

# 9800 WLC의 메모리 누수 문제 이해

## 목차

---

[소개](#)

[메모리 누수](#)

[Syslog](#)

[9800 WLC에서 메모리 누수 문제가 발생했음을 확인하는 방법](#)

[IOS 프로세스의 메모리 누수 문제 해결](#)

[WLC의 기본 로그](#)

[프로세서 메모리 풀용](#)

[IO 메모리 풀용](#)

[Polaris/Platform 수준에서 메모리 누수 문제 해결](#)

[권장 사항](#)

---

## 소개

이 문서에서는 Cisco Catalyst 9800 WLC(Wireless LAN Controller)와 관련된 메모리 누수에 대해 설명합니다.

## 메모리 누수

프로그램이나 프로세스가 임시용으로 메모리를 할당한 후 더 이상 필요하지 않을 때 메모리를 올바르게 할당 해제하지 않으면 운영 체제의 관점에서 해당 메모리가 "사용 중"으로 유지됩니다. 프로세스가 계속 작동하고 반복적으로 메모리 할당을 취소하면 프로세스에서 사용하는 총 메모리 양이 증가하고 다른 프로세스 및 시스템 기능에 사용할 수 있는 메모리가 감소합니다. 메모리 누수는 일반적으로 소프트웨어 버그 또는 시스템 펌웨어 또는 해당 펌웨어에서 실행 중인 애플리케이션의 문제로 인해 발생합니다.

Cisco Catalyst 9800 WLC의 경우 다음과 같이 메모리 누수가 나타날 수 있습니다.

- 성능 저하: 메모리가 점점 부족해짐에 따라 WLC의 속도가 느려져 관리 기능에 대한 응답 시간이 느려지거나 네트워크에 연결된 클라이언트 장치의 성능이 저하될 수 있습니다.
- 시스템 불안정: 중요한 프로세스에 오류가 발생할 수 있으며, 이로 인해 클라이언트 연결이 끊어지거나 WLC 관리가 불가능하거나 기타 비정상적인 동작이 발생할 수 있습니다.
- 시스템 충돌: 심각한 경우 WLC가 충돌하여 다시 시작할 수 있습니다. 특히 필수 작업을 위해 메모리가 부족할 경우 더욱 그렇습니다.

---

참고: 9800 WLC는 갑작스러운 재부팅/충돌이 발생하여 유출된 메모리를 다시 확보하여 자체적으로 복구할 수 있습니다. 메모리 누수는 버그 동작이므로 컨피그레이션/기능을 일으키는 누수를 비활성화하지 않는 한 리부팅 후에도 누수가 발생합니다.

---

## Syslog

%PLATFORM-4-ELEMENT\_WARNING:R0/0: smand: RP/0 사용된 메모리 값 91%가 경고 수준 88%를 초과합니다.

이 메시지는 traceky, callsite ID 및 diff 통화와 함께 프로세스 이름을 사용하는 상위 3개 메모리를 인쇄합니다.

%PLATFORM-4-ELEMENT\_WARNING: 새시 1 R0/0: smand: 1/RP/0: 사용된 메모리 값 91%가 경고 수준 88%를 초과합니다. 상위 메모리 할당자는 Process: sessmgrd\_rp\_0입니다. Tracekey: 1#258b8858a63c7998252e96352473c9c6 Callsite ID: 11B8F825A8768000(diff\_call: 20941). 프로세스: fman\_fp\_image\_fp\_0. 추적 키: 1#36b34d8e636a89f6397a3b12acab9706 Callsite ID: 1944E78DF68EC002(diff\_call: 19887). 프로세스: linux\_iosd-imag\_rp\_0. 추적 키:

1#8ec74901dc8e23a44e060e69d5820ece 호출 사이트 ID: E2AA338E11594003 (diff\_call: 13404).

## 9800 WLC에서 메모리 누수 문제가 발생했음을 확인하는 방법

메모리 누수는 WLC에서 제공하는 네트워크 서비스의 안정성과 신뢰성을 손상시킬 수 있으므로 신속하게 해결하는 것이 중요합니다. WLC의 메모리 누수를 진단하려면 CLI에서 다양한 명령을 사용하여 시간의 경과에 따른 메모리 사용량을 모니터링할 수 있습니다. 메모리를 해제하지 않고 증가하는 양의 메모리를 사용하는 프로세스나 메모리가 예상대로 재확보되지 않음을 나타내는 패턴을 찾을 수 있습니다.

