ASA SNMP Polling for Memory-Related Statistics

Contents

Introduction

This document describes how to use Simple Network Management Protocol (SNMP) in order to query the Cisco Adaptive Security Appliance (ASA) memory statistics—such as free memory, used memory, and so on.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is valid only for Cisco Adaptive Security Appliance devices.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

You can monitor the free memory and the used memory statistics in order to identify the memory performance of the network device. Cisco ASA supports memory statistics to be polled through SNMP and uses these supported OIDs:

- **32-bit Memory Counters**Use the ‘CiscoMemoryPoolEntry’ object. Object and the ID mappings are shown in this sample output.

  ```
  ciscoMemoryPoolType1.3.6.1.4.1.9.9.48.1.1.1.1
  ciscoMemoryPoolName1.3.6.1.4.1.9.9.48.1.1.1.2
  ciscoMemoryPoolAlternate1.3.6.1.4.1.9.9.48.1.1.1.3
  ciscoMemoryPoolValid1.3.6.1.4.1.9.9.48.1.1.1.4
  ciscoMemoryPoolUsed1.3.6.1.4.1.9.9.48.1.1.1.5
  ciscoMemoryPoolFree1.3.6.1.4.1.9.9.48.1.1.1.6
  ciscoMemoryPoolLargestFree1.3.6.1.4.1.9.9.48.1.1.1.7
  ```
- **64-bit Memory Counters**Use the ‘cempMemPoolEntry’ object. Object and the ID mappings are shown in this sample output.

  ```
  cempMemPoolIndex1.3.6.1.4.1.9.9.221.1.1.1.1.1
  cempMemPoolLowestFree1.3.6.1.4.1.9.9.221.1.1.1.1.10
  cempMemPoolUsedLowWaterMark1.3.6.1.4.1.9.9.221.1.1.1.1.11
  cempMemPoolAllocHit1.3.6.1.4.1.9.9.221.1.1.1.1.12
  cempMemPoolAllocMiss1.3.6.1.4.1.9.9.221.1.1.1.1.13
  cempMemPoolFreeHit1.3.6.1.4.1.9.9.221.1.1.1.1.14
  cempMemPoolFreeMiss1.3.6.1.4.1.9.9.221.1.1.1.1.15
  cempMemPoolType1.3.6.1.4.1.9.9.221.1.1.1.1.2
  cempMemPoolName1.3.6.1.4.1.9.9.221.1.1.1.1.3
  cempMemPoolPlatformMemory1.3.6.1.4.1.9.9.221.1.1.1.1.4
  cempMemPoolAlternate1.3.6.1.4.1.9.9.221.1.1.1.1.5
  cempMemPoolValid1.3.6.1.4.1.9.9.221.1.1.1.1.6
  cempMemPoolUsed1.3.6.1.4.1.9.9.221.1.1.1.1.7
  cempMemPoolFree1.3.6.1.4.1.9.9.221.1.1.1.1.8
  cempMemPoolLargestFree1.3.6.1.4.1.9.9.221.1.1.1.1.9
  ```

SNMP Polled Output

When the memory statistics are queried from the console of an SNMP client, the output looks similar to this sample output.

For 32-bit counters:

```
iso.3.6.1.4.1.9.9.48.1.1.1.2.1 = STRING: "System memory"
iso.3.6.1.4.1.9.9.48.1.1.1.2.6 = STRING: "MEMPOOL_DMA"
iso.3.6.1.4.1.9.9.48.1.1.1.2.7 = STRING: "MEMPOOL_GLOBAL_SHARED"
iso.3.6.1.4.1.9.9.48.1.1.1.3.1 = INTEGER: 0
iso.3.6.1.4.1.9.9.48.1.1.1.3.6 = INTEGER: 0
iso.3.6.1.4.1.9.9.48.1.1.1.3.7 = INTEGER: 0
iso.3.6.1.4.1.9.9.48.1.1.1.4.1 = INTEGER: 1
iso.3.6.1.4.1.9.9.48.1.1.1.4.6 = INTEGER: 1
iso.3.6.1.4.1.9.9.48.1.1.1.4.7 = INTEGER: 1
iso.3.6.1.4.1.9.9.48.1.1.1.5.1 = Gauge32: 230971224
iso.3.6.1.4.1.9.9.48.1.1.1.5.6 = Gauge32: 21585704
iso.3.6.1.4.1.9.9.48.1.1.1.5.7 = Gauge32: 50616136
iso.3.6.1.4.1.9.9.48.1.1.1.6.1 = Gauge32: 37464232
iso.3.6.1.4.1.9.9.48.1.1.1.6.6 = Gauge32: 32964824
iso.3.6.1.4.1.9.9.48.1.1.1.6.7 = Gauge32: 37464248
iso.3.6.1.4.1.9.9.48.1.1.1.7.1 = Gauge32: 37460160
iso.3.6.1.4.1.9.9.48.1.1.1.7.6 = Gauge32: 32945592
iso.3.6.1.4.1.9.9.48.1.1.1.7.7 = Gauge32: 37460160
```

