Catalyst 9000 Series 스위치에서 Netflow, AVC 및 ETA 구성 및 확인

목차

소개

이 문서에서는 NetFlow, AVC(Application Visibility and Control), ETA(Encrypted Traffic Analytics)를 구성하고 검증하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Netflow
- AVC
- 에타

사용되는 구성 요소

이 문서의 정보는 Cisco IOS XE 소프트웨어 16.12.4를 실행하는 Catalyst 9300 스위치를 기반으로 합니다.

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

관련 제품

이 문서는 다음 하드웨어 및 소프트웨어 버전에서도 사용할 수 있습니다.

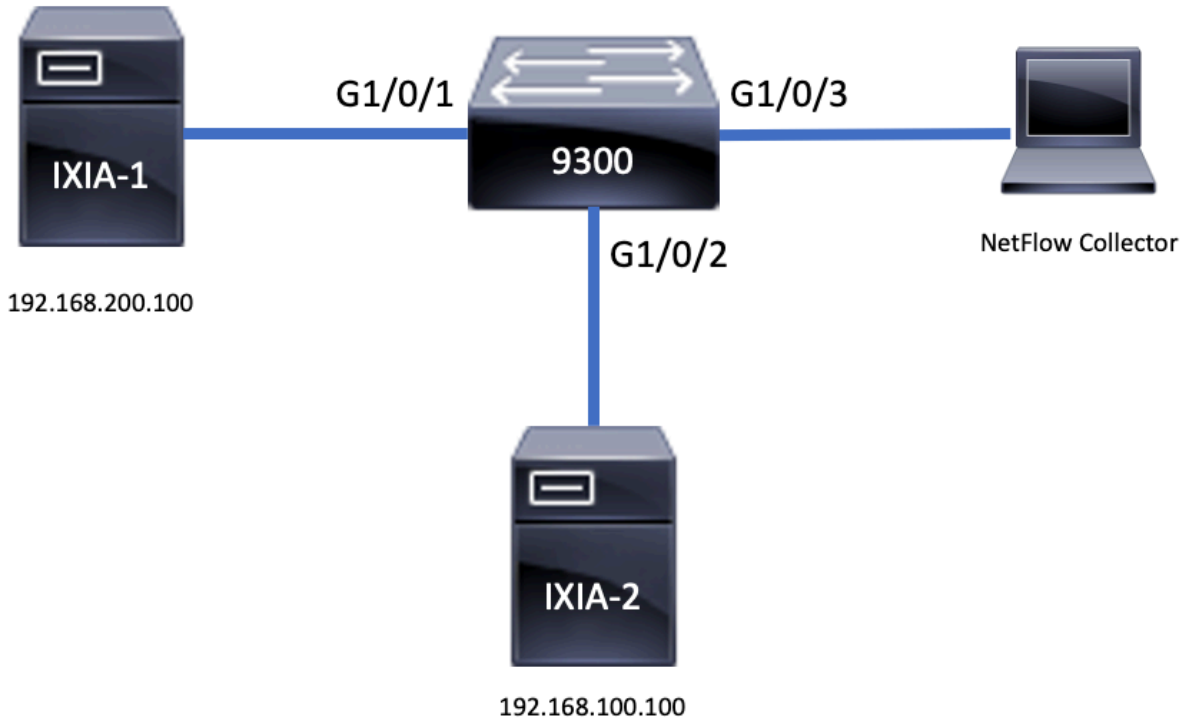
- 9200
- 9400
- 9500
- 9600
- Cisco IOS XE 16.12 이상

배경 정보

- Flexible NetFlow는 네트워크의 모든 라우터 또는 스위치가 원격 분석 소스가 될 수 있도록 데이터를 수집하고 측정하는 차세대 플로우 기술입니다.
- Flexible NetFlow는 매우 세분화되고 정확한 트래픽 측정 및 높은 수준의 종합적인 트래픽 수집을 지원합니다.
- Flexible NetFlow는 플로우를 사용하여 어카운팅, 네트워크 모니터링 및 네트워크 계획에 대한 통계를 제공합니다.
- 흐름은 소스 인터페이스에 도착하고 키에 대해 동일한 값을 가지는 패킷의 단방향 스트림입니다. 키는 패킷 내의 필드에 대해 식별된 값입니다. 플로우 레코드를 통해 플로우를 생성하여 플로우의 고유 키를 정의합니다.

참고: 플랫폼(fed) 명령은 다를 수 있습니다. 명령은 "show platform fed <active|standby>"와 "**show platform fed switch <active|standby>**"가 될 수 있습니다. 예제에 나온 구문이 구문 분석되지 않으면 variant를 사용해 보십시오.

네트워크 다이어그램



구성

구성 요소

NetFlow 컨피그레이션은 트래픽 분석 및 데이터 내보내기를 수행하기 위해 함께 사용할 수 있는 세 가지 주요 구성 요소로 구성됩니다.

플로우 레코드

- 레코드는 키 필드와 키 필드가 아닌 필드의 조합입니다. Flexible NetFlow Flow Records는 Flexible NetFlow Flow Monitor에 할당되어 플로우 데이터 저장에 사용되는 캐시를 정의합니다.
- Flexible NetFlow에는 트래픽을 모니터링하는 데 사용할 수 있는 몇 가지 미리 정의된 레코드가 포함되어 있습니다.
- 또한 Flexible NetFlow를 사용하면 키 필드와 키 필드가 아닌 필드를 지정하여 특정 요구 사항에 맞게 데이터 수집을 사용자 지정하여 Flexible NetFlow 흐름 모니터 캐시에 대해 사용자 지정 레코드를 정의할 수 있습니다.

예에 표시된 대로 흐름 레코드 컨피그레이션 세부사항은 다음과 같습니다.

```
flow record TAC-RECORD-IN
match flow direction
match ipv4 source address
match interface input
match ipv4 destination address
match ipv4 protocol
collect counter packets long
```

```
collect counter bytes long
collect timestamp absolute last
collect transport tcp flags

flow record TAC-RECORD-OUT
match flow direction
match interface output
match ipv4 source address
match ipv4 destination address
match ipv4 protocol
collect counter packets long
collect counter bytes long
collect timestamp absolute last
collect transport tcp flags
```

플로우 내보내기

- 플로우 내보내기는 플로우 모니터 캐시의 데이터를 분석 및 저장을 위해 원격 시스템(NetFlow 컬렉터 역할을 하는 서버)으로 내보내는 데 사용됩니다.
- 플로우 내보내기는 플로우 모니터에 할당되어 플로우 모니터에 대한 데이터 내보내기 기능을 제공합니다.

예에 표시된 대로 플로우 내보내기 컨피그레이션 세부사항은 다음과 같습니다.

```
flow exporter TAC-EXPORT
destination 192.168.69.2
source Vlan69
```

플로우 모니터

- 플로우 모니터는 네트워크 트래픽 모니터링을 수행하기 위해 인터페이스에 적용되는 Flexible NetFlow 구성 요소입니다.
- 플로우 데이터는 네트워크 트래픽에서 수집되며 프로세스가 실행되는 동안 플로우 모니터 캐시에 추가됩니다. 이 프로세스는 흐름 레코드의 키 필드와 키 필드가 아닌 필드를 기반으로 합니다.

예에 표시된 대로 플로우 모니터 컨피그레이션 세부사항은 다음과 같습니다.

```
flow monitor TAC-MONITOR-IN
exporter TAC-EXPORT
record TAC-RECORD-IN

flow monitor TAC-MONITOR-OUT
exporter TAC-EXPORT
record TAC-RECORD-OUT

Switch#show run int g1/0/1
Building configuration...

Current configuration : 185 bytes
!
interface GigabitEthernet1/0/1
switchport access vlan 42
switchport mode access
ip flow monitor TAC-MONITOR-IN input
ip flow monitor TAC-MONITOR-OUT output
load-interval 30
```

```
end
```

플로우 샘플러(선택 사항)

- 플로우 샘플러는 라우터 컨피그레이션에서 별도의 구성 요소로 생성됩니다.
- 플로우 샘플러는 Flexible NetFlow를 사용하는 디바이스의 부하를 줄이기 위해 분석을 위해 선택하는 패킷의 수를 제한합니다.
- 플로우 샘플러는 분석을 위해 선택한 패킷 수의 제한을 통해 달성되는 Flexible NetFlow를 사용하는 디바이스의 부하를 줄이는 데 사용됩니다.
- 플로우 샘플러는 라우터 성능을 위해 정확도를 교환합니다. 플로우 모니터에서 분석하는 패킷 수가 감소하면 플로우 모니터의 캐시에 저장된 정보의 정확성에 영향을 줄 수 있습니다.

예에 표시된 대로 플로우 샘플러 컨피그레이션의 예는 다음과 같습니다.

```
sampler SAMPLE-TAC
description Sample at 50%
mode random 1 out-of 2

Switch(config)#interface GigabitEthernet1/0/1
Switch(config-if)#ip flow monitor TAC-MONITOR-IN sampler SAMPLE-TAC input
Switch(config-if)#end
```

제한 사항

- 전체 Flexible NetFlow에는 DNA Addon 라이센스가 필요하며, 그렇지 않으면 샘플링된 NetFlow만 사용할 수 있습니다.
- 플로우 내보내기는 관리 포트를 소스로 사용할 수 없습니다.