플랫폼에 할당된 메모리가 총 얼마인지 확인합니다.

```
9800WLC#show version | in memory
cisco C9800-L-F-K9 (KATAR) processor (revision KATAR) with 1634914K/6147K bytes of memory.

32768K bytes of non-volatile configuration memory.
16777216K bytes of physical memory.
```

!! Determines Total platform memory available, Here it is 16GB

각 폴에 할당된 메모리 양을 확인합니다.

```
9800WLC#show processes memory
Processor Pool Total: 1674013452 Used: 823578520 Free: 850434932
reserve P Pool Total: 102404 Used: 88 Free: 102316
lsmapi_io Pool Total: 6295128 Used: 6294296 Free: 832
```

메모리 사용량을 포함하여 리소스 사용률을 확인합니다. Warning 또는 Critical 수준을 초과하면 메모리 누수가 발생할 수 있습니다.

```
9800WLC#show platform resources
**State Acronym: H - Healthy, W - Warning, C - Critical
```

Resource	Usage	Max	Warning	Critical	State
-----					
RP0 (ok, active)					H
Control Processor	21.70%	100%	80%	90%	H
DRAM	5444MB (35%)	15467MB	88%	93%	H
ESP0(ok, active)					H
QFP					H
DRAM	234658KB (12%)	1835008KB	85%	95%	H
IRAM	414KB (20%)	2048KB	85%	95%	H
CPU Utilization	0.00%	100%	90%	95%	H

9800 WLC의 메모리 사용률

컨트롤 플레인 리소스의 전체 메모리 사용량 모니터링

9800WLC#show platform software status control-processor brief

Slot Status 1-Min 5-Min 15-Min  
1-RP0 Healthy 0.52 0.75 0.80

Memory (kB)

Slot Status Total Used (Pct) Free (Pct) Committed (Pct)  
1-RP0 Healthy 16327028 4898110(30%) 114218918 (70%) 5387920 (33%)

상위 프로세스에 대해 할당 및 사용된 메모리 크기를 모니터링합니다. 사용 가능한 메모리가 고정된 상태에서 메모리 사용량이 계속 증가하거나 메모리가 너무 적을 경우 IOSd 레벨에서 메모리 누수가 발생할 가능성이 높습니다.

```
9800WLC#show process memory sorted
Processor Pool Total: 1674013452 Used: 492934952 Free: 1181078500
reserve P Pool Total: 102404 Used: 88 Free: 102316
lsmpi_io Pool Total: 6295128 Used: 6294296 Free: 832

PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 737247000 444817776 268572424 0 0 *Init*
736 0 147160744 85216176 43848536 0 0 Stby Cnfg Parse
722 0 34348696 205824 34480984 0 0 SBC main process
4 0 62523104 35323288 23572272 27362640 27360228 RF Slave Main Th
81 0 22061704 91560 21946768 0 0 EWLC IOSD CAPWAP
93 0 70079512 14591040 19359760 0 0 IOSD ipc task
0 0 0 0 6236576 0 0 *MallocLite*
224 0 10665096 619664 6202672 0 0 SNMP MA SA
```

가장 높은 대기 프로세스부터 시작하는 프로세스당 메모리 통계

플랫폼 레벨 메모리 누수 문제의 경우 RSS(Resident Set Size) 카운터를 모니터링합니다. RSS는 실행 중 프로세스에 할당된 메모리의 양을 나타냅니다. 이 값이 빠르게 증가하면 메모리 누수가 발생할 수 있습니다.

```
9800WLC#show process memory platform sorted
System memory: 15838752K total, 5409956K used, 10428796K free,
Lowest: 10379012K
Pid Text Data Stack Dynamic RSS Name
-----
4272 409975 1482448 136 468 1482448 linux_iosd-imag
19727 22205 448216 136 1680 448216 ucode_pkt_PPE0
19880 182 373884 136 5772 373884 wncmgrd
20381 991 370916 136 16416 370916 wncd_0
24705 536 334212 136 6928 334212 dbm
21097 342 302808 136 1432 302808 cpp_cp_svr
26601 91 295656 136 19228 295656 pubd
31626 58 274280 136 6744 274280 paed
26889 361 263072 136 368 263072 ndbmand
23222 478 259024 136 11136 259024 repm
24961 57 229112 136 228 229112 cli_agent
```

플랫폼이 가장 높은 보유 프로세스로부터 메모리 사용을 처리

# IOS 프로세스의 메모리 누수 문제 해결

IOS XE에서 IOS는 IOSd라고 하는 Linux 커널의 상단에서 실행되는 프로세스(데몬)로 작동합니다. 일반적으로 IOSd는 총 가용 플랫폼 DRAM의 35%에서 50% 사이에 할당됩니다.

