

FTD에서 높은 CPU를 사용하는 Linux 프로세스 및 코어 식별

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제 해결](#)

[높은 CPU를 사용하는 Linux 프로세스 식별](#)

[Cisco Secure Firewall 시스템 코어에서 높은 CPU 사용량](#)

[관련 정보](#)

소개

이 문서에서는 Cisco FTD(Secure Firewall Threat Defense)의 높은 CPU 사용량에 대해 설명하며, 여기서 high는 90% 이상의 CPU 사용량을 의미합니다.

사전 요구 사항

요구 사항

- Cisco FMC(Secure Firewall Management Center).
- Cisco FTD(Secure Firewall Threat Defense).

사용되는 구성 요소

- 버전 7.4.2를 실행하는 Cisco Secure Firewall Management Center
- 버전 7.4.2를 실행하는 Cisco Secure Firewall Threat Defense

배경 정보

Cisco Secure Firewall은 Lina와 Snort의 통합 이미지입니다. 프로세스가 Cisco Secure Firewall 시스템의 백엔드를 손상시킵니다. Cisco Secure Firewall의 프로세스 및 스레드에는 선호도가 있습니다. 즉 특정 CPU 코어 또는 코어 집합에 프로세스 또는 스레드를 바인딩하는 기능을 의미합니다. Lina나 Snort와 같은 일부 프로세스는 특정 코어에 독점적인 친화성을 가지므로 이러한 코어에서 다른 프로세스나 스레드를 실행할 수 없습니다.

문제 해결

높은 CPU를 사용하는 Linux 프로세스 식별

Cisco Secure Firewall에서 높은 CPU를 사용하는 프로세스를 식별하려면 다음 출력을 참조하십시오.

top 명령 출력은 대부분의 CPU를 사용하는 프로세스를 식별하는 데 도움이 됩니다.

```
top - 17:27:37 up 62 days, 22:52, 2 users, load average: 0.41, 0.55, 0.48
Tasks: 158 total, 1 running, 157 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.9 us, 1.1 sy, 0.7 ni, 93.4 id, 3.9 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7981.8 total, 821.0 free, 3184.3 used, 3976.5 buff/cache
MiB Swap: 5378.2 total, 4361.2 free, 1017.0 used. 4563.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3394	root	25	5	425680	4860	4652	S	7.6	0.1	6494:36	loggerd
3685	root	0	-20	2958720	1.4g	130268	S	6.0	18.0	5062:29	lina
4494	root	1	-19	1339224	509956	44688	S	1.7	6.2	1659:12	snort3
3979	root	20	0	2270592	141480	16840	S	0.7	1.7	637:29.39	SFDataCorrelator

1을 누르고 코어당 CPU 사용률을 확인합니다.

```
top - 17:30:36 up 62 days, 22:55, 2 users, load average: 0.33, 0.47, 0.45
Tasks: 157 total, 1 running, 156 sleeping, 0 stopped, 0 zombie
%Cpu0 : 3.0 us, 0.0 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu1 : 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu2 : 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu3 : 1.3 us, 5.3 sy, 2.7 ni, 90.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7981.8 total, 831.5 free, 3173.6 used, 3976.7 buff/cache
MiB Swap: 5378.2 total, 4361.2 free, 1017.0 used. 4574.7 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3394	root	25	5	425680	4860	4652	S	7.6	0.1	6494:36	loggerd
3685	root	0	-20	2958720	1.4g	13026	S	6.0	18.0	5062:29	lina
4494	root	1	-19	1339224	509956	44688	S	1.7	6.2	1659:12	snort3
3979	root	20	0	2270592	141480	16840	S	0.7	1.7	637:29.39	SFDataCorrelator