이 목록은 포괄적 목록이 아닙니다. 해당 플랫폼 및 코드에 대해서는 컨피그레이션 가이드를 참조하십시오.

다음을 확인합니다.

플랫폼 독립적인 검증

구성을 확인하고 필요한 NetFlow 구성 요소가 있는지 확인합니다.

1. **플로우 레코드**
2. **플로우 내보내기**
3. **플로우 모니터**
4. **플로우 샘플러(선택 사항)**

> **팁:** 하나의 명령으로 흐름 레코드, 흐름 내보내기 및 흐름 모니터 출력을 보려면 "show running**-config flow monitor <flow monitor name> expand**"를 실행합니다.

이 예에 표시된 것처럼 흐름 모니터는 입력 방향과 관련 구성 요소에 연결됩니다.

```
Switch#show running-config flow monitor TAC-MONITOR-IN expand
Current configuration:
!
flow record TAC-RECORD-IN
 match ipv4 protocol
 match ipv4 source address
```

```
 match ipv4 destination address
 match interface input
 match flow direction
 collect transport tcp flags
 collect counter bytes long
 collect counter packets long
 collect timestamp absolute last
!
flow exporter TAC-EXPORT
 destination 192.168.69.2
 source Vlan69
!
flow monitor TAC-MONITOR-IN
 exporter TAC-EXPORT
 record TAC-RECORD-IN
!
```

이 예에 표시된 것처럼 흐름 모니터는 출력 방향과 관련 구성 요소에 연결됩니다.

```
Switch#show run flow monitor TAC-MONITOR-OUT expand
Current configuration:
!
flow record TAC-RECORD-OUT
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match interface output
 match flow direction
 collect transport tcp flags
 collect counter bytes long
 collect counter packets long
 collect timestamp absolute last
!
flow exporter TAC-EXPORT
 destination 192.168.69.2
 source Vlan69
!
flow monitor TAC-MONITOR-OUT
 exporter TAC-EXPORT
 record TAC-RECORD-OUT
!
```

"show flow monitor **<flow monitor name>" statistics 명령을 실행합니다.** 이 출력은 데이터가 기록되었는지 확인하는 데 유용합니다.

```
Switch#show flow monitor TAC-MONITOR-IN statistics
  Cache type:                               Normal (Platform cache)
  Cache size:                                10000
  Current entries:                               1

  Flows added:                                   1
  Flows aged:                                    0
```

다음 명령을 실행하여 **"show flow monitor <flow monitor name>** 캐시를 실행하여 NetFlow 캐시의 출력을 확인합니다.

```
Switch#show flow monitor TAC-MONITOR-IN cache
  Cache type:                               Normal (Platform cache)
  Cache size:                                10000
  Current entries:                               1
```

```
  Flows added:                                  1
  Flows aged:                                   0

IPV4 SOURCE ADDRESS:       192.168.200.100
IPV4 DESTINATION ADDRESS:  192.168.100.100
INTERFACE INPUT:           Gi1/0/1
FLOW DIRECTION:            Input
IP PROTOCOL:               17
tcp flags:                 0x00
counter bytes long:        4606617470
counter packets long:      25311085
timestamp abs last:        22:44:48.579
```

"show flow exporter <exporter name> statistics" 명령을 실행하여 내보내기가 패킷을 전송했는지 확인합니다.

```
Switch#show flow exporter TAC-EXPORT statistics
Flow Exporter TAC-EXPORT:
  Packet send statistics (last cleared 00:08:38 ago):
    Successfully sent:         2                     (24 bytes)

  Client send statistics:
    Client: Flow Monitor TAC-MONITOR-IN
      Records added:           0
      Bytes added:             12
        - sent:                12

    Client: Flow Monitor TAC-MONITOR-OUT
      Records added:           0
      Bytes added:             12
        - sent:                12
```

플랫폼에 따른 확인

NetFlow 초기화 - NFL 파티션 테이블

- NetFlow 파티션은 방향당 16개의 파티션이 있는 서로 다른 기능에 대해 초기화됩니다(입력 대 출력).
- NetFlow 파티션 테이블 컨피그레이션은 인그레스 및 이그레스 플로우 뱅크로 세분화되는 전역 뱅크 할당으로 나뉩니다.

주요 필드

- 파티션 수
- 파티션 사용 상태
- 파티션 제한
- 현재 파티션 사용량

NetFlow 파티션 테이블을 보려면 **"show platform software fed switch active|standby|member| fnf sw-table-size asic <asic number> shadow 0" 명령을** 실행할 수 있습니다.

> **참고:** 생성된 플로우는 스위치 및 기본 코어에 따라 생성되며, 이에 따라 스위치 번호(액티브, 스탠바이 등)를 지정해야 합니다. 입력된 ASIC 번호는 각 인터페이스에 연결되어 있으며 **"show platform software fed switch active|standby|member ifm mappings"를** 사용하여 인터페이스에 해당하는 ASIC를 결정합니다. Shadow 옵션에는 항상 "0"을 사용합니다.

```
Switch#show platform software fed switch active fnf sw-table-sizes asic 0 shadow 0

---------------------------------
     Global Bank Allocation
---------------------------------
Ingress Banks : Bank 0 Bank 1
Egress Banks  : Bank 2 Bank 3
---------------------------------
    Global flow table Info                                  <--- Provides the number of entries
used per direction
INGRESS    usedBankEntry           0  usedOvfTcamEntry     0
 EGRESS    usedBankEntry           0  usedOvfTcamEntry     0
---------------------------------
    Flows Statistics
INGRESS    TotalSeen=0 MaxEntries=0 MaxOverflow=0
EGRESS     TotalSeen=0 MaxEntries=0 MaxOverflow=0

---------------------------------
       Partition Table
---------------------------------
## Dir  Limit  CurrFlowCount  OverFlowCount  MonitoringEnabled
 0 ING      0              0              0                  0
 1 ING  16640              0              0                  1          <-- Current flow count in hardware
 2 ING      0              0              0                  0
 3 ING  16640              0              0                  0
 4 ING      0              0              0                  0
 5 ING   8192              0              0                  1
 6 ING      0              0              0                  0
 7 ING      0              0              0                  0
 8 ING      0              0              0                  0
 9 ING      0              0              0                  0
10 ING      0              0              0                  0
11 ING      0              0              0                  0
12 ING      0              0              0                  0
13 ING      0              0              0                  0
14 ING      0              0              0                  0
15 ING      0              0              0                  0
 0 EGR      0              0              0                  0
 1 EGR  16640              0              0                  1          <-- Current flow count in hardware
 2 EGR      0              0              0                  0
 3 EGR  16640              0              0                  0
 4 EGR      0              0              0                  0
 5 EGR   8192              0              0                  1
 6 EGR      0              0              0                  0
 7 EGR      0              0              0                  0
 8 EGR      0              0              0                  0
 9 EGR      0              0              0                  0
10 EGR      0              0              0                  0
11 EGR      0              0              0                  0
12 EGR      0              0              0                  0
13 EGR      0              0              0                  0
14 EGR      0              0              0                  0
15 EGR      0              0              0                  0
```

플로우 모니터

플로우 모니터 컨피그레이션에는 다음이 포함됩니다.

1. ACL TCAM 테이블 내의 항목을 생성하는 NetFlow ACL 컨피그레이션

ACL TCAM 항목은 다음으로 구성됩니다.

- 일치하는 키 조회
- 다음을 포함하는 NetFlow 조회에 사용되는 결과 매개변수
  프로필 IDNetFlow ID

2. NflLookupTable 및 NflFlowMaskTable에 항목을 만드는 플로우 마스크 구성

- NetFlow ACL 결과 매개변수로 인덱싱하여 netflow 조회를 위한 흐름 마스크 찾기

NetFlow ACL

NetFlow ACL 컨피그레이션을 보려면 "**show platform hardware fed switch active fwd-asic resource tcam table nfl_acl asic <asic number> 명령을 실행합니다.**

> **팁:** PACL(Port ACL)이 있는 경우 인터페이스가 매핑된 ASIC에 항목이 생성됩니다. RACL(Router ACL)의 경우 모든 ASIC에 항목이 존재합니다.

