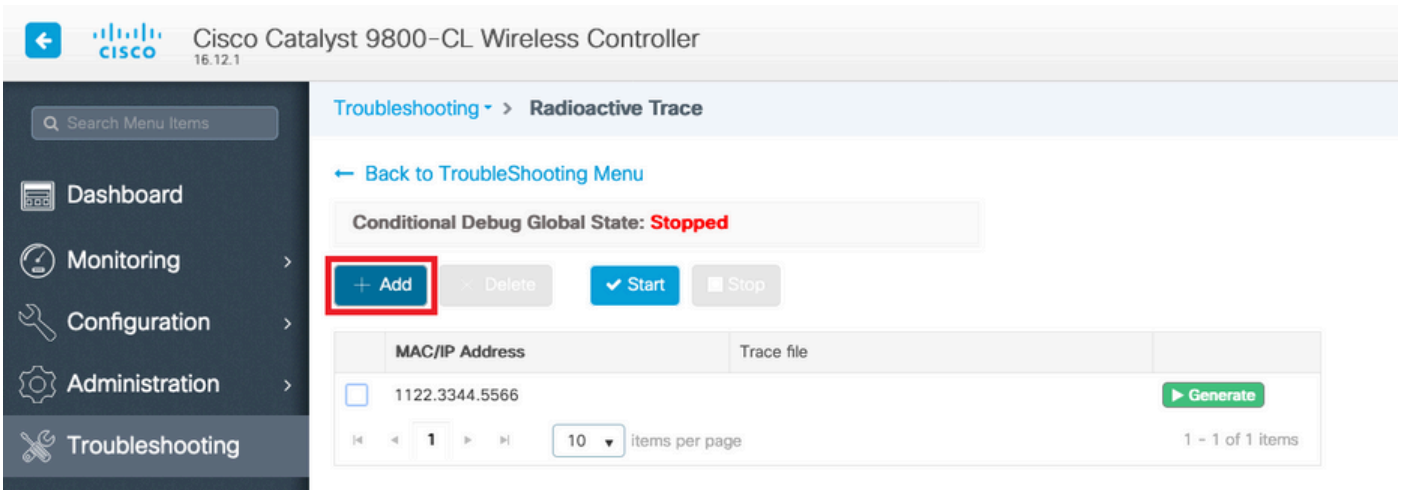
클라이언트 연결에서는 문제 해결을 위한 예로 클라이언트 mac에 대해 조건부 디버깅이 실행되어 제어 평면에서 엔드 투 엔드 보기를 제공합니다.

### 웹 UI를 통한 방사능 흔적

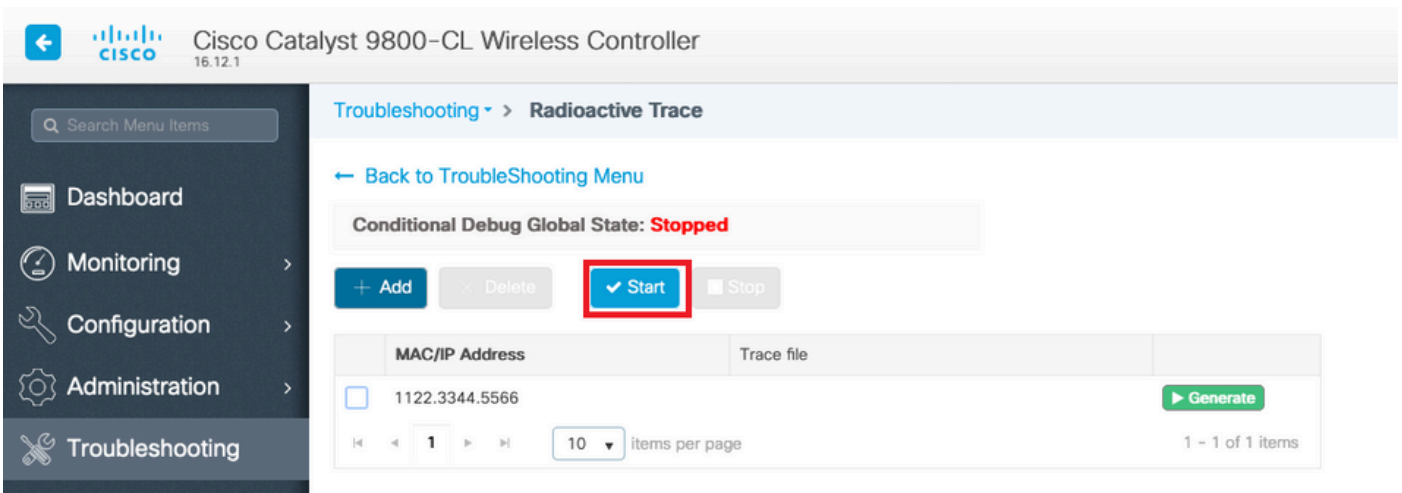
Troubleshooting(문제 해결) 페이지 메뉴로 이동하여 Radioactive Tracing(방사능 추적)을 선택합니다