c를 눌러 프로세스의 세부사항을 확인합니다.

```
top - 17:31:53 up 62 days, 22:56, 2 users, load average: 0.34, 0.45, 0.45
Tasks: 157 total, 1 running, 156 sleeping, 0 stopped, 0 zombie
%Cpu0 : 3.7 us, 1.0 sy, 0.0 ni, 95.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu1 : 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu2 : 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu3 : 2.0 us, 3.7 sy, 2.7 ni, 91.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7981.8 total, 831.4 free, 3173.7 used, 3976.7 buff/cache
MiB Swap: 5378.2 total, 4361.2 free, 1017.0 used. 4574.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3394	root	25	5	425680	4860	4652	S	7.3	0.1	6494:55	/ngfw/usr/local/sf/bin/loggerd
3685	root	0	-20	2958720	1.4g	130268	S	6.0	18.0	5062:45	lina -p 3562 -t -l
4494	root	1	-19	1339224	509956	44688	S	2.3	6.2	1659:17	/ngfw/var/sf/detection_engines/ad5
3418	root	20	0	1640540	16824	13900	S	0.7	0.2	274:48.05	/ngfw/usr/local/sf/bin/adi
3979	root	20	0	2270592	141480	16840	S	0.7	1.7	637:31.13	/ngfw/usr/local/sf/bin/SFDataCorre
3380	root	20	0	3860	2852	2336	S	0.3	0.0	6:27.14	/bin/bash /ngfw/usr/local/sf/bin/p

monetdb가 원인임을 최상위 명령에서 확인한 후 높은 CPU를 소비하는 프로세스(monetdb)의 예를 고려하십시오.

expert 모드 ngfw/var/log/monetdb/merovingian.log에서 merovingian.log를 추적할 수 있습니다.

```
root@FirePower:/# cd /ngfw/var/log/#catmerovingian.log
```

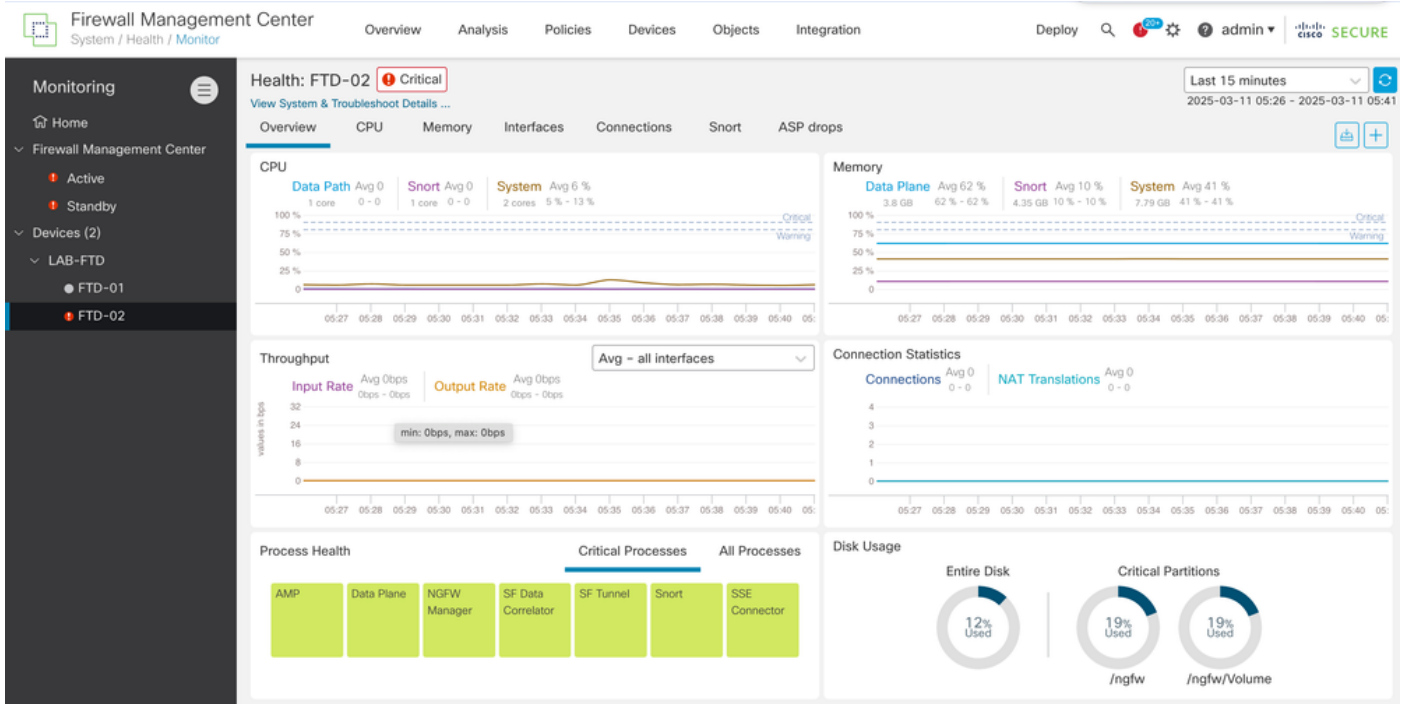
```
2025-03-06 23:24:11 ERR eventdb[26531]: #client12: createExceptionInternal:
!ERROR: SQLException:sql.drop_table:42000!DROP TABLE: unable to drop table connectionevent_1720808160_0
2025-03-06 23:24:11 ERR eventdb[26531]: #client12: createExceptionInternal:
!ERROR: SQLException:sql.drop_table:42000!DROP TABLE: unable to drop table
connectionevent_1720971900_0 (there are database objects which depend on it)
```

Cisco Secure Firewall 시스템 코어에서 높은 CPU 사용량

다음은 Cisco Secure Firewall 시스템 코어에서 CPU가 높은 경우의 문제 해결 단계입니다.

1. FMC 상태 모니터링에서 높은 CPU 시스템 사용량을 확인합니다.

- FMC에서 알림 열을 확인하여 높은 CPU 사용량을 확인합니다.
- System(시스템) > Health(상태) > Monitor(모니터링)로 이동하여 영향받는 코어를 관찰합니다



2. FTD CLI에서 영향을 받는 CPU 코어를 확인합니다.

- CPU 선호도를 확인하고 영향을 받는 코어가 시스템 코어인지 확인합니다.