- 이 출력에는 4비트 값인 NFCMD0 및 NFCMD1이 있습니다. 프로파일 ID를 계산하기 위해 값을 이진으로 변환합니다.
- 이 출력에서 NFCMD0은 1이고 NFCMD1은 2입니다. 이진으로 변환하면 다음과 같습니다. 000100010
- Cisco IOS-XE 16.12 이상 8비트 조합에서 처음 4비트는 프로필 ID이며, 7비트는 조회가 활성화되었음을 나타냅니다. 00010010의 **예에서** 프로파일 ID는 1입니다.
- Cisco IOS XE 16.11 및 이전 버전의 코드에서는 8비트 조합 내에서 처음 6비트는 프로파일 ID이고 7비트는 조회가 활성화되었음을 나타냅니다. 이 예에서는 **00010010**이고 프로파일 ID는 4입니다.

```
Switch#show platform hardware fed switch active fwd-asic resource tcam table nfl_acl asic 0
Printing entries for region INGRESS_NFL_ACL_CONTROL (308) type 6 asic 0
=======================================================
Printing entries for region INGRESS_NFL_ACL_GACL (309) type 6 asic 0
=======================================================
Printing entries for region INGRESS_NFL_ACL_PACL (310) type 6 asic 0
=======================================================
TAQ-2 Index-32 (A:0,C:0) Valid StartF-1 StartA-1 SkipF-0 SkipA-0
Input IPv4 NFL PACL

Labels  Port   Vlan   L3If   Group
M:      00ff   0000   0000   0000
V:      0001   0000   0000   0000

   vcuResults l3Len l3Pro l3Tos SrcAddr  DstAddr  mtrid vrfid  SH
M:  00000000   0000  00    00  00000000 00000000  00   0000   0000
V:  00000000   0000  00    00  00000000 00000000  00   0000   0000

   RMAC RA MEn IPOPT MF NFF DF SO DPT TM DSEn  l3m
M:  0    0  0    0    0  0   0  0  0   0  0  0
V:  0    0  0    0    0  0   0  0  0   0  0  0

   SrcPort DstPortIITypeCode TCPFlags TTL ISBM QosLabel ReQOS S_P2P D_P2P
M:  0000    0000              00     00    0000    00       0     0     0
```

```
V:  0000    0000              00    00    0000    00      0     0     0

   SgEn SgLabel    AuthBehaviorTag  l2srcMiss  l2dstMiss ipTtl SgaclDeny
M:  0    000000    0  0   0 0     0
V:  0    000000    0  0   0 0     0

NFCMD0 NFCMD1 SMPLR LKP1 LKP2 PID QOSPRI MQLBL MPLPRO LUT0PRI CPUCOPY
     1      2     0    1    0   0      0     0      0 0x0000f       0
Start/Skip Word: 0x00000003
Start Feature, Terminate

----------------------------------------
Printing entries for region INGRESS_NFL_ACL_VACL (311) type 6 asic 0
========================================================
Printing entries for region INGRESS_NFL_ACL_RACL (312) type 6 asic 0
========================================================
Printing entries for region INGRESS_NFL_ACL_SSID (313) type 6 asic 0
========================================================
Printing entries for region INGRESS_NFL_CATCHALL (314) type 6 asic 0
========================================================
TAQ-2 Index-224 (A:0,C:0) Valid StartF-1 StartA-1 SkipF-0 SkipA-0
Input IPv4 NFL RACL

Labels  Port   Vlan   L3If   Group
M:      0000   0000   0000   0000
V:      0000   0000   0000   0000

   vcuResults l3Len l3Pro l3Tos SrcAddr  DstAddr  mtrid vrfid  SH
M:  00000000   0000  00    00  00000000 00000000  00   0000   0000
V:  00000000   0000  00    00  00000000 00000000  00   0000   0000

   RMAC RA MEn IPOPT MF NFF DF SO DPT TM DSEn  l3m
M:  0    0  0    0    0  0   0  0  0   0  0  0
V:  0    0  0    0    0  0   0  0  0   0  0  0

   SrcPort DstPortIITypeCode TCPFlags TTL ISBM QosLabel ReQOS S_P2P D_P2P
M:  0000    0000              00    00    0000    00      0     0     0
V:  0000    0000              00    00    0000    00      0     0     0

   SgEn SgLabel    AuthBehaviorTag  l2srcMiss  l2dstMiss ipTtl SgaclDeny
M:  0    000000    0  0   0 0     0
V:  0    000000    0  0   0 0     0

NFCMD0 NFCMD1 SMPLR LKP1 LKP2 PID QOSPRI MQLBL MPLPRO LUT0PRI CPUCOPY
     0      0     0    0    0   0      0     0      0 0x00000       0
Start/Skip Word: 0x00000003
Start Feature, Terminate

----------------------------------------
TAQ-2 Index-225 (A:0,C:0) Valid StartF-0 StartA-0 SkipF-0 SkipA-0
Input IPv4 NFL PACL

Labels  Port   Vlan   L3If   Group
M:      0000   0000   0000   0000
V:      0000   0000   0000   0000

   vcuResults l3Len l3Pro l3Tos SrcAddr  DstAddr  mtrid vrfid  SH
M:  00000000   0000  00    00  00000000 00000000  00   0000   0000
V:  00000000   0000  00    00  00000000 00000000  00   0000   0000

   RMAC RA MEn IPOPT MF NFF DF SO DPT TM DSEn  l3m
M:  0    0  0    0    0  0   0  0  0   0  0  0
V:  0    0  0    0    0  0   0  0  0   0  0  0
```

```
   SrcPort DstPortIITypeCode TCPFlags TTL ISBM QosLabel ReQOS S_P2P D_P2P
M:  0000    0000              00     00    0000    00       0     0     0
V:  0000    0000              00     00    0000    00       0     0     0

   SgEn SgLabel    AuthBehaviorTag  l2srcMiss  l2dstMiss ipTtl SgaclDeny
M:  0    000000    0  0   0 0     0
V:  0    000000    0  0   0 0     0

NFCMD0 NFCMD1 SMPLR LKP1 LKP2 PID QOSPRI MQLBL MPLPRO LUT0PRI CPUCOPY
     0      0     0    0    0   0      0     0      0 0x00000       0
Start/Skip Word: 0x00000000
No Start, Terminate

----------------------------------------
TAQ-2 Index-226 (A:0,C:0) Valid StartF-0 StartA-0 SkipF-0 SkipA-0
Input IPv6 NFL PACL

Labels  Port   Vlan   L3If   Group
Mask    0x0000 0x0000 0x0000 0x0000
Value   0x0000 0x0000 0x0000 0x0000

vcuResult dstAddr0 dstAddr1 dstAddr2 dstAddr3 srcAddr0
00000000  00000000 00000000 00000000 00000000 00000000
00000000  00000000 00000000 00000000 00000000 00000000

srcAddr1 srcAddr2 srcAddr3 TC HL l3Len fLabel vrfId toUs
00000000 00000000 00000000 00 00  0000  00000  000   0
00000000 00000000 00000000 00 00  0000  00000  000   0

l3Pro mtrId AE FE RE HE MF NFF SO IPOPT RA MEn RMAC DPT TMP l3m
 00    00   0  0  0  0  0   0  0    0   0   0   0    0   0    0
 00    00   0  0  0  0  0   0  0    0   0   0   0    0   0    0

DSE srcPort dstPortIITypeCode tcpFlags IIPresent cZId dstZId
 0   0000    0000       00       00      00      00
 0   0000    0000       00       00      00      00

v6RT AH ESP mREn ReQOS QosLabel PRole VRole  AuthBehaviorTag
M: 0   0   0    0    0       00          0     0 0
V: 0   0   0    0    0       00          0     0 0

   SgEn SgLabel
M:  0    000000
V:  0    000000

NFCMD0 NFCMD1 SMPLR LKP1 LKP2 PID QOSPRI MQLBL MPLPRO LUT0PRI CPUCOPY
     0      0     0    0    0   0      0     0      0 0x00000       0
Start/Skip Word: 0x00000000
No Start, Terminate

----------------------------------------
TAQ-2 Index-228 (A:0,C:0) Valid StartF-0 StartA-0 SkipF-0 SkipA-0
conversion to string vmr l2p not supported
----------------------------------------
TAQ-2 Index-230 (A:0,C:0) Valid StartF-0 StartA-0 SkipF-0 SkipA-0
Input MAC NFL PACL

Labels  Port   Vlan   L3If   Group
M:      0000   0000   0000   0000
V:      0000   0000   0000   0000

   arpSrcHwAddr  arpDestHwAddr  arpSrcIpAddr  arpTargetIp  arpOperation
M: 000000000000  000000000000   00000000       00000000     0000
V: 000000000000  000000000000   00000000       00000000     0000
```

```
   TRUST  SNOOP  SVALID  DVALID
M:     0      0       0       0
V:     0      0       0       0


   arpHardwareLength  arpHardwareType  arpProtocolLength  arpProtocolType
M:    00000000           00000000           00000000           00000000
V:    00000000           00000000           00000000           00000000


   VlanId l2Encap l2Protocol cosCFI   srcMAC      dstMAC    ISBM  QosLabel
M:  000    0         0000     0     000000000000 000000000000 00     00
V:  000    0         0000     0     000000000000 000000000000 00     00


  ReQOS isSnap  isLLC  AuthBehaviorTag
M:  0      0       0        0
V:  0      0       0         0

NFCMD0 NFCMD1 SMPLR LKP1 LKP2 PID QOSPRI MQLBL MPLPRO LUT0PRI CPUCOPY
     0       0     0    0    0   0       0     0       0 0x00000        0
Start/Skip Word: 0x00000000
No Start, Terminate

----------------------------------------
```

