

Understand Memory Leak Issue on 9800 WLC

Contents

[Introduction](#)

[Memory Leak](#)

[Syslog](#)

[How to Identify 9800 WLC have experienced Memory leak issue](#)

[Troubleshooting Memory Leak in IOS Process](#)

[Basic Logs from WLC](#)

[For Processor Memory Pool](#)

[For IO Memory Pool](#)

[Troubleshooting Memory Leak at Polaris/Platform Level](#)

[Recommendation](#)

Introduction

This document describes a memory leak in the context of a Cisco Catalyst 9800 Wireless LAN Controller (WLC).

Memory Leak

When a program or process allocates memory for temporary use and does not correctly deallocate it when it is no longer needed, that memory remains "in use" from the perspective of the operating system. As the process continues to operate and repeatedly fails to deallocate memory, the total amount of memory used by the process grows, and less memory is available for other processes and system functions. Memory leaks are usually caused by software bugs or issues in the system firmware or applications running on it.

In the case of a Cisco Catalyst 9800 WLC, a memory leak can manifest as follows:

- **Decreased Performance:** As memory becomes increasingly scarce, the WLC possibly slows down, resulting in slower response times for management functions or decreased performance of client devices connected to the network.
- **System Instability:** Critical processes can start to fail, possibly leading to dropped client connections, inability to manage the WLC, or other erratic behaviours.
- **System Crashes:** In severe cases, the WLC can possibly crash and restart, especially if it runs out of memory for essential operations.

Note: 9800 WLC can experience a sudden reboot/crash to reclaim the leaked memory and recover itself. Since memory leak is buggy behaviour, leaks occur even after reboot unless the leak causing configuration/feature is disabled.

Syslog

```
%PLATFORM-4-ELEMENT_WARNING:R0/0: smand: RP/0 Used Memory Value 91% exceeds warning level 88%
```

This message prints the top 3 memory consuming process' name along with the tracekey, callsite ID and diff calls:

```
%PLATFORM-4-ELEMENT_WARNING: Chassis 1 R0/0: smand: 1/RP/0: Used Memory value 91% exceeds warning level 88%. Top memory allocators are: Process: sessmgrd_rp_0. Tracekey: 1#258b8858a63c7998252e96352473c9c6 Callsite ID: 11B8F825A8768000 (diff_call: 20941). Process: fman_fp_image_fp_0. Tracekey: 1#36b34d8e636a89f6397a3b12acab9706 Callsite ID: 1944E78DF68EC002 (diff_call: 19887). Process: linux_iosd-imag_rp_0. Tracekey: 1#8ec74901dc8e23a44e060e69d5820ece Callsite ID: E2AA338E11594003 (diff_call: 13404).
```

How to Identify 9800 WLC have experienced Memory leak issue

It is important to address memory leaks promptly as they can compromise the stability and reliability of the network services provided by the WLC. To diagnose a memory leak on a WLC, you can use various commands on the CLI to monitor memory usage over time. They might look for processes that are using an increasing amount of memory without releasing it or patterns that indicate memory is not being reclaimed as expected.

Check how much memory is totally allocated to platform.

```
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

Check how much memory is allocated to each pool.

```
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
```

Check resource utilization, including memory usage. If it exceeds the Warning or Critical levels, it can indicate a potential memory leak.

```
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

Memory Utilization on 9800 WLC

Monitor overall memory usage for control plane resources

```
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-RPO Healthy 16327028 4898110(30%) 114218918 (70%) 5387920 (33%)

Monitor the allocated and used memory size for the top processes. If the memory usage continues to increase while the free memory remains fixed or is too low, there is a high chances of a memory leak at the IOSd level.

```
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
```

Per process memory stats starting from the highest holding process

For platform-level memory leak issues, monitor the RSS (Resident Set Size) counters. RSS indicates the amount of memory allocated to a process during execution. If this value increases rapidly, it could signify a potential memory leak.

```
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
```

Platform processes memory usage from the highest holding process

Troubleshooting Memory Leak in IOS Process

In IOS XE, IOS operates as a process (daemon) running on top of the Linux kernel, known as IOSd. Typically, IOSd is allocated between 35% to 50% of the total available platform DRAM.

Basic Logs from WLC

Enable timestamp to have time reference for all the commands.

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

To review the configuration and memory related information:

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

Collect Core Dump file or System Report if generated


Via GUI

Navigate to 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	

1 - 1 of 1 items

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	

Core Dump and System Report

Via CLI

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

For Processor Memory Pool

Check per process memory starting from the highest holding process.

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

Check the total memory stats for the concerned pool. It also shows largest free block and lowest available

memory since boot.

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

Check the program counter (PC) which allocated large amount of memory.

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

Check leaked blocks and chunks.