```
> pmtool show affinity
```

```
Received status (0):
```

```
Affinity Status
```

```
System CPU Affinity: 3,9 (desired 3,9 )
```

```
..
```

```
<#root>
```

```
admin@firepower:~$
```

```
sudo mpstat -P 3,9 1
```

```
Linux 4.18.45-yocto-standard (firepower)      03/03/25      _x86_64_      (64 CPU)
```

```
1:06:15      CPU
```

```
%usr    %nice    %sys
```

```
%iowait  %irq     %soft  %steal  %guest  %gnice
```

```
%idle
```

```
1:06:15      3
```

```
4
```

.00 54.00 24.00

0.00 0.00 0.00 0.00 0.00 0.00

14.00

1:06:15 9

0.00 74.00 25.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00

1:06:15 CPU

%usr %nice %sys

 %iwait %irq %soft %steal %guest %gnice %idle

1:06:16 3

6

.00 44.00 34.00

0.00 0.00 0.00 0.00 0.00 0.00

4

.00

1:06:16 9

2.00 74.00 22.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00

1:06:16 CPU

%usr %nice %sys

 %iwait %irq %soft %steal %guest %gnice %idle

1:06:16 3

6

.00 34.00 24.00

0.00 0.00 0.00 0.00 0.00 0.00

34.00