플로우 마스크

"show platform software fed switch active|standby|member fnf fmask-entry asic <asic number> entry 1" 명령을 실행하여 플로우 마스크가 하드웨어에 설치되어 있는지 확인합니다. 키 필드 목록의 수도 여기에서 확인할 수 있습니다.

```
Switch#show platform software fed switch active fnf fmask-entry asic 1 entry 1

---------------------------------
mask0_valid : 1
Mask hdl0   : 1
Profile ID  : 0
Feature 0   : 148
Fmsk0 RefCnt: 1
Mask M1     :
[511:256] => :00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[255:000] => :FFFFFFFF 00000000 FFFFFFFF 03FF0000 00000000 00FF0000 00000000 C00000FF

Mask M2     :

Key Map     :

Source   Field-Id   Size   NumPFields   Pfields
 002       090       04        01       (0 1 1 1)
 002       091       04        01       (0 1 1 0)
 002       000       01        01       (0 1 0 7)
 000       056       08        01       (0 0 2 4)
 001       011       11        04       (0 0 0 1) (0 0 0 0) (0 1 0 6) (0 0 2 0)
 000       067       32        01       (0 1 12 0)
 000       068       32        01       (0 1 12 2)
```

플로우 통계 및 타임스탬프 오프로드 데이터

netflow 통계와 타임스탬프를 보려면 "show platform software fed switch active fnf flow-record asic <asic number> start-index <index number> num-flows <number of flows> 명령을 실행합니다.

```
Switch#show platform software fed switch active fnf flow-record asic 1 start-index 1 num-flows 1
1 flows starting at 1 for asic 1:-------------------------------------------------
Idx 996 :
{90, ALR_INGRESS_NET_FLOW_ACL_LOOKUP_TYPE1 = 0x01}
{91, ALR_INGRESS_NET_FLOW_ACL_LOOKUP_TYPE2 = 0x01}
{0, ALR_INGRESS_NFL_SPECIAL1 = 0x00}
{56, PHF_INGRESS_L3_PROTOCOL = 0x11}
{11 PAD-UNK = 0x0000}
{67, PHF_INGRESS_IPV4_DEST_ADDRESS = 0xc0a86464}
{68, PHF_INGRESS_IPV4_SRC_ADDRESS = 0xc0a8c864}
FirstSeen = 0x4b2f, LastSeen = 0x4c59, sysUptime = 0x4c9d
PKT Count = 0x000000000102d5df, L2ByteCount = 0x00000000ca371638
```

```
Switch#show platform software fed switch active fnf flow-record asic 1 start-index 1 num-flows 1
1 flows starting at 1 for asic 1:-------------------------------------------------
Idx 996 :
{90, ALR_INGRESS_NET_FLOW_ACL_LOOKUP_TYPE1 = 0x01}
{91, ALR_INGRESS_NET_FLOW_ACL_LOOKUP_TYPE2 = 0x01}
{0, ALR_INGRESS_NFL_SPECIAL1 = 0x00}
{56, PHF_INGRESS_L3_PROTOCOL = 0x11}
{11 PAD-UNK = 0x0000}
{67, PHF_INGRESS_IPV4_DEST_ADDRESS = 0xc0a86464}
{68, PHF_INGRESS_IPV4_SRC_ADDRESS = 0xc0a8c864}
FirstSeen = 0x4b2f, LastSeen = 0x4c5b, sysUptime = 0x4c9f
PKT Count = 0x0000000001050682, L2ByteCount = 0x00000000cbed1590
```

AVC(Application Visibility and Control)

배경 정보

- AVC(Application Visibility and Control)는 NBAR2(Network-Based Recognition Version 2), **NetFlow V9**, 다양한 보고서 및 관리 툴(**Cisco Prime**)을 활용하여 DPI(Deep Packet Inspection)를 통해 애플리케이션을 분류하는 데 도움을 주는 솔루션입니다.
- AVC는 독립형 스위치 또는 스위치 스택용 유선 액세스 포트에서 구성할 수 있습니다.
- AVC는 Cisco Wireless Controller에서 DPI를 기반으로 애플리케이션을 식별한 다음 특정 DSCP 값으로 표시할 수도 있습니다. 또한 애플리케이션 및 클라이언트 측면에서 대역폭 사용량과 같은 다양한 무선 성능 메트릭을 수집할 수 있습니다.

성능 및 확장

성능: 각 스위치 멤버는 CPU 사용률 50% 미만으로 CPS(Connections Per Second) 500개를 처리할 수 있습니다. 이 속도 이상에서는 AVC 서비스가 보장되지 않습니다.

확장: 액세스 포트 24개당 최대 5000개의 양방향 흐름을 처리할 수 있습니다(액세스 포트당 약 200개의 흐름).

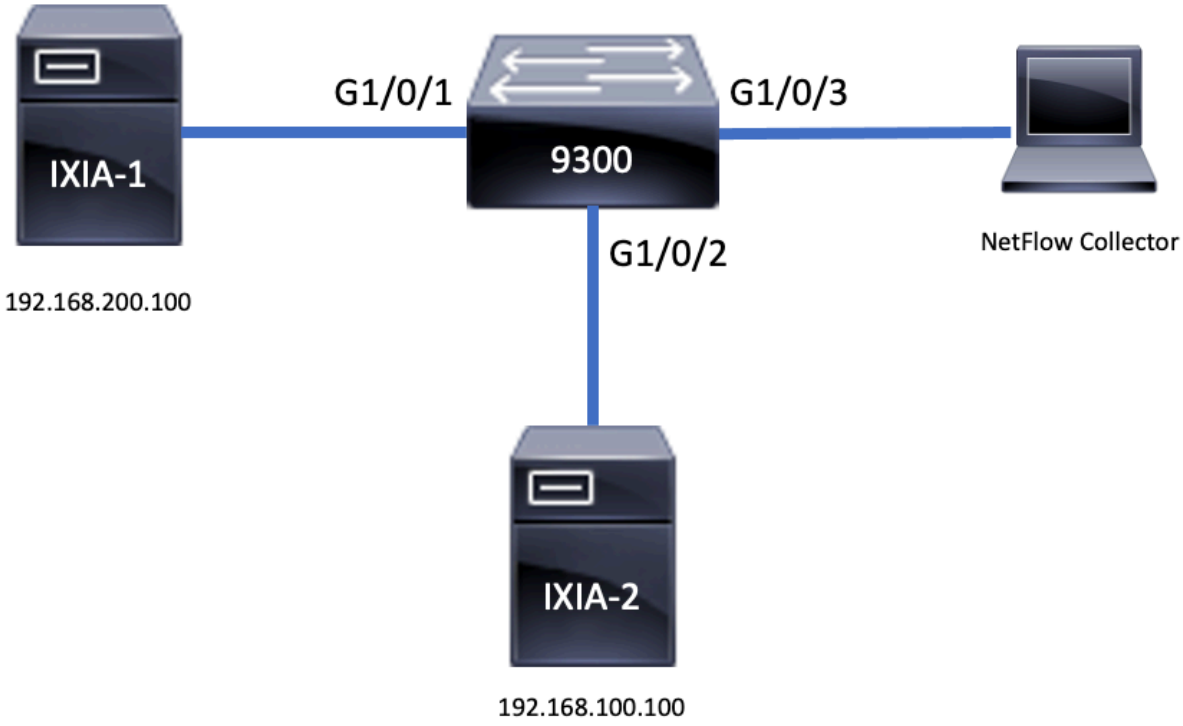
유선 AVC 제한

- AVC와 ETA(Encrypted Traffic Analytics)는 동일한 인터페이스에서 동시에 구성할 수 없습니다
  .

- 패킷 분류는 유니캐스트 IPv4(TCP/UDP) 트래픽에만 지원됩니다.
- NBAR 기반 QoS 정책 컨피그레이션은 유선 물리적 포트에서만 지원됩니다. 여기에는 레이어 2 액세스 및 트렁크 포트와 레이어 3 라우팅 포트가 포함됩니다.
- NBAR 기반 QoS 정책 컨피그레이션은 포트 채널 멤버, SVI(Switch Virtual Interface) 또는 하위 인터페이스에서 지원되지 않습니다.
- NBAR2 기반 분류자**(match protocol)는** 마킹 및 폴리싱의 QoS 작업만 지원합니다.
- "일치 프로토콜"은 모든 정책에서 255개의 서로 다른 프로토콜로 제한됩니다(8비트 하드웨어 제한).

참고: 이 모든 제한 사항은 완전한 목록이 아닙니다. 사용 중인 플랫폼 및 코드 버전에 맞는 AVC 컨피그레이션 가이드를 참조하십시오.

네트워크 다이어그램



구성 요소

AVC 컨피그레이션은 솔루션을 구성하는 **세 가지 주요** 구성 요소로 구성됩니다.

가시성: 프로토콜 검색

- 프로토콜 검색은 인터페이스당 방향 및 애플리케이션 바이트/패킷 통계를 제공하는 NBAR를 통해 이루어집니다.
- 프로토콜 검색은 인터페이스 컨피그레이션을 통해 특정 인터페이스에 대해 활성화됩니다. **ip nbar protocol-discovery**

출력에 표시된 대로 프로토콜 검색을 활성화하는 방법은 다음과 같습니다.

```
Switch(config)#interface fi4/0/5
```

```
Switch(config-if)#ip nbar protocol-discovery
Switch(config-if)#exit

Switch#show run int fi4/0/5
Building configuration...

Current configuration : 70 bytes
!
interface FiveGigabitEthernet4/0/5
ip nbar protocol-discovery
end
```

제어: 애플리케이션 기반 QoS

AVC는 IP 주소 및 UDP/TCP 포트에서 일치하는 기존 QoS에 비해 애플리케이션 기반 QoS를 통해 더 세밀한 제어를 구현합니다. 이를 통해 애플리케이션을 일치시킬 수 있으며 마킹 및 폴리싱과 같은 QoS 작업을 통해 더 세밀한 제어를 제공할 수 있습니다.

