

# 메모리 관련 통계에 대한 ASA SNMP 폴링

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[SNMP 폴링된 출력](#)

[알려진 주의 사항](#)

[SNMP용 CPU 속도](#)

[완화](#)

[관련 정보](#)

## 소개

이 문서에서는 Cisco ASA(Adaptive Security Appliance) 메모리 통계(예: 사용 가능한 메모리, 사용 메모리 등)를 쿼리하기 위해 SNMP(Simple Network Management Protocol)를 사용하는 방법에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

이 문서에 대한 특정 요건이 없습니다.

### 사용되는 구성 요소

이 문서의 정보는 Cisco Adaptive Security Appliance 장치에만 유효합니다.

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

## 배경 정보

사용 가능한 메모리 및 사용된 메모리 통계를 모니터링하여 네트워크 디바이스의 메모리 성능을 식별할 수 있습니다. Cisco ASA는 SNMP를 통해 폴링될 메모리 통계를 지원하며 다음과 같은 지원되는 OID를 사용합니다.

- **32비트 메모리 카운터** 'CiscoMemoryPoolEntry' 개체를 사용합니다. 개체 및 ID 매핑이 이 샘플

출력에 표시됩니다.

```
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
```

- **64비트 메모리 카운터**'cempMemPoolEntry' 개체를 사용합니다. 개체 및 ID 매핑이 이 샘플 출력에 표시됩니다.

```
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 폴링된 출력

SNMP 클라이언트의 콘솔에서 메모리 통계를 쿼리하면 출력은 이 샘플 출력과 유사합니다.

32비트 카운터:

```
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
```

**show mem**의 출력 또는 **show mem detail** 명령을 사용하여 동일한 내용을 해석할 수 있습니다.

'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

64비트 카운터:

```
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
```

**show mem**의 출력 또는 **show mem detail** 명령을 사용하여 동일한 내용을 해석할 수 있습니다.

```
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.*

## 알려진 주의 사항

이 섹션에서는 SNMP를 사용하여 메모리 통계를 폴링하는 동안 알려진 몇 가지 주의 사항에 대해 설명합니다

ASA에 메모리 정보를 폴링하도록 쿼리하면 SNMP는 아래에 나열된 ASA 메모리의 세 가지 기본 세그먼트에서 정보를 검색할 수 있습니다.

1. 시스템 메모리 풀
2. MEMPOOL\_DMA 풀
3. MEMPOOL\_GLOBAL\_SHARED 풀

SNMP를 통해 MEMPOOL\_GLOBAL\_SHARED 풀 정보를 쿼리하면 CPU 속도가 저하됩니다 .SNMP를 사용하여 ASA가 연결된 대용량 메모리 청크를 통해 정보를 쿼리해야 하는 메모리 통계를 폴링할 때 과부하/과부하 트래픽 시 패킷이 삭제/오버런되는 것을 볼 수 있으며, 이로 인해 SNMP 관련 CPU 돼지를 얻을 수 있습니다.ASA의 CPU는 SNMP 프로세스에서 너무 오랫동안 유지한 후 CPU를 다른 프로세스로 해제할 수 있습니다.ASA를 통해 데이터 속도가 충분히 높으면 인터페이스 카운터에서 오버런이 증가하며 패킷이 삭제될 수 있습니다.

싱글 코어 및 멀티 코어 플랫폼 모두에 적용됩니다.메모리 풀 MIB를 사용하여 메모리 세부 정보 표시와 관련된 통계를 폴링하지 않고 **show mem** 출력과 연결된 MIB만 사용하는 것이 좋습니다.CLI에

서 show mem detail을 실행하여 이러한 CPU 돼지를 볼 수 있습니다.

## SNMP용 CPU 속도

이 섹션에서는 Cisco ASA의 샘플 CPU 호그 메시지를 제공합니다.

```
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 0x0000000001228b9a 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
```

Cisco ASA에서 이러한 오류 메시지가 표시될 수도 있습니다.

```
[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
```

## 안화

1. global\_shared\_mem\_pool과 관련된 OID는 폴링하지 마십시오.
2. ASA 플랫폼에 대한 snmpwalk를 실행하고 global\_shared\_mem\_pool과 관련된 OID가 폴링되

고 있는지 확인합니다.

```
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
```

*enterprise.9.9.48.1.1.1.7*은 largest\_continuous\_memory의 OID이므로 사용하지 마십시오. 또한 *enterprises.9.9.48.1.1.1.X.7*은 MEMPOOL\_GLOBAL\_SHARED와 관련되므로 피해야 합니다.

패밀리에서 OID를 폴링할 때 9.9.48.1.1.1.x.y, 'y'가 global\_mempool과 일치하는지 확인합니다.대답이 "예"인 경우 SNMP CPU 돼지를 완화하기 위해 이러한 OID를 사용하지 마십시오.이 'y'는 동적으로 생성되며 다른 Cisco ASA 플랫폼과 다를 수 있습니다.

자세한 내용은 [CSCtx43501](#)을 참조하십시오.

## 관련 정보

- [기술 지원 및 문서 - Cisco Systems](#)