1:06:16 9

0.00 74.00 24.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00

...

Average: 9 3.71 67.43 22.14 0.00 0.00 0.00 0.00 0.00 0.00 9.71

- 시스템 코어에서 CPU 사용률을 모니터링하려면 runmpstatsarto를 실행합니다. 이 예에서 1은 모니터링 간격 1/초를 나타냅니다.

CPU 사용률이 높을수록 %의 유휴 상태가 100 - 다른 % 열의 합계와 같기 때문에 낮습니다.

<#root>

admin@firepower:~\$

sudo

sar -P 3,9 1

Linux 4.18.45-yocto-standard (firepower) 03/03/24 _x86_64_ (64 CPU)

1:06:17 CPU

%user %nice %system

%iowait %steal

%idle

1:06:18 3

5

.00 54.00 28.00

0.00 0.00

14.00

1:06:18 9

1.00 80.00 19.00

0.00 0.00

0.00

1:06:18 CPU

%user %nice %system

%iowait %steal

%idle

1:06:19 3

5
.00 41.00 22.00
0.00 0.00
24.00

1:06:19 9
2.00 84.00 12.00
0.00 0.00
0.00

1:06:19 CPU
%user %nice %system
%iowait %steal
%idle

1:06:20 3
1.00 67.00 15.00
0.00 0.00
4
.00

1:06:20 9
5
.00 68.00 24.00
0.00 0.00
0.00

1:06:20 CPU
%user %nice %system
%iowait %steal
%idle

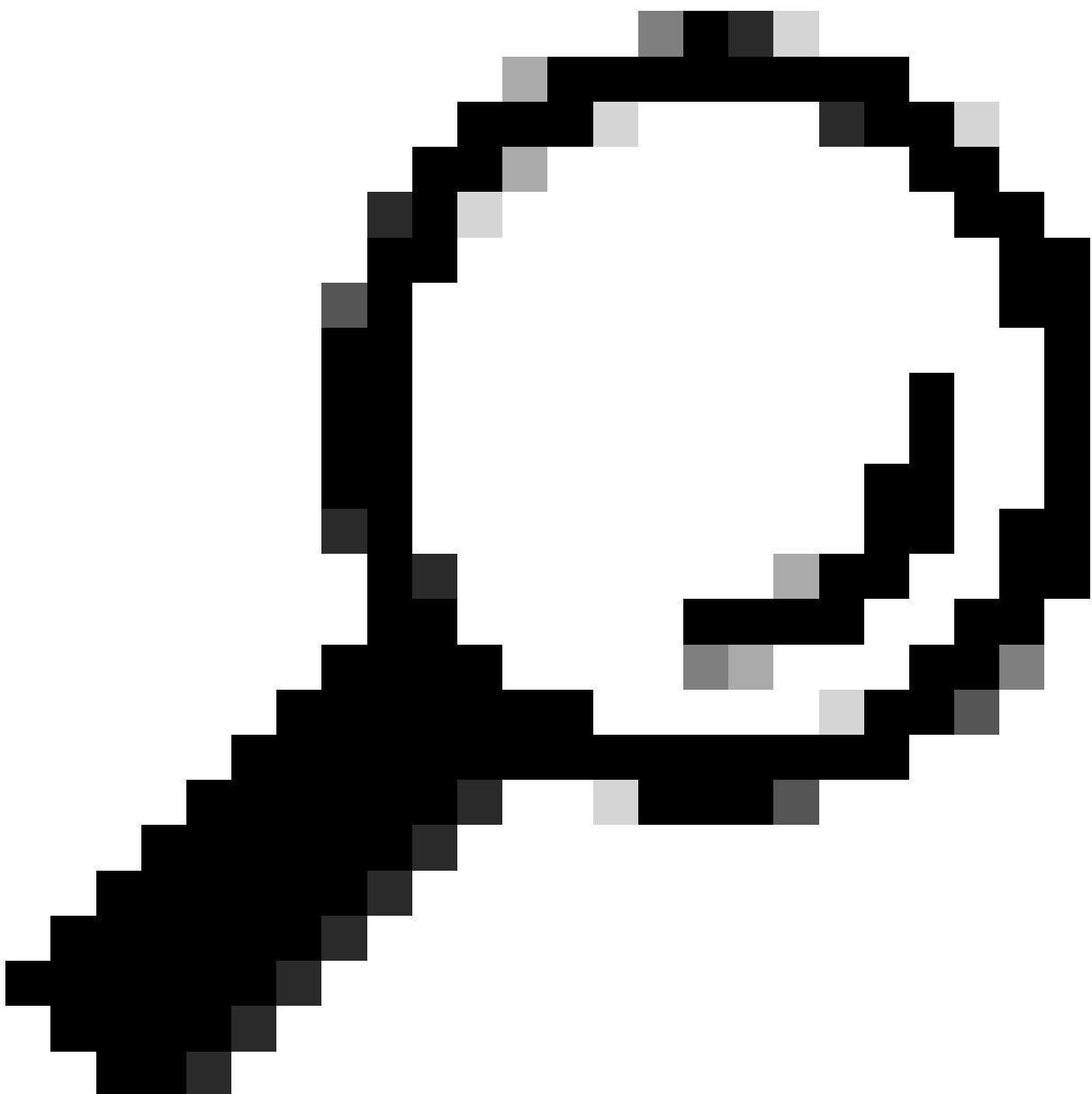
1:09:21 3
6.00 26.00 27.00
0.00 0.00
41.00

1:09:21 9