- 플로우 단위가 아닌 집계된 트래픽에 대해 작업이 수행됩니다.
- 애플리케이션 기반 QoS는 클래스 맵을 만들고, 프로토콜을 일치시킨 다음, 정책 맵을 만드는 방식으로 구현됩니다.
- 애플리케이션 기반 QoS 정책이 인터페이스에 연결됩니다.

출력에 표시된 대로 애플리케이션 기반 QoS의 컨피그레이션 예는 다음과 같습니다.

```
Switch(config)#class-map WEBEX
Switch(config-cmap)#match protocol webex-media
Switch(config)#end

Switch(config)#policy-map WEBEX
Switch(config-pmap)#class WEBEX
Switch(config-pmap-c)#set dscp af41
Switch(config)#end

Switch(config)#interface fi4/0/5
Switch(config-if)#service-policy input WEBEX
Switch(config)#end

Switch#show run int fi4/0/5
Building configuration...

Current configuration : 98 bytes
!
interface FiveGigabitEthernet4/0/5
service-policy input WEBEX
ip nbar protocol-discovery
end
```

애플리케이션 기반 Flexible NetFlow

유선 AVC FNF는 두 가지 유형의 미리 정의된 플로우 레코드를 지원합니다. **레거시 양방향 흐름 레코드** 및 새로운 **방향 흐름 레코드**.

양방향 플로우 레코드는 클라이언트/서버 애플리케이션 통계를 추적합니다.

출력에 표시된 것처럼 양방향 흐름 레코드의 컨피그레이션 예입니다.

```
Switch(config)#flow record BIDIR-1
Switch(config-flow-record)#match ipv4 version
```

```
Switch(config-flow-record)#match ipv4 protocol
Switch(config-flow-record)#match application name
Switch(config-flow-record)#match connection client ipv4 address
Switch(config-flow-record)#match connection server ipv4 address
Switch(config-flow-record)#match connection server transport port
Switch(config-flow-record)#match flow observation point
Switch(config-flow-record)#collect flow direction
Switch(config-flow-record)#collect connection initiator
Switch(config-flow-record)#collect connection new-connections
Switch(config-flow-record)#collect connection client counter packets long
Switch(config-flow-record)#connection client counter bytes network long
Switch(config-flow-record)#collect connection server counter packets long
Switch(config-flow-record)#connection server counter bytes network long
Switch(config-flow-record)#collect timestamp absolute first
Switch(config-flow-record)#collect timestamp absolute last
Switch(config-flow-record)#end

Switch#show flow record BIDIR-1
flow record BIDIR-1:
Description: User defined
No. of users: 0
Total field space: 78 bytes
Fields:
match ipv4 version
match ipv4 protocol
match application name
match connection client ipv4 address
match connection server ipv4 address
match connection server transport port
match flow observation point
collect flow direction
collect timestamp absolute first
collect timestamp absolute last
collect connection initiator
collect connection new-connections
collect connection server counter packets long
collect connection client counter packets long
collect connection server counter bytes network long
collect connection client counter bytes network long
```

방향 레코드는 입력/출력에 대한 애플리케이션 통계입니다.

출력에 표시된 대로 입력 및 출력 방향 레코드의 컨피그레이션 예는 다음과 같습니다.

> **참고**: "match interface input" 명령은 입력 인터페이스에 대한 일치를 지정합니다. "match interface **output**" 명령은 출력 인터페이스에 대한 일치를 지정합니다. AVC 지원에는 "**match application name**" 명령이 필수입니다.

```
Switch(config)#flow record APP-IN
Switch(config-flow-record)#match ipv4 version
Switch(config-flow-record)#match ipv4 protocol
Switch(config-flow-record)#match ipv4 source address
Switch(config-flow-record)#match ipv4 destination address
Switch(config-flow-record)#match transport source-port
Switch(config-flow-record)#match transport destination-port
Switch(config-flow-record)#match interface input
Switch(config-flow-record)#match application name
Switch(config-flow-record)#collect interface output
Switch(config-flow-record)#collect counter bytes long
Switch(config-flow-record)#collect counter packets long
Switch(config-flow-record)#collect timestamp absolute first
```

```
Switch(config-flow-record)#collect timestamp absolute last
Switch(config-flow-record)#end

Switch#show flow record APP-IN
flow record APP-IN:
Description: User defined
No. of users: 0
Total field space: 58 bytes
Fields:
match ipv4 version
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
match interface input
match application name
collect interface output
collect counter bytes long
collect counter packets long
collect timestamp absolute first
collect timestamp absolute last

Switch(config)#flow record APP-OUT
Switch(config-flow-record)#match ipv4 version
Switch(config-flow-record)#match ipv4 protocol
Switch(config-flow-record)#match ipv4 source address
Switch(config-flow-record)#match ipv4 destination address
Switch(config-flow-record)#match transport source-port
Switch(config-flow-record)#match transport destination-port
Switch(config-flow-record)#match interface output
Switch(config-flow-record)#match application name
Switch(config-flow-record)#collect interface input
Switch(config-flow-record)#collect counter bytes long
Switch(config-flow-record)#collect counter packets long
Switch(config-flow-record)#collect timestamp absolute first
Switch(config-flow-record)#collect timestamp absolute last
Switch(config-flow-record)#end

Switch#show flow record APP-OUT
flow record APP-OUT:
Description: User defined
No. of users: 0
Total field space: 58 bytes
Fields:
match ipv4 version
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
match interface output
match application name
collect interface input
collect counter bytes long
collect counter packets long
collect timestamp absolute first
collect timestamp absolute last
```

플로우 내보내기

내보내기 매개변수를 정의하기 위한 플로우 내보내기를 생성합니다.

출력에 표시된 대로 플로우 엑스포터의 컨피그레이션 예는 다음과 같습니다.

```
Switch(config)#flow exporter AVC
Switch(config-flow-exporter)#destination 192.168.69.2
Switch(config-flow-exporter)#source vlan69
Switch(config-flow-exporter)#end

Switch#show run flow exporter AVC
Current configuration:
!
flow exporter AVC
destination 192.168.69.2
source Vlan69
!
```

플로우 모니터

플로우 모니터를 생성하여 플로우 레코드에 연결합니다.

출력에 표시된 대로 플로우 모니터의 컨피그레이션 예는 다음과 같습니다.

```
Switch(config)#flow monitor AVC-MONITOR
Switch(config-flow-monitor)#record APP-OUT
Switch(config-flow-monitor)#exporter AVC
Switch(config-flow-monitor)#end

Switch#show run flow monitor AVC-MONITOR
Current configuration:
!
flow monitor AVC-MONITOR
exporter AVC
record APP-OUT
```

인터페이스에 플로우 모니터 연결

서로 **다른 사전** 정의된 레코드로 최대 2개의 서로 다른 AVC 모니터를 인터페이스에 동시에 연결할 수 있습니다.

출력에 표시된 대로 플로우 모니터의 컨피그레이션 예는 다음과 같습니다.

```
Switch(config)#interface fi4/0/5
Switch(config-if)#ip flow monitor AVC-MONITOR out
Switch(config-if)#end

Switch#show run interface fi4/0/5
Building configuration...
Current configuration : 134 bytes
!
interface FiveGigabitEthernet4/0/5
ip flow monitor AVC-MONITOR output
service-policy input WEBEX
ip nbar protocol-discovery
end
```

NBAR2

NBAR2 Dynamic Hitless Protocol Pack 업그레이드

프로토콜 팩은 디바이스에서 Cisco 소프트웨어를 대체하지 않고 디바이스에서 NBAR2 프로토콜 지원을 업데이트하는 소프트웨어 패키지입니다. 프로토콜 팩에는 NBAR2에서 공식 지원하는 응용 프로그램에 대한 정보가 포함되어 있으며, 이러한 정보는 함께 컴파일되고 압축됩니다. 각 응용 프로그램에 대해 프로토콜 팩에는 응용 프로그램 서명 및 응용 프로그램 특성에 대한 정보가 포함됩니다. 각 소프트웨어 릴리스에는 기본 제공 프로토콜 팩이 번들로 포함되어 있습니다.

- NBAR2는 트래픽 또는 서비스 중단 없이 디바이스에서 소프트웨어 이미지를 수정할 필요 없이 프로토콜 패킷을 업데이트하는 방법을 제공합니다
- NBAR2 프로토콜 패킷은 Cisco Software Center에서 다음 URL에서 다운로드할 수 있습니다. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html

NBAR2 프로토콜 팩 업그레이드

새 프로토콜 팩을 설치하기 전에 모든 스위치의 플래시에 프로토콜 패킷을 복사해야 합니다. 새 프로토콜 팩을 로드하려면 "ip nbar protocol**-pack flash:<Pack Name> 명령을 사용합니다.**

NBAR2 업그레이드를 위해 스위치를 다시 로드할 필요는 없습니다.