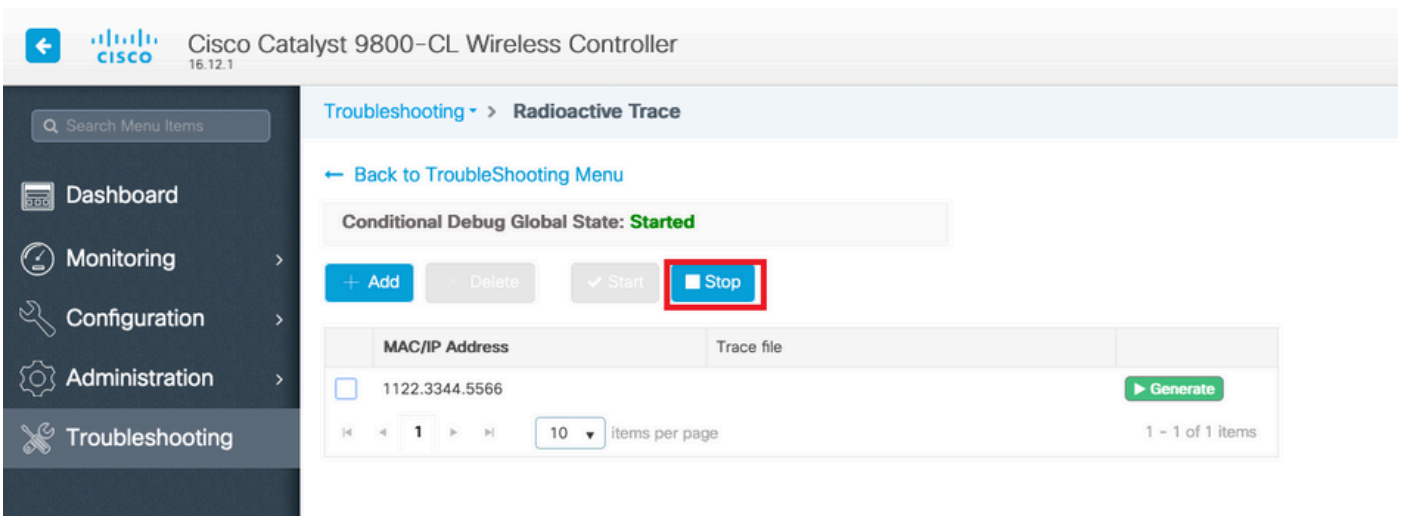
Add(추가)를 클릭하고 문제를 해결할 클라이언트 또는 AP mac 주소를 입력합니다. 16.12부터 GUI를 통해 mac 주소만 추가할 수 있습니다. CLI를 통한 IP 주소 추가 가능



추적할 여러 mac 주소를 추가할 수 있습니다. 방사성 추적을 시작할 준비가 되면 시작을 클릭합니다.



일단 시작되면, 추적 맥 주소와 관련된 제어 평면 처리에 대한 디버그 로깅이 디스크에 기록된다. 문제를 재현한 후 문제를 해결하려면 [중지]를 클릭하십시오.



디버깅된 각 mac 주소에 대해 Generate(생성)를 클릭하여 해당 mac 주소와 관련된 모든 로그를 취합하는 로그 파일을 생성할 수 있습니다.

Cisco Catalyst 9800-CL Wireless Controller  
16.12.1

Troubleshooting > Radioactive Trace

← Back to Troubleshooting Menu

Conditional Debug Global State: **Stopped**

+ Add    - Delete    ✓ Start    ■ Stop

MAC/IP Address	Trace file
<input type="checkbox"/> 1122.3344.5566	<input type="button" value="Generate"/>

10 items per page    1 - 1 of 1 items

취합된 로그 파일을 저장할 기간을 선택하고 Apply to Device(디바이스에 적용)를 클릭합니다.