```
0.00      86.00      14.00
          0.00      0.00
0.00
```

열 머리글은 다음과 같이 요약됩니다.

- %user - 사용자 공간(응용 프로그램 코드)에서 CPU 사용량이 기본값인 nice입니다.
- %nice - 사용자 공간(응용 프로그램 코드)의 CPU 사용량이 양수 nice 값입니다.
- %system - 커널 공간의 CPU 사용량입니다.%idle - 나머지 필드를 100% 뺀 값입니다.



팁: 특정 코어에서 실행 중인 스레드 목록을 찾으려면 이 명령을 실행합니다.
`pidstat -h -t -p 모두 1 | awk '($10=="%CPU") || ($10==XX || $10==YY) && $9+0 > 10.00)`

expert 모드에서 top 명령을 실행하여 특정 코어에서 실행 중인 프로세스 목록을 찾습니다.

3. 프로세스가 다중 스레드인지 확인합니다.

<#root>

[admin@firepower](#)

:

~\$

sudo pidstat -h -t -p \$(pidof snort3) 1

```
Linux 4.18.45-yocto-standard (firepower)      03/03/25      _x86_64_      (64 CPU)

# Time      UID        TGID        TID      %usr    %system    %guest    %wait    %CPU    CPU    Command
4:20:05      0         4162         -    279.08    17.17      0.00      0.99    285.15    3    snort3 <--

4:20:05      0          -         4162     3.96      0.99      0.00      0.99      4.95     3    |__
snort3

4:20:05      0          -         4165     0.00      0.00      0.00      0.00      0.00     7    |__
snort3

4:20:05      0          -         4167     0.00      0.00      0.00      0.00      0.00     9    |__
snort3

4:20:05      0          -         4168     0.00      0.00      0.00      0.00      0.00     9    |__
snort3

4:20:05      0          -         4169     0.00      0.00      0.00      0.00      0.00     9    |__snort3
4:20:05      0          -         4172     0.00      0.00      0.00      0.00      0.00     9    |__snort3
4:20:05      0          -         4175     0.00      0.00      0.00      0.00      0.00     9    |__snort3
4:20:05      0          -         4177     0.00      0.00      0.00      0.00      0.00     3    |__snort3
```

<#root>

admin@firepower

:

~\$

sudo

pidstat -h -t -p \$(pidof rsyslog) 1

```
Linux 4.18.45-yocto-standard (firepower)      03/03/25      _x86_64_      (64 CPU)

# Time      UID      TGID      TID      %usr %system %guest  %wait  %CPU  CPU  Command
4:22:47      0        1        -      0.00  0.00  0.00   0.00   0.00  32  init
4:22:47      0        -        1      0.00  0.00  0.00   0.00   0.00  32  |__init <---- single
```

4. 프로세스 스레드가 시스템 코어에서 실행되는지 확인합니다.

<#root>

admin@firepower:~\$

sudo pidstat -h -t -p \$(pidof EventHandler) 1

Password:

```
Linux 4.18.45-yocto-standard (firepower)      03/03/25      _x86_64_      (64 CPU)

# Time      UID      TGID      TID      %usr  %system %guest  %wait  %CPU  CPU  Command
04:46:01      0      327455      -      61.00  23.00  0.00   0.00   83.00  3