## WLC의 기본 로그

모든 명령에 대한 시간 참조를 포함하도록 타임스탬프를 활성화합니다.

```
9800WLC#term exec prompt timestamp
```

## 구성 및 메모리 관련 정보를 검토하려면

```
9800WLC#show tech-support wireless  
9800WLC#show tech-support memory
```

## 코어 덤프 파일 또는 시스템 보고서 수집(생성된 경우)

### GUI 사용

탐색 Troubleshooting > Core Dumps and System Report

Troubleshooting > Core Dump and System Report

Need help on what logs to collect for various scenarios?

Core Dump

Delete

	Date & Time	Size (Bytes)	Name	Download
<input type="checkbox"/>	29 Apr 2024 23:56:21	125665	bootflash-2/core/WLC-1_1_RP_0_code_sign_verify_894_20240429-182620-UTC.core.gz	

System Report

Delete

	Date & Time	Size (Bytes)	Name	Download
<input type="checkbox"/>	03 Jul 2024 00:38:23	14560784	bootflash/core/WLC-2_1_RP_0-system-report_20240703-003816-IST.tar.gz	
<input type="checkbox"/>	25 Jun 2024 23:54:31	16580832	bootflash/core/WLC-2_1_RP_0-system-report_20240625-235418-IST.tar.gz	

## 코어 덤프 및 시스템 보고서

### CLI를 통해

```
9800WLC#show bootflash: | in core/system-report  
9800WLC#copy bootflash:system-report/Core_file {tftp: | ftp: | https: ...}
```

## 프로세서 메모리 풀용

가장 높은 대기 프로세스부터 각 프로세스 메모리를 확인합니다.

```
9800WLC#show process memory sorted
```

관련 풀에 대한 총 메모리 통계를 확인합니다. 또한 부팅 이후 가장 큰 여유 블록 및 가장 낮은 가용 메모리를 보여줍니다.

```
9800WLC#show memory Statistics
```

대용량 메모리를 할당한 프로그램 카운터(PC)를 확인합니다.

```
9800WLC#show memory allocation-process totals
```

유출된 블록과 청크를 확인합니다.

```
9800WLC#show memory debug leak chunks
```

!!This is CPU intensive cli and use only if above CLI output is not helping.

## IO 메모리 풀용

상위 할당자를 확인합니다.

```
9800WLC#show memory io allocating-process totals
```

상위 할당자가 '패킷 데이터 또는 풀 관리자'인 경우 어떤 caller\_pc가 많은 수의 버퍼를 요청했는지 확인합니다.

```
9800WLC#show buffers
```

```
9800WLC#show buffers usage
```

상단 할당자가 또는 'managed\_chunk\_process()' 프로세스이면, 'Chunk Manager' 이는 하나 이상의 청크가 대량의 메모리를 할당함을 의미한다.

9800WLC#show chunk summary  
9800WLC#show chunk brief

프로세스 MallocLite가 상위 할당자인 경우

9800WLC#show memory lite-chunks totals  
9800WLC#show memory lite-chunks stats

Polaris/Platform 수준에서 메모리 누수 문제 해결

플랫폼에서 사용 가능한 메모리 리소스에 대한 메모리 사용량(%)을 확인합니다.

9800WLC#show Platform resources

전체 시스템 메모리 스냅샷을 확인합니다.

9800WLC#show platform software process slot chassis active R0 Monitor | in Mem

모든 플랫폼 프로세스 메모리가 정렬되었는지 확인합니다.

9800WLC#show process memory platform sorted  
9800WLC#show platform software process memory chassisid active r0 all sorted

Callsites의 마지막 시간별 상태를 확인합니다.

9800WLC#show process memory platform accounting

앞의 두 CLI 출력에서 상위 경쟁자를 선택하고 개별 프로세스에 대한 디버그를 활성화합니다.