### Enter time interval ✕

Generate logs for last

- 10 minutes
- 30 minutes
- 1 hour
- since last boot
- 


이제 파일 이름 옆에 있는 작은 아이콘을 클릭하여 파일을 다운로드할 수 있습니다. 이 파일은 컨트롤러의 부트플래시 드라이브에 있으며 CLI를 통해 즉시 복사할 수도 있습니다.



← Back to TroubleShooting Menu

Conditional Debug Global State: **Stopped**

+ Add    × Delete    ✓ Start    ■ Stop

	MAC/IP Address	Trace file	
<input type="checkbox"/>	1122.3344.5566	debugTrace_1122.3344.5566.txt 	<b>▶ Generate</b>

◀ 1 ▶ 10 items per page      1 - 1 of 1 items

### CLI를 통한 방사성 추적

조건부 디버깅을 활성화하려면 명령을 실행합니다

```
# debug wireless {mac | ip} {aaaa.bbbb.cccc | x.x.x.x } {monitor-time} {N seconds}
```

현재 활성화된 조건을 보려면 명령을 실행합니다


```
# show debugging
```

이러한 디버그는 터미널 세션의 출력을 인쇄하지 않지만 디버그 출력 파일을 플래시에 저장하여 이후에 검색하고 분석합니다. 파일은 명명 규칙 ra\_trace와 함께 저장됩니다\_\*

예를 들어 mac 주소 aaaa.bbbb.ccc의 경우 생성되는 파일 이름은 ra\_trace\_MAC\_aaaabbbbccc\_HHMMSS.XXX\_timezone\_DayWeek\_Month\_Day\_year.log입니다

한 가지 장점은 동일한 명령을 사용하여 AP 가입 문제(입력 AP 라디오 mac 및 이더넷 mac), 클라이언트 연결 문제(입력 클라이언트 mac), 모빌리티 터널 문제(입력 피어 ip), 클라이언트 로밍 문제(입력 클라이언트 mac)를 해결할 수 있다는 것입니다.

즉, debug capwap, debug client, debug mobility 등과 같은 여러 명령을 기억할 필요가 없습니다.

 참고: 디버그 무선에서는 FTP 서버를 가리키고 internal 키워드를 사용하여 더 자세한 로깅 정보를 실행할 수도 있습니다. 일부 문제가 해결되기 때문에 지금은 이러한 옵션을 권장하지 않습니다.

터미널 세션에서 출력 파일을 디버깅하려면 명령을 실행합니다

```
# more bootflash:ra_trace_MAC_*.log
```

디버그 출력을 오프라인 분석을 위해 외부 서버로 리디렉션하려면 명령을 실행합니다

```
# copy bootflash:ra_trace_MAC_*.log ftp://username:password@FTPSERVERIP/path/RATRACE_FILENAME.txt
```


같은 디버그 로그 수준에 대해 훨씬 더 자세한 보기가 있습니다. 이 자세한 보기를 보려면 명령을 실행하십시오

```
# show logging profile wireless internal filter mac to-file
```

특정 컨텍스트에 대해 또는 구성된 모니터 시간 또는 기본 모니터 시간이 시작되기 전에 디버깅을 비활성화하려면 명령을 실행합니다.

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

---

 주의: 조건부 디버깅은 디버그 레벨 로깅을 활성화하므로 생성된 로그의 볼륨이 증가합니다. 이 작업을 계속 실행하면 로그를 볼 수 있는 시간이 줄어듭니다. 따라서 트러블슈팅 세션이 끝나면 항상 디버깅을 비활성화하는 것이 좋습니다.

---

모든 디버깅을 비활성화하려면 다음 명령을 실행합니다

```
# clear platform condition all  
# undebg all
```

## 프로세스별 비조건부 디버깅

방사성 추적에 대해 구현되지 않은 사용 사례 및 프로세스의 경우 디버그 레벨 추적을 가져올 수 있습니다. 특정 프로세스에 대한 디버그 수준을 설정하려면

```
# set platform software trace <PROCESS_NAME> wireless chassis active R0 { module_name | all-modules }
```

다양한 모듈의 추적 수준을 확인하려면 명령을 실행합니다

```
# show platform software trace level <PROCESS_NAME> chassis active R0
```

수집된 추적을 보려면 명령을 실행합니다

```
# show logging process to-file
```

## 데이터 플레인 패킷 추적

패킷이 처음 9800 WLC에 들어갈 때, 트래픽이 컨트롤 플레인인지 데이터 플레인인지를 식별하기 위해 데이터 플레인에서 일부 처리가 발생합니다.