출력에 표시된 대로 NBAR2 프로토콜 팩을 로드하는 방법의 컨피그레이션 예는 다음과 같습니다.

```
Switch(config)#ip nbar protocol-pack flash:newProtocolPack
```

내장 프로토콜 팩으로 되돌리려면 "default ip nbar protocol**-pack" 명령을 사용합니다**

출력에 표시된 대로 내장 프로토콜 팩으로 되돌리는 방법의 컨피그레이션 예입니다.

```
Switch(config)#default ip nbar protocol-pack
```

NBAR2 프로토콜 팩 정보 표시

프로토콜 팩 정보를 표시하려면 다음 명령을 사용하십시오.

- **ip nbar 버전 표시**
- **show ip nbar protocol-pack active detail**

출력에 표시된 대로 이러한 명령의 출력 예:

```
Switch#show ip nbar version
NBAR software version: 37
NBAR minimum backward compatible version: 37
NBAR change ID: 293126

Loaded Protocol Pack(s):
Name: Advanced Protocol Pack
Version: 43.0
Publisher: Cisco Systems Inc.
NBAR Engine Version: 37
State: Active

Switch#show ip nbar protocol-pack active detail
Active Protocol Pack:
Name: Advanced Protocol Pack
Version: 43.0
Publisher: Cisco Systems Inc.
NBAR Engine Version: 37
```

```
State: Active
```

NBAR2 맞춤형 애플리케이션

NBAR2는 맞춤형 애플리케이션을 식별하기 위한 맞춤형 프로토콜의 사용을 지원합니다. 사용자 지정 프로토콜은 NBAR2가 현재 지원하지 않는 프로토콜과 애플리케이션을 지원합니다.

여기에는 다음이 포함될 수 있습니다.

- 조직에 대한 특정 애플리케이션
- 지역에 맞는 애플리케이션

NBAR2에서는 **ip** nbar custom<myappname> 명령을 통해 애플리케이션을 수동으로 **사용자 지정할 수 있는 방법*을 제공합니다*.**

> **참고:** 맞춤형 애플리케이션이 내장형 프로토콜보다 우선

다양한 유형의 애플리케이션 맞춤화가 있습니다.

일반 프로토콜 사용자 지정

- HTTP
- SSL
- DNS

복합:여러 프로토콜 기반의 사용자 지정 -**server-name**

Layer3/Layer4 사용자 지정

- IPv4 주소
- DSCP 값
- TCP/UDP 포트
- 흐름 소스 또는 대상 방향

바이트 오프셋 : 페이로드의 특정 바이트 값을 기반으로 사용자 정의

HTTP 사용자 지정

HTTP 사용자 지정은 다음과 같은 HTTP 필드의 조합을 기반으로 할 수 있습니다.

- **cookie** - HTTP 쿠키
- **host** - 리소스가 포함된 오리진 서버의 호스트 이름
- **method** - HTTP 메서드
- **referrer** - 리소스 요청을 가져온 주소
- **url** - Uniform Resource Locator 경로
- **user-agent** - 요청을 전송하는 에이전트에서 사용하는 소프트웨어
- **version** - HTTP 버전
- **via** - HTTP via 필드

Selector ID 10의 HTTP 호스트 "*mydomain.com"을 사용하는 MYHTTP라는 사용자 지정 애플리케이션의 예.

```
Switch(config)#ip nbar custom MYHTTP http host *mydomain.com id 10
```

SSL 사용자 지정

사용자 지정은 SSL SNI(Server Name Indication) 또는 CN(Common Name)에서 추출된 정보를 통해 SSL 암호화 트래픽에 대해 수행할 수 있습니다.

SSL 고유 이름 "mydomain.com"과 선택기 ID 11을 사용하는 MYSSL이라는 사용자 지정 애플리케이션의 예.

```
Switch(config)#ip nbar custom MYSSL ssl unique-name *mydomain.com id 11
```

DNS 사용자 지정

NBAR2는 DNS 요청 및 응답 트래픽을 검사하고, DNS 응답을 애플리케이션과 상호 연결할 수 있습니다. DNS 응답에서 반환된 IP 주소는 캐시되어 해당 특정 애플리케이션과 관련된 이후의 패킷 흐름에 사용됩니다.

commandip nbar customapplication-*namednsdomain-nameidapplication-idis는* DNS 사용자 지정에 사용됩니다. 응용 프로그램을 확장하려면 commandip nbar customapplication-namedns domain-*namedns domain*-nameextensionsexisting-*application을 사용합니다*.

DNS 도메인 이름 "mydomain.com"과 선택기 ID 12를 사용하는 MYDNS라는 사용자 지정 애플리케이션의 예.

```
Switch(config)#ip nbar custom MYDNS dns domain-name *mydomain.com id 12
```

복합 사용자 지정

NBAR2는 HTTP, SSL 또는 DNS에 나타나는 도메인 이름을 기반으로 애플리케이션을 사용자 정의하는 방법을 제공합니다.

HTTP, SSL 또는 DNS 도메인 이름 "mydomain.com"과 선택기 ID 13을 사용하는 MYDOMAIN이라는 사용자 지정 애플리케이션의 예.

```
Switch(config)#ip nbar custom MYDOMAIN composite server-name *mydomain.com id 13
```

L3/L4 사용자 지정

Layer3/Layer4 사용자 지정은 패킷 튜플을 기반으로 하며 흐름의 첫 번째 패킷에서 항상 일치됩니다.

IP 주소 10.56.1.10 및 10.56.1.11, TCP 및 DSCP ef(선택기 ID 14)와 일치하는 사용자 지정 애플리케이션 LAYER4CUSTOM의 예.

```
Switch(config)#ip nbar custom LAYER4CUSTOM transport tcp id 14
Switch(config-custom)#ip address 10.56.1.10 10.56.1.11
Switch(config-custom)#dscp ef
Switch(config-custom)#end
```

맞춤형 애플리케이션 모니터링

사용자 지정 애플리케이션을 모니터링하려면 다음과 같은 show 명령을 사용합니다.

show ip nbar protocol-id | inc 사용자 지정

```
Switch#show ip nbar protocol-id | inc Custom
LAYER4CUSTOM              14            Custom
MYDNS                     12            Custom
MYDOMAIN                  13            Custom
MYHTTP                    10            Custom
MYSSL                     11            Custom
```

show ip nbar protocol-id CUSTOM_APP

```
Switch#show ip nbar protocol-id MYSSL
Protocol Name              id             type
-----------------------------------------------
MYSSL                     11            Custom
```

AVC 확인

AVC의 기능을 검증하기 위한 여러 단계가 있으며, 이 섹션에서는 명령과 출력 예를 제공합니다.

NBAR가 활성 상태인지 확인하려면 "show ip nbar control-**plane" 명령을** 실행할 수 있습니다

주요 영역:

- NBAR 상태는 올바른 **시나리오**에서 활성화되어야 합니다.
- NBAR 컨피그레이션 상태는 올바른 **시나리오**에서 준비되어야 합니다.

```
Switch#show ip nbar control-plane
NGCP Status:
============

graph sender info:
NBAR state is ACTIVATED
NBAR config send mode is ASYNC
NBAR config state is READY

NBAR update ID 3
NBAR batch ID ACK 3
NBAR last batch ID ACK clients 1 (ID: 4)
Active clients 1 (ID: 4)
NBAR max protocol ID ever 1935
NBAR Control-Plane Version: 37

<snip>
```

show platform software fed switch active|standby|member wdavc function wdavc_stile_cp_show_info_ui 명령을 사용하여 각 스위치 멤버가 활성 데이터 평면을 가지고 있는지 확인합니다.

DP가 활성화되어야 올바른 **시나리오**에서 TRUE여야 합니까?

```
Switch#show platform software fed switch active wdavc function wdavc_stile_cp_show_info_ui

Is DP activated : TRUE
MSG ID : 3
Maximum number of flows: 262144
```

```
Current number of graphs: 1
Requests queue state : WDAVC_STILE_REQ_QUEUE_STATE_UP
Number of requests in queue : 0
Max number of requests in queue (TBD): 1
Counters:
activate_msgs_rcvd : 1
graph_download_begin_msgs_rcvd : 3
stile_config_msgs_rcvd : 1584
graph_download_end_msgs_rcvd : 3
deactivate_msgs_rcvd : 0
intf_proto_disc_msgs_rcvd : 1
intf_attach_msgs_rcvd : 2
cfg_response_msgs_sent : 1593
num_of_handle_msg_from_fmanfp_events : 1594
num_of_handle_request_from_queue : 1594
num_of_handle_process_requests_events : 1594
```

"show platform software fed **switch active|standby|member wdavc flows 명령을 사용하여** 키 정보를 표시합니다.

```
Switch#show platform software fed switch active wdavc flows

CurrFlows=1, Watermark=1

IX |IP1 |IP2 |PORT1|PORT2|L3 |L4 |VRF |TIMEOUT|APP |TUPLE|FLOW |IS FIF |BYPASS|FINAL |#PKTS
|BYPASS
  | | | | |PROTO|PROTO|VLAN|SEC |NAME |TYPE |TYPE |SWAPPED | | | |PKT
-------------------------------------------------------------------------------------
1 |192.168.100.2 |192.168.200.2 |68 |67 |1 |17 |0 |360 |unknown |Full |Real Flow|Yes |True |True
|40 |40
```

주요 필드:

CurrFlows: AVC에서 추적하는 활성 흐름 수를 보여 줍니다.