You can use the output from the **show mem** or **show mem detail** commands in order to interpret the same.

```
‘iso.3.6.1.4.1.9.9.48.1.1.1.5.1 = Gauge32:’ correlates to the ‘Used Memory’ in 'sh
mem' output.

‘iso.3.6.1.4.1.9.9.48.1.1.1.6.1 = Gauge32:’ correlates to the ‘Free Memory’ in ‘sh
mem' output
```

For 64-bit counters:

```
iso.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 = INTEGER: 2
iso.3.6.1.4.1.9.9.221.1.1.1.1.3.1.1 = STRING: "System memory"
iso.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 = INTEGER: 0
iso.3.6.1.4.1.9.9.221.1.1.1.1.6.1.1 = INTEGER: 1
iso.3.6.1.4.1.9.9.221.1.1.1.1.7.1.1 = Gauge32: 230971320
iso.3.6.1.4.1.9.9.221.1.1.1.1.8.1.1 = Gauge32: 37464144
iso.3.6.1.4.1.9.9.221.1.1.1.1.17.1.1 = Gauge32: 0
iso.3.6.1.4.1.9.9.221.1.1.1.1.18.1.1 = Counter64: 230971312
iso.3.6.1.4.1.9.9.221.1.1.1.1.19.1.1 = Gauge32: 0
iso.3.6.1.4.1.9.9.221.1.1.1.1.20.1.1 = Counter64: 37464144
```

You can use the output from the **show mem** or **show mem detail** commands in order to interpret the same.

```
ASA1#
ASA1#
ASA1#
ASA1# show mem Free memory: 37498488 bytes (14%) Used memory: 230936968 bytes (86%) -
------------ ------------------ Total memory: 268435456 bytes (100%) ASA1# ASA1#
ASA1# ASA1# show mem detail Free memory: 37498488 bytes (14%) Used memory: Allocated
memory in use: 50581896 bytes (19%) Reserved memory: 180355072 bytes (67%) ----------
------------------- ------------------ Total memory: 268435456 bytes (100%) Least
free memory: 37463768 bytes (14%) Most used memory: 230971688 bytes (86%) !--- Some
output excluded.
```

Known Caveats

This section describes some known caveats while polling memory statistics using SNMP

When the ASA is queried to poll memory information, SNMP could search the information from three main segments of ASA memory as listed below.

1. The system memory pool
2. The MEMPOOL_DMA pool
3. The MEMPOOL_GLOBAL_SHARED pool

If the MEMPOOL_GLOBAL_SHARED pool information is queried via SNMP, it results in CPU hogs. It is evident that you might see packet drops/overruns at times of bursty/overloaded traffic when you use SNMP to poll memory statistics that require the ASA to query the information through huge chunks of memory that it is associated which results in SNMP related CPU hogs. The ASA's CPU may be held by the SNMP process for too long before releasing the CPU to other processes. If the data rate is high enough through the ASA, overruns will increase on the interface counters and packets might be dropped.

It is applicable for both for Single-Core and Multi-Core platforms. It is advisable not to use the memory pool MIBs to poll those statistics that relate to **show mem detail** but to use only those MIBs that associate with the **show mem** output. You can run **show mem detail** from the CLI in order to view these CPU hogs.

CPU Hogs for SNMP

This section provides sample CPU hog messages from the Cisco ASA.

```
Process:      snmp, PROC_PC_TOTAL: 124, MAXHOG: 306, LASTHOG: 299
LASTHOG At:   12:00:24 EDT May 17 2013
PC:           0x000000000124fd5c (suspend)

Process:      snmp, NUMHOG: 124, MAXHOG: 306, LASTHOG: 299
LASTHOG At:   12:00:24 EDT May 17 2013
PC:           0x000000000124fd5c (suspend)
Call stack:   0x000000000124fd5c  0x000000000124e72b  0x000000000124b5da
              0x000000000124e3e7  0x00000000001228b9a  0x000000000122732a
              0x0000000000423cc5

Process:      snmp, PROC_PC_TOTAL: 248, MAXHOG: 306, LASTHOG: 298
LASTHOG At:   12:01:34 EDT May 17 2013
PC:           0x00000000013780cf (suspend)

Process:      snmp, NUMHOG: 248, MAXHOG: 306, LASTHOG: 298
LASTHOG At:   12:01:34 EDT May 17 2013
PC:           0x00000000013780cf (suspend)
Call stack:   0x000000000124803b  0x00000000012289e5  0x000000000122732a
              0x0000000000423cc5
```