패킷 추적 기능은 데이터 플레인에서 수행된 이 Cisco IOS® XE 처리 및 패킷을 펀트, 전달, 삭제 또는 사용할지 여부를 결정하는 방법에 대한 자세한 보기를 제공합니다.

WLC 9800의 이 기능은 ASR1k의 구현과 정확히 동일합니다.


9800 WLC의 Packet Tracer는 ASR1K와 동일한 세 가지 검사 레벨을 제공합니다.

- Statistics(통계) - 네트워크 프로세서를 들어오고 나가는 패킷 수를 제공합니다
- 요약-
  - 이는 특정 관심 조건과 일치하는 유한한 수의 패킷에 대해 수집됩니다.
  - 요약 출력은 인그레스 및 이그레스 인터페이스, 데이터 플레인에서 수행한 조회 결정을 나타내며 punt, drop 및 inject 패킷(있는 경우)도 추적합니다.
  - 이 출력은 데이터 플레인 처리에 대한 간결한 뷰를 제공합니다
- Path Data(경로 데이터) - DP 패킷 처리에 대한 가장 자세한 보기를 제공합니다. 한정된 수의 패킷에 대해 수집되며, DP 패킷과 컨트롤 플레인 디버깅, 타임스탬프 및 기능 특정 경로 추적 데이터를 연계하는 데 사용할 수 있는 조건부 디버깅 ID를 포함합니다. 이 세부 보기에는 두 가지 선택 기능이 있습니다
  - 패킷 복사를 사용하면 패킷의 다양한 레이어(레이어 2, 레이어 3 및 레이어 4)에서 인그레스 및 이그레스 패킷을 복사할 수 있습니다
  - FIA(Feature Invocation Array)는 데이터 플레인에 의해 패킷에서 실행되는 기능의 순차적 목록입니다. 이러한 기능은 WLC 9800의 기본 및 사용자 지원 컨피그레이션에서 파생됩니다

기능 및 하위 옵션에 대한 자세한 설명은 Cisco IOS [XE Datapath 패킷 추적 기능을 참조하십시오](#)

AP 가입, 클라이언트 연결 등과 같은 무선 워크플로의 경우 양방향으로 업링크를 추적합니다

---


 주의: dataplane packet-tracer는 외부 CAPWAP 헤더만 구문 분석합니다. 따라서 무선 클라이언트 mac과 같은 조건은 유용한 출력을 산출하지 않습니다.

---

1단계. 관심 조건을 정의합니다.

```
# debug platform condition { interface | mac | ingress | egress | both | ipv4 | ipv6 | mpls | match }
```

---

 경고: 디버그 플랫폼 조건 기능 및 디버그 플랫폼 조건 mac aaaa.bbb.ccc 명령 모두 컨트롤 플레인 패킷 추적을 위한 것이며 데이터 플레인 패킷 추적을 반환하지 않습니다.

---

2단계. 현재 활성화된 조건을 보려면 명령을 실행합니다

```
# show platform conditions
```

3단계. 한정된 수의 패킷에 대해 packet-tracer를 활성화합니다. 이 패킷 번호는 16~8192 범위에서 2의 거듭제곱으로 정의됩니다. 기본적으로 요약 및 기능 데이터가 모두 캡처됩니다. 선택적으로, 요약 전용 하위 옵션을 사용하는 경우 요약 보기만 가져오도록 선택할 수 있습니다. 패킷 크기(바이트)를 정의하고, 패킷 크기를 추적하거나, 패킷을 삽입하거나, 삭제하는 등의 하위 옵션을 사용할 수도 있습니다.

```
# debug platform packet-tracer packet <packet-number> {fia-trace}
```

4단계. (선택 사항) 추적되는 대로 패킷을 복사하여 덤프할 수 있습니다

```
# debug platform packet-trace copy packet both size 2048 { 12 | 13 | 14 }
```

5단계. 조건부 디버깅을 활성화합니다.

```
# debug platform condition start
```

6단계. 패킷 추적이 출력을 수집하는지 확인하려면 통계를 확인합니다

```
# show platform packet-trace statistics
```

7단계. packet-trace의 출력을 보려면 명령을 실행합니다

```
# show platform packet-tracer summary
```