워터마크: 지금까지 AVC에서 추적한 가장 많은 플로우 표시

시간 초과 초: 식별된 애플리케이션에 따른 비활성 시간 초과

앱 이름: 식별된 애플리케이션

흐름 유형: Real Flow는 인바운드 데이터의 결과로 생성된 것입니다. Pre Flow는 인바운드 데이터의 결과로 이 흐름이 생성되었음을 나타냅니다. 사전 플로우는 예상되는 미디어 플로우에 사용됩니다

튜플 유형: 실제 플로우는 항상 전체 튜플이며, 사전 플로우는 전체 튜플 또는 절반 튜플입니다.

우회: TRUE로 설정된 경우 이 플로우를 식별하기 위해 소프트웨어에서 더 이상 패킷이 필요하지 않음을 나타냅니다

최종: TRUE로 설정된 경우 이 흐름에 대해 응용 프로그램이 더 이상 변경되지 않음을 나타냅니다

PKT 우회: 최종 분류에 도달하기 위해 필요한 패킷 수

#패킷: 이 흐름에 대해 소프트웨어로 실제 전송된 패킷 수

현재 흐름에 대한 추가 세부 정보를 **보려면 "show platform software fed switch active wdavc function wdavc_ft_show_all_flows_seg_ui" 명령을** 사용할 수 있습니다.

```
Switch#show platform software fed switch active wdavc function wdavc_ft_show_all_flows_seg_ui
CurrFlows=1, Watermark=1

IX |IP1 |IP2 |PORT1|PORT2|L3 |L4 |VRF |TIMEOUT|APP |TUPLE |FLOW |IS FIF |BYPASS|FINAL |#PKTS
|BYPASS
| | | | |PROTO|PROTO|VLAN|SEC |NAME |TYPE |TYPE |SWAPPED | | | |PKT
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------
1 |192.168.100.2 |192.168.200.2|68 |67 |1 |17 |0 |360 |unknown |Full |Real Flow|Yes |True |True
|40 |40

SEG IDX |I/F ID |OPST I/F |SEG DIR |FIF DIR |Is SET |DOP ID |NFL HDL |BPS PND |APP PND |FRST TS
|LAST TS |BYTES |PKTS |TCP FLGS
-------------------------------------------------------------------------------------------------
-----------------------------------
0 |9 |---- |Ingress |True |True |0 |50331823 |0 |0 |177403000|191422000|24252524|70094 |0
```

주요 필드

I/F ID: 인터페이스 ID를 지정합니다

SEG DIR: 이그레스 방향의 인그레스(ingress)를 지정합니다.

FIF DIR: 이것이 흐름 개시자 방향인지 여부를 결정합니다.

NFL HDL: 하드웨어의 흐름 ID

하드웨어의 항목을 보려면 **"show platform software fed switch active fnf flow-record asic <number> start-index <number> num-flows <number of flows> 명령을 실행합니다.**

> **참고:** ASIC를 선택하기 위해 포트가 매핑되는 ASIC 인스턴스입니다. ASIC를 식별하려면 **"show platform software fed switch active|standby|member ifm mappings" 명령을 사용합니다** . 특정 플로우에 관심이 없는 경우 start-index를 "0"으로 설정할 수 있습니다. 그렇지 않으면 start-index 를 지정해야 합니다. num-flows의 경우 볼 수 있는 플로우의 수를 지정합니다(최대 10).

```
Switch#show platform software fed switch active fnf flow-record asic 3 start-index 0 num-flows 1
1 flows starting at 0 for asic 3:-------------------------------------------------
Idx 175 :
{90, ALR_INGRESS_NET_FLOW_ACL_LOOKUP_TYPE1 = 0x01}
{91, ALR_INGRESS_NET_FLOW_ACL_LOOKUP_TYPE2 = 0x01}
{0, ALR_INGRESS_NFL_SPECIAL1 = 0x00}
{11 PAD-UNK = 0x0000}
{94, PHF_INGRESS_DEST_PORT_OR_ICMP_OR_IGMP_OR_PIM_FIRST16B = 0x0043}
{93, PHF_INGRESS_SRC_PORT = 0x0044}
{67, PHF_INGRESS_IPV4_DEST_ADDRESS = 0xc0a8c802}
{68, PHF_INGRESS_IPV4_SRC_ADDRESS = 0xc0a86402}
{56, PHF_INGRESS_L3_PROTOCOL = 0x11}
FirstSeen = 0x2b4fb, LastSeen = 0x2eede, sysUptime = 0x2ef1c
PKT Count = 0x000000000001216f, L2ByteCount = 0x0000000001873006
```

데이터 경로에서 다양한 오류 및 경고 검색

"show platform software fed switch active|standby|member wdavc function

wdavc_ft_show_stats_ui" 명령을 사용합니다. | inc err|warn|잠재적인 플로우 테이블 오류 보기 실패
:

```
Switch#show platform software fed switch active wdavc function wdavc_ft_show_stats_ui | inc
err|warn|fail
Bucket linked exceed max error : 0
extract_tuple_non_first_fragment_warn : 0
ft_client_err_alloc_fail : 0
ft_client_err_detach_fail : 0
ft_client_err_detach_fail_intf_attach : 0
ft_inst_nfl_clock_sync_err : 0
ft_ager_err_invalid_timeout : 0
ft_intf_err_alloc_fail : 0
ft_intf_err_detach_fail : 0
ft_inst_err_unreg_client_all : 0
ft_inst_err_inst_del_fail : 0
ft_flow_seg_sync_nfl_resp_pend_del_warn : 0
ager_sm_cb_bad_status_err : 0
ager_sm_cb_received_err : 0
ft_ager_to_time_no_mask_err : 0
ft_ager_to_time_latest_zero_ts_warn : 0
ft_ager_to_time_seg_zero_ts_warn : 0
ft_ager_to_time_ts_bigger_curr_warn : 0
ft_ager_to_ad_nfl_resp_error : 0
ft_ager_to_ad_req_all_recv_error : 0
ft_ager_to_ad_req_error : 0
ft_ager_to_ad_resp_error : 0
ft_ager_to_ad_req_restart_timer_due_err : 0
ft_ager_to_flow_del_nfl_resp_error : 0
ft_ager_to_flow_del_all_recv_error : 0
ft_ager_to_flow_del_req_error : 0
ft_ager_to_flow_del_resp_error : 0
ft_consumer_timer_start_error : 0
ft_consumer_tw_stop_error : 0
ft_consumer_memory_error : 0
ft_consumer_ad_resp_error : 0
ft_consumer_ad_resp_fc_error : 0
ft_consumer_cb_err : 0
ft_consumer_ad_resp_zero_ts_warn : 0
ft_consumer_ad_resp_zero_pkts_bytes_warn : 0
ft_consumer_remove_on_count_zero_err : 0
ft_ext_field_ref_cnt_zero_warn : 0
ft_ext_gen_ref_cnt_zero_warn : 0
```

"show platform software fed switch active wdavc function wdavc_stile_stats_show_ui 명령을 활용합니다. | inc err" 잠재적인 NBAR 오류를 보려면

```
Switch#show platform software fed switch active wdavc function wdavc_stile_stats_show_ui | inc
err
find_flow_error : 0
add_flow_error : 0
remove_flow_error : 0
detach_fo_error : 0
is_forward_direction_error : 0
set_flow_aging_error : 0
ft_process_packet_error : 0
sys_meminfo_get_error : 0
```

패킷이 CPU에 클론되었는지 확인합니다.

명령을 활용합니다. **"show platform software fed switch active punt cpuq 21 | inc received"** - NBAR

처리를 위해 패킷이 CPU에 복제되었는지 확인합니다.

> **참고:** Lab에서 이 숫자는 증가하지 않았습니다.

```
Switch#show platform software fed switch active punt cpuq 21 | inc received
Packets received from ASIC : 63
```

CPU 혼잡 **파악**

혼잡 시 WDAVC 프로세스로 전송되기 전에 패킷을 삭제할 수 있습니다. **"show platform software fed switch active wdavc function fed_wdavc_show_ots_stats_ui" 명령을 사용하여** 다음을 검증합니다.

```
Switch#show platform software fed switch active wdavc function fed_wdavc_show_ots_stats_ui
OTS Limits
-------------------------------------------
ots_queue_max : 20000
emer_bypass_ots_queue_stress : 4000
emer_bypass_ots_queue_normal : 200
OTS Statistics
-------------------------------------------
total_requests : 40
total_non_wdavc_requests : 0
request_empty_field_data_error : 0
request_invalid_di_error : 0
request_buf_coalesce_error : 0
request_invalid_format_error : 0
request_ip_version_error : 0
request_empty_packet_error : 0
memory_allocation_error : 0
emergency_bypass_requests_warn : 0
dropped_requests : 0
enqueued_requests : 40
max_ots_queue : 0
```

> **팁:** punt drop 카운터를 지우려면 **"show platform software fed switch active wdavc function fed_wdavc_clear_ots_stats_ui" 명령을 사용합니다.**

확장 문제 파악

하드웨어에 사용 가능한 FNF 항목이 없는 경우 트래픽은 NBAR2 분류의 대상이 아닙니다. **"show platform software fed switch active fnf sw-table-sizes asic <number> shadow 0" 명령을 사용하여** 다음을 확인합니다.