EventHandler
```

<--- All threads run on system cores

```
04:46:01      0        -      327455      0.00   0.00  0.00   0.00   0.00  3  |__Event
04:46:01      0        -      327456      30.00   0.00  0.00   60.00  40.00  3  |__Event
04:46:01      0        -      327457      29.00   9.00  0.00   64.00  33.00  9  |__Event
04:46:01      0        -      327458      0.00   0.00  0.00   0.00   0.00  3  |__Event
04:46:01      0        -      327459      0.00   0.00  0.00   0.00   0.00  3  |__zmqio
04:46:01      0        -      327465      0.00   0.00  0.00   0.00   0.00  9  |__zmqio
04:46:01      0        -      327466      0.00   0.00  0.00   1.00   0.00  9  |__Event
04:46:01      0        -      327468      3.00   2.00  0.00   3.00   5.00  9  |__Event
04:46:01      0        -      327470      0.00   0.00  0.00   0.00   0.00  9  |__Event
04:46:01      0        -      327475      0.00   0.00  0.00   0.00   0.00  3  |__Event
04:46:01      0        -      327476      0.00   0.00  0.00   0.00   0.00  9  |__Event
04:46:01      0        -      327477      0.00   0.00  0.00   0.00   0.00  3  |__Event
04:46:01      0        -      327478      0.00   0.00  0.00   0.00   0.00  9  |__zmqio
04:46:01      0        -      327479      0.00   0.00  0.00   0.00   0.00  9  |__zmqio
```

5. 확인모든 스레드가 시스템 코어에서 CPU 사용량이 많아집니다.

<#root>

[admin@firepower:~\\$](#)

sudo pidstat -h -t -p \$(pidof EventHandler) 1

```
Linux 4.18.45-yocto-standard (firepower)      03/03/25      _x86_64_      (64 CPU)

# Time      UID      TGID      TID      %usr  %system %guest  %wait  %CPU  CPU  Command
05:13:30      0      327455      -     111.77  19.80  0.00   0.00  136.57  3  EventHa
```

<---- Process CPU usage = Σ (threads %CPU).

05:13:30	0	-	327455	0.00	0.00	0.00	0.00	0.00	3	__Event
05:13:30	0	-	327456	33.64	0.00	0.00	65.17	37.54	9	__Event

<---- Thread on system cores with highest CPU usage

05:13:30	0	-	327457	74.24	15.83	0.00	7.92	94.18	3	__Event
----------	---	---	--------	-------	-------	------	------	-------	---	---------

<---- Thread on system cores with highest CPU usage

05:13:30	0	-	327458	0.00	0.00	0.00	0.00	0.00	3	__Event
05:13:30	0	-	327459	0.00	0.00	0.00	0.00	0.00	9	__zmqi
05:13:30	0	-	327465	0.00	0.00	0.00	0.00	0.00	9	__zmqi
05:13:30	0	-	327466	0.00	0.99	0.00	0.00	0.99	3	__Event
05:13:30	0	-	327468	1.98	2.97	0.00	5.94	4.85	9	__Event
05:13:30	0	-	327470	0.00	0.00	0.00	0.99	0.00	3	__Event
05:13:30	0	-	327475	0.00	0.00	0.00	0.00	0.00	9	__Event
05:13:30	0	-	327476	0.00	0.00	0.00	0.00	0.00	9	__Event
05:13:30	0	-	327477	0.00	0.00	0.00	0.00	0.00	9	__Event
05:13:30	0	-	327478	0.00	0.00	0.00	0.00	0.00	3	__zmqi
05:13:30	0	-	327479	0.00	0.00	0.00	0.00	0.00	9	__zmqi

- EventHandler는 시스템 코어에서 실행 중인 다중 스레드 CPU입니다.
- 스레드 TID=327456 및 TID=327457이 CPU 사용량이 가장 높습니다.
- 프로세스 CPU 사용량은 모든 스레드의 CPU 사용량의 합계와 같습니다.

관련 정보

[Cisco Secure Firewall Management Center 관리 가이드, 7.6](#)

[Cisco Secure Firewall Management Center Device Configuration Guide, 7.6](#)

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.