8단계. (선택 사항) Cisco TAC에서 오프라인 분석을 위해 패킷 덤프를 내보낼 수 있습니다

```
# show platform packet-trace packet all | redirect { bootflash: | tftp: | ftp: } pacrac.txt
```

## 임베디드 패킷 캡처

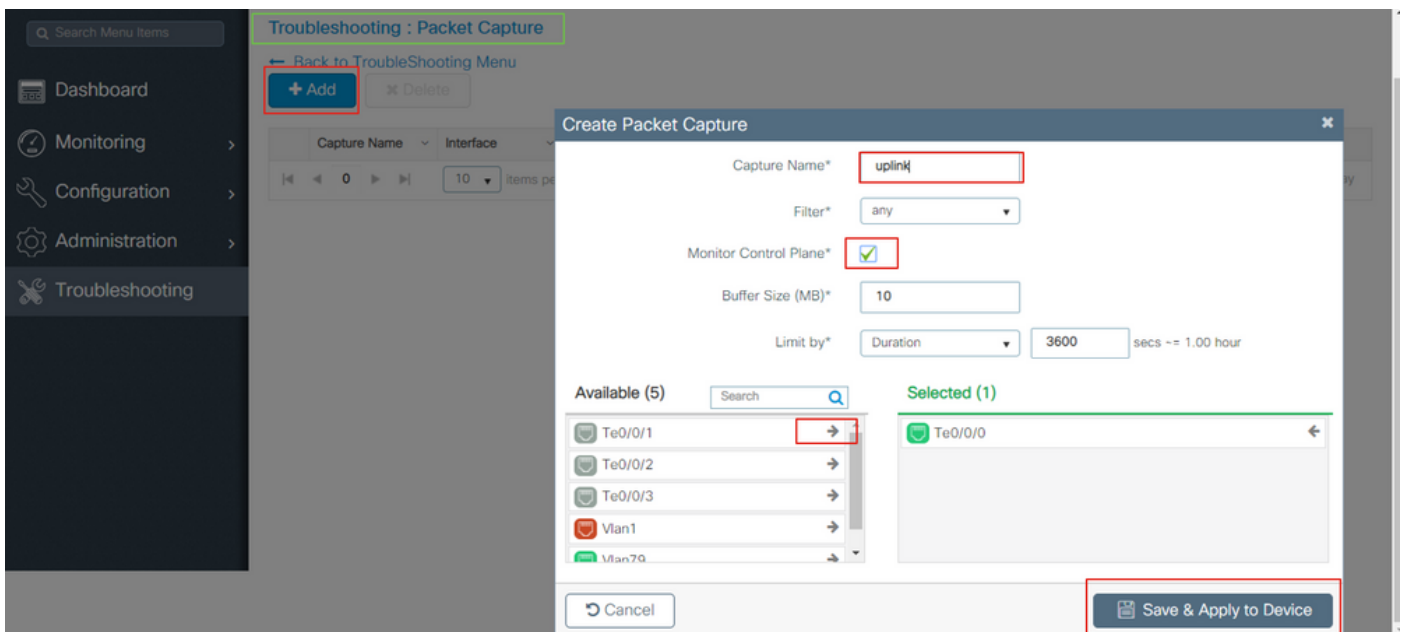
EPC(Embedded Packet Capture)는 Catalyst 9800 WLC를 통과하도록 지정된 패킷으로 볼 수 있는 패킷 캡처 기능입니다. 이러한 캡처는 Wireshark를 사용하여 오프라인 분석을 위해 내보낼 수 있습니다.

기능에 대한 자세한 내용은 [EPC 컨피그레이션 가이드를 참조하십시오](#)

9800 WLC는 AireOS에 비해 업링크 스위치의 패킷 캡처 및 트래픽 미러링 기능에 의존하는 대신 박스 자체에서 pcap 캡처를 지원합니다.

9800에서는 CLI(Command Line Interface) 및 GUI(Graphical User Interface)에서 이 캡처를 설정할 수 있습니다.

GUI를 통해 구성하려면 Troubleshooting(문제 해결) > Packet Capture(패킷 캡처) > +Add(추가)로 이동합니다



1단계. 패킷 캡처의 이름을 정의합니다. 최대 8자까지 허용됩니다.