You might also see these error messages on Cisco ASA.

```
[local5.warning] %ASA-4-711004: Task ran for 305 msec, Process = snmp, PC = 1250117,
Call stack =
2013-05-17T09:33:12-04:00 CISCO-ASA-TEST(10.10.10.1) [local5.warning] %ASA-4-711004:
Task ran for 305 msec, Process = snmp, PC = 1250117, Call stack =
0x0000000001250117  0x000000000124ea07  0x000000000124b5da  0x000000000124e3e7
0x0000000001228b9a  0x000000000122732a  0x0000000000423cc5
2013-05-17T09:33:12-04:00 CISCO-ASA-TEST(10.10.10.2) [local5.warning] %ASA-4-711004:
Task ran for 354 msec, Process = snmp, PC = 1250117, Call stack =
2013-05-17T09:33:12-04:00 CISCO-ASA-TEST(10.10.10.2) [local5.warning] %ASA-4-711004:
Task ran for 354 msec, Process = snmp, PC = 1250117, Call stack =
0x0000000001250117  0x000000000124ea07  0x000000000124b5da  0x000000000124e3e7
0x0000000001228b9a  0x000000000122732a  0x0000000000423cc5
2013-05-17T09:33:22-04:00 CISCO-ASA-TEST(10.10.10.2) [local5.warning] %ASA-4-711004:
Task ran for 348 msec, Process = snmp, PC = 124fd5c, Call stack =
2013-05-17T09:33:22-04:00 CISCO-ASA-TEST(10.10.10.2) [local5.warning] %ASA-4-711004:
Task ran for 348 msec, Process = snmp, PC = 124fd5c, Call stack =
0x000000000124fd5c  0x000000000124e72b  0x000000000124b5da  0x000000000124e3e7
0x0000000001228b9a  0x000000000122732a  0x0000000000423cc5
2013-05-17T09:36:17-04:00 CISCO-ASA-TEST(10.10.10.1) [local5.warning] %ASA-4-711004:
Task ran for 300 msec, Process = snmp, PC = 13780cf, Call stack =
2013-05-17T09:36:17-04:00 CISCO-ASA-TEST(10.10.10.1) [local5.warning] %ASA-4-711004:
Task ran for 300 msec, Process = snmp, PC = 13780cf, Call stack =
0x000000000124803b  0x00000000012289e5  0x000000000122732a  0x0000000000423cc5
```

Mitigation

1. Avoid polling the OIDs that relate to global_shared_mem_pool.
2. Run the snmpwalk for your ASA platform and verify if any of the OIDs are being polled that relate to the global_shared_mem_pool.

```
snmpwalk -c public -v2c -Os <IP Address> 1.3.6.1.4.1.9.9.48
enterprises.9.9.48.1.1.1.2.1 = STRING: "System memory" enterprises.9.9.48.1.1.1.2.6 =
STRING: "MEMPOOL_DMA" enterprises.9.9.48.1.1.1.2.7 = STRING: "MEMPOOL_GLOBAL_SHARED"
enterprises.9.9.48.1.1.1.3.1 = INTEGER: 0 enterprises.9.9.48.1.1.1.3.6 = INTEGER: 0
enterprises.9.9.48.1.1.1.3.7 = INTEGER: 0 enterprises.9.9.48.1.1.1.4.1 = INTEGER: 1
enterprises.9.9.48.1.1.1.4.6 = INTEGER: 1 enterprises.9.9.48.1.1.1.4.7 = INTEGER: 1
enterprises.9.9.48.1.1.1.5.1 = Gauge32: 804874736 enterprises.9.9.48.1.1.1.5.6 =
Gauge32: 125674744 enterprises.9.9.48.1.1.1.5.7 = Gauge32: 153938632
enterprises.9.9.48.1.1.1.6.1 = Gauge32: 3490092567 enterprises.9.9.48.1.1.1.6.6 =
Gauge32: 146135816 enterprises.9.9.48.1.1.1.6.7 = Gauge32: 3084064048
enterprises.9.9.48.1.1.1.7.1 = Gauge32: 3083999920 enterprises.9.9.48.1.1.1.7.6 =
Gauge32: 146133824 enterprises.9.9.48.1.1.1.7.7 = Gauge32: 3083999920
```

Avoid using *enterprise.9.9.48.1.1.1.7* as this is the OID for largest_contiguous_memory. Also, the *enterprises.9.9.48.1.1.1.X.7* should be avoided as it relates to MEMPOOL_GLOBAL_SHARED.

When polling OIDs from the family, 9.9.48.1.1.1.x.y , verify if ‘y’ is matching global_mempool; if yes, avoid using those OIDs to mitigate the SNMP CPU hogs. This ‘y’ is generated dynamically and might be different with different Cisco ASA platforms.

Please refer CSCtx43501 for additional details.

Related Information

- **Technical Support & Documentation - Cisco Systems**