```
9800WLC#show memory debug leak chunks  
!!This is CPU intensive cli and use only if above CLI output is not helping.
```

For IO Memory Pool

Check the top allocators.

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

If the top allocator is 'Packet Data or Pool Manager' , check which caller_pc requested large number of buffers

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

If the top allocator is 'managed_chunk_process()' or 'Chunk Manager' process, then it means one or more chunks is/are allocating large amount of memory.

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

If the process MallocLite is the top allocator

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

Troubleshooting Memory Leak at Polaris/Platform Level

Check memory usage % for available memory resources on platform.

```
9800WLC#show Platform resources
```

Check the overall system memory snapshot.

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

Check all platform processes memory sorted.

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

Check last hourly status of callsites.

```
9800WLC#show process memory platform accounting
```

Pick the top contender from the previous two CLI outputs and enable the debugs for the individual processes.

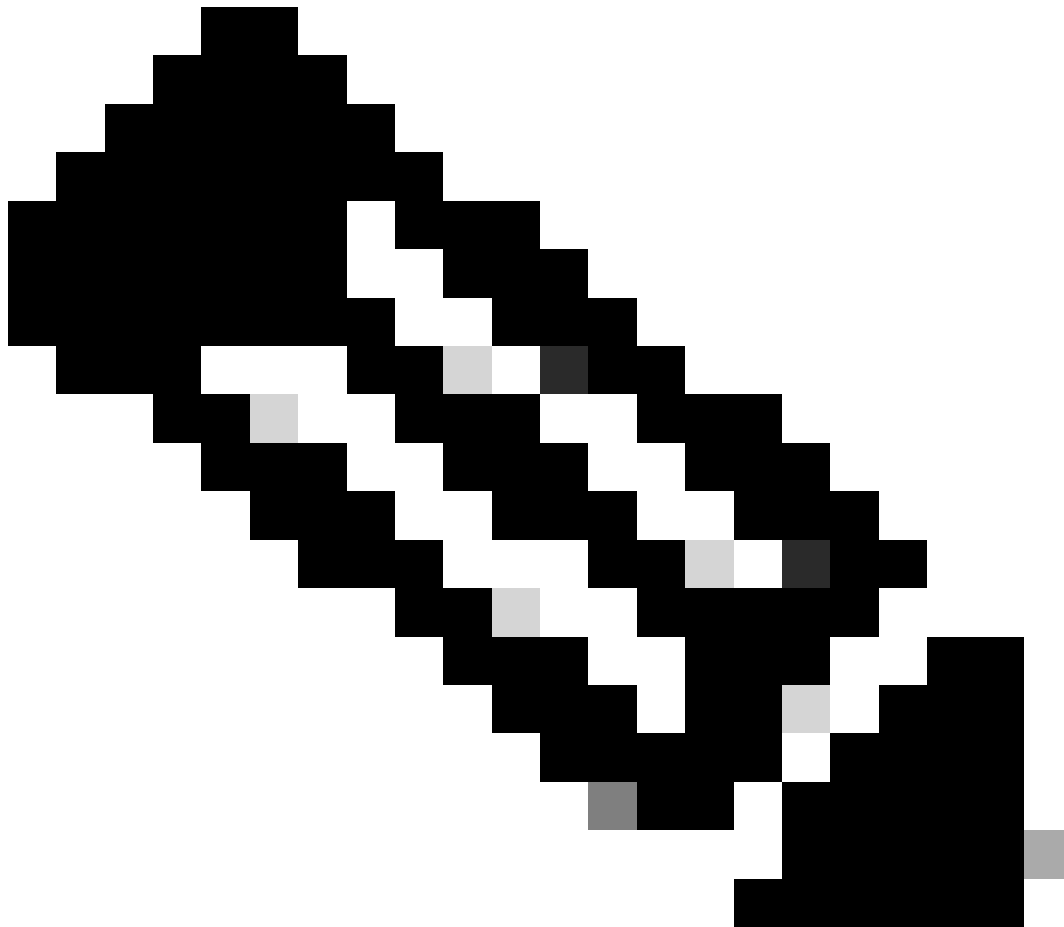
```
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite stop
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite clear
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc backtrace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

Collect the output a few minutes (15 minutes to one hour) after initiating the debugs.

```
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
```

Check for call_diff, allocs and frees value with the respective backtrace for each process.

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



Note: $call_diff = allocs - frees$

If $allocs = frees$, no memory leak

If $frees = 0$, memory leak

If $allocs \neq frees$, maybe or maybe not be memory leak (If $call_diff$ is more, it indicates high chances of memory leak)

Capture data of database memory for individual process.

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


Check system mount information to check the memory usage for temporarily created virtual file system.

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

Recommendation

Refer to the relevant configuration guides, data sheets, and release notes for memory recommendations and scaling limits, and ensure the WLC is upgraded to the latest recommended release.