2단계. 필터 정의(있는 경우)

3단계. 시스템 CPU로 보내지고 데이터 플레인으로 다시 주입되는 트래픽을 보려면 Monitor Control

Traffic(제어 트래픽 모니터링) 확인란을 선택합니다

4단계. 버퍼 크기를 정의합니다. 최대 100MB가 허용됩니다

5단계. 원하는 대로 1~1000000초 범위를 허용하는 기간 또는 1~100000 패킷 범위를 허용하는 패킷 수로 제한을 정의합니다

6단계. 왼쪽 열의 인터페이스 목록에서 인터페이스를 선택하고 화살표를 선택하여 오른쪽 열로 이동합니다

7단계. 저장 및 장치에 적용

8단계. 캡처를 시작하려면 시작을 선택합니다.

9단계. 캡처가 정의된 한도까지 실행되도록 할 수 있습니다. 캡처를 수동으로 중지하려면 중지를 선택합니다.

10단계. 중지되면 Export(내보내기) 버튼을 클릭하여 https 또는 TFTP 서버나 FTP 서버 또는 로컬 시스템 하드 디스크나 플래시를 통해 로컬 데스크톱에 캡처 파일(.pcap)을 다운로드할 수 있습니다.

Capture Name	Interface	Monitor Control Plane	Buffer Size	Filter by	Limit	Status	Action
<input type="checkbox"/> uplink	TenGigabitEthernet0/0/0	Yes	0%	any	0 secs	Inactive	<a href="#">Start</a>

참고: CLI는 Limit by와 같은 좀 더 세분화된 옵션을 제공합니다. GUI는 일반적인 활용 사례의 경우 패킷을 캡처하는 데 충분합니다.

CLI를 통해 구성하려면

모니터 캡처를 생성합니다.

```
monitor capture uplink interface <uplink_of_the_9800> both
```

필터를 연결합니다. 필터를 인라인으로 지정하거나 ACL 또는 클래스 맵을 참조할 수 있습니다.

이 예에서는 9800의 2개 ip 주소와 다른 WLC 5520 간의 트래픽을 확인하기 위한 ACL입니다. 일반적인 모빌리티 문제 해결 시나리오:

```
conf t
```

```
ip access-list extended mobilitywlc
```

```
permit ip host <5520_ip_address> host <9800_ip_address>
  permit ip host <9800_ip_address> host <5520_ip_address>
end

monitor capture uplink access-list mobilitywlcs
```

캡처가 순환 버퍼에서 실행되도록 하려면 문제를 알아차린 다음 캡처를 중지하고 저장할 수 있는 시간을 제공합니다.

예를 들어 50MB 버퍼로 설정하는 경우 9800에는 최대 50MB의 디스크가 필요하며, 문제의 발생에 대비하여 몇 분간의 데이터를 캡처하는 데 상당히 큰 용량입니다.

```
monitor capture uplink buffer circular size 50
```

캡처를 시작합니다. GUI 또는 CLI에서 이를 수행할 수 있습니다.

```
monitor capture uplink start
```

이제 캡처가 활성 상태입니다.

필요한 데이터를 수집하도록 허용합니다.

캡처를 중지합니다. GUI 또는 CLI를 통해 수행할 수 있습니다.

```
monitor capture uplink stop
```

GUI > Troubleshooting(문제 해결) > Packet Capture(패킷 캡처) > Export(내보내기)에서 캡처를 검색할 수 있습니다.

또는 CLI에서 서버에 업로드합니다. ftp를 통한 예:

```
monitor capture uplink export ftp://x.x.x.x/MobilityCAP.pcap
```

필요한 데이터가 수집되면 캡처를 제거합니다.

```
no monitor capture uplink
```

# 경보 LED 및 중요한 플랫폼 경보

모든 9800 어플라이언스(9800-L, 9800-40 및 9800-80)의 전면 패널에는 ALM LED가 있습니다. 해당 LED가 빨간색으로 켜지면 플랫폼에 중대한 경보가 올린다는 의미입니다.

show facility-alarm status 명령을 사용하여 LED가 빨간색으로 바뀌는 알람을 확인할 수 있습니다

```
WLC#show facility-alarm status
```

```
System Totals Critical: 2 Major: 0 Minor: 0
```

Source	Time	Severity	Description [Index]
-----	-----	-----	-----
TenGigabitEthernet0/1/0	Jul 26 2019 15:14:04	CRITICAL	Physical Port Link Down [1]
TenGigabitEthernet0/1/1	Jul 26 2019 15:14:04	CRITICAL	Physical Port Link Down [1]



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.