9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite stop  
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite clear  
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc backtrace start <CALL\_SITE> depth 10

9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite start  
!! Running these debugs has no impact to device

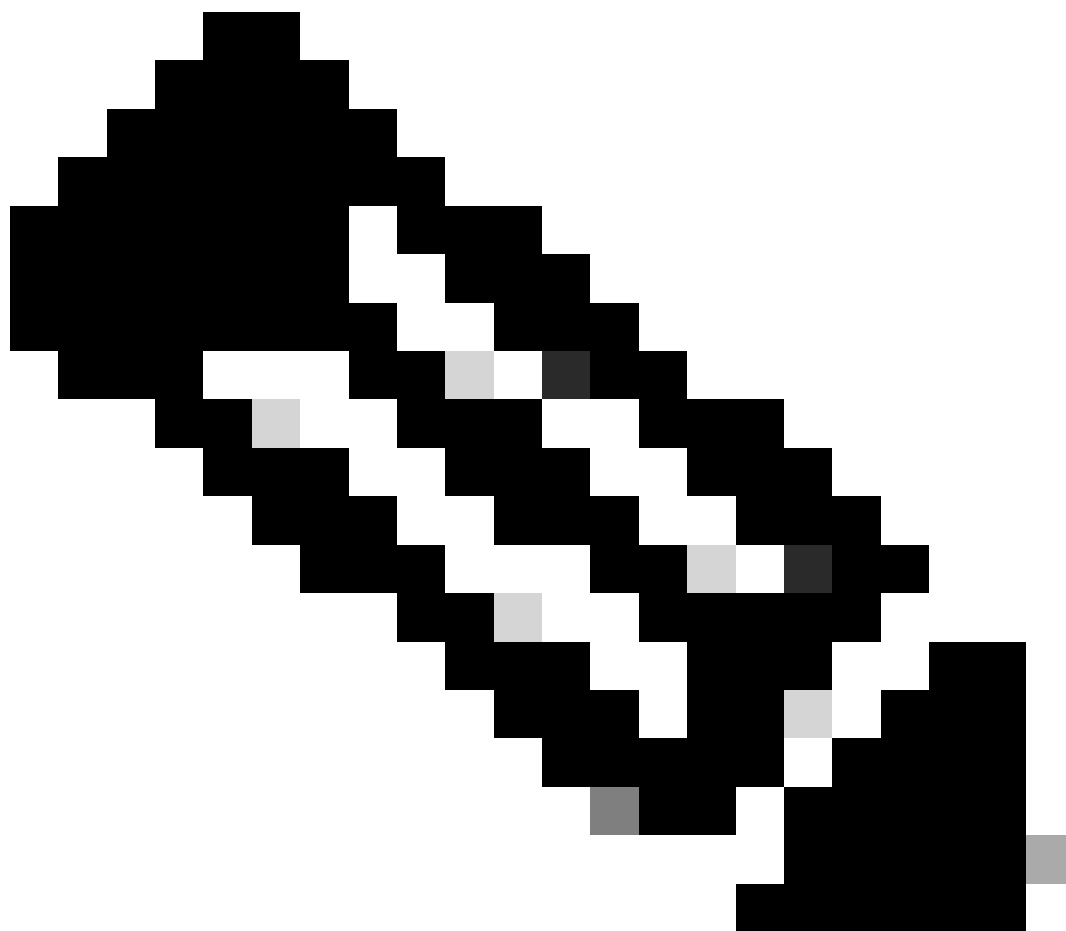
디버그를 시작한 후 몇 분(15분~1시간) 후에 출력을 수집합니다.

9800WLC#show platform software memory <process> chassis <1-2/active/standby> R0 alloc backtrace  
!! Capture this output three times, with a 5-10 minutes interval between each capture, to identify the pattern.

call\_diff를 확인하고 각 프로세스에 대한 해당 백트레이스를 사용하여 값을 할당 및 해제합니다.

9800WLC#show platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite brief

---





---

**참고:**  $call\_diff = allocs - free$

$allocs = frees$ , 메모리 누수 없음

여유 공간 = 0, 메모리 누수

$allocs$ 가  $\neq$  메모리 누수가 될 수도 있고 아닐 수도 있습니다( $call\_diff$ 가 더 많은 경우 메모리 누수의 가능성이 높음을 나타냄).

---

개별 프로세스에 대한 데이터베이스 메모리의 데이터를 캡처합니다.

```
9800WLC#show platform software memory <process> chassis <1-2/active/standby>active R0 alloc type data brief
9800WLC#show platform software memory database <process> chassis <1-2/active/standby> chassis active R0 brief
```

시스템 마운트 정보를 확인하여 임시로 생성된 가상 파일 시스템의 메모리 사용량을 확인합니다.

```
9800WLC#show platform software mount
```

권장 사항

메모리 권장 사항 및 확장 제한에 대한 내용은 관련 컨피그레이션 가이드, 데이터 시트 및 릴리스 정보를 참조하고 WLC가 최신 권장 릴리스로 업그레이드되었는지 확인합니다.

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.