> **참고:** 생성된 플로우는 스위치 및 기본 코어에 따라 생성되며, 이에 따라 스위치 번호(액티브, 스탠바이 등)를 지정해야 합니다. 입력된 ASIC 번호는 각 인터페이스에 연결되어 있으며 **"show platform software fed switch active|standby|member ifm mappings"를** 사용하여 인터페이스에 해당하는 ASIC를 결정합니다. Shadow 옵션에는 항상 "0"을 사용합니다.

```
Switch#show platform software fed switch active fnf sw-table-sizes asic 3 shadow 0

---------------------------------
Global Bank Allocation
---------------------------------
```

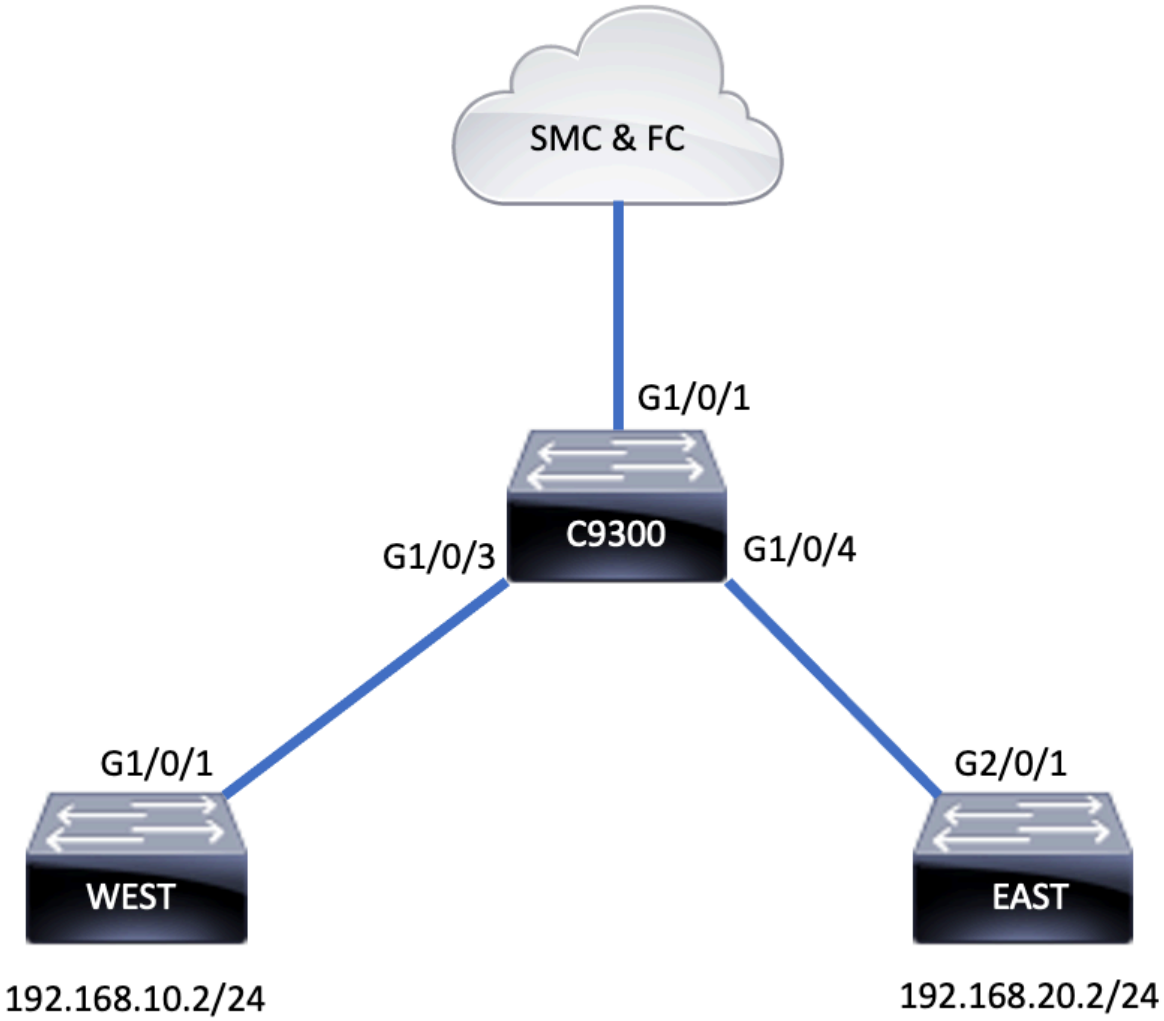
```
Ingress Banks : Bank 0
Egress Banks : Bank 1
--------------------------------
Global flow table Info
INGRESS usedBankEntry 1 usedOvfTcamEntry 0
EGRESS usedBankEntry 0 usedOvfTcamEntry 0 <-- 256 means TCAM entries are full
--------------------------------
Flows Statistics
INGRESS TotalSeen=1 MaxEntries=1 MaxOverflow=0
EGRESS TotalSeen=0 MaxEntries=0 MaxOverflow=0

--------------------------------
Partition Table
--------------------------------
## Dir Limit CurrFlowCount OverFlowCount MonitoringEnabled
0 ING 0 0 0 0
1 ING 16640 1 0 1
2 ING 0 0 0 0
3 ING 16640 0 0 0
4 ING 0 0 0 0
5 ING 8192 0 0 1
6 ING 0 0 0 0
7 ING 0 0 0 0
8 ING 0 0 0 0
9 ING 0 0 0 0
10 ING 0 0 0 0
11 ING 0 0 0 0
12 ING 0 0 0 0
13 ING 0 0 0 0
14 ING 0 0 0 0
15 ING 0 0 0 0
0 EGR 0 0 0 0
1 EGR 16640 0 0 1
2 EGR 0 0 0 0
3 EGR 16640 0 0 0
4 EGR 0 0 0 0
5 EGR 8192 0 0 1
6 EGR 0 0 0 0
7 EGR 0 0 0 0
8 EGR 0 0 0 0
9 EGR 0 0 0 0
10 EGR 0 0 0 0
11 EGR 0 0 0 0
12 EGR 0 0 0 0
13 EGR 0 0 0 0
14 EGR 0 0 0 0
15 EGR 0 0 0 0
```

암호화된 트래픽 분석(ETA)

배경 정보

- ETA는 클라우드 기반 글로벌 보안과 함께 패시브 모니터링, 관련 데이터 요소 추출, 동작 모델링 및 기계 학습의 조합을 통해 암호화된 트래픽에서 악성코드 통신을 식별하는 데 중점을 둡니다.
- ETA는 NetFlow의 텔레메트리, 암호화된 악성코드 탐지 및 암호화 규정 준수 기능을 활용하여 이 데이터를 Cisco Stealthwatch에 전송합니다.
- ETA는 두 가지 주요 데이터 요소를 추출합니다. IDP(Initial Data Packet) 및 SLT(Sequence of Packet Length and Time).

네트워크 다이어그램



구성 요소

ETA는 ETA 솔루션을 생성하는 데 함께 사용되는 여러 가지 구성 요소로 구성됩니다.

- NetFlow - 네트워크의 플로우를 설명하는 네트워크 디바이스에서 내보낸 데이터 요소를 정의하는 표준입니다.
- Cisco Stealthwatch - NetFlow, IPFIX, 프록시 로그 및 원시 패킷의 심층 패킷 검사를 포함하는 네트워크 텔레메트리의 기능을 활용하여 고급 네트워크 가시성, 보안 인텔리전스 및 분석을 제공합니다.
- Cisco Cognitive Intelligence - 보안 제어를 우회하거나 모니터링되지 않는 채널을 통해 조직 환경 내에 침입한 악의적인 활동을 찾아냅니다.
- Encrypted Traffic Analytics - 고급 동작 알고리즘을 사용하여 암호화된 트래픽의 전파 확산 메타데이터를 분석하여 악의적인 트래픽 패턴을 식별하고, 암호화된 트래픽에 숨겨진 잠재적 위협을 탐지하는 Cisco IOS XE 기능입니다.

참고: 이 문서에서는 Catalyst 9000 Series 스위치의 ETA 및 NetFlow에 대한 컨피그레이션 및 검증에만 초점을 맞추고 있으며, Cognitive Intelligence Cloud에 SMC(Stealthwatch Management Console) 및 FC(Flow Collector) 구축에 대해서는 다루지 않습니다.

제한 사항

- ETA를 구축하려면 DNA Advantage가 필요합니다.
- ETA 및 TX(Transmit) SPAN(Switched Port Analyzer)은 동일한 인터페이스에서 지원되지 않습니다.

포괄적 목록이 아닙니다. 모든 제한 사항에 대한 스위치 및 코드 버전에 대한 적절한 컨피그레이션 가이드를 참조하십시오.

설정

출력에 표시된 대로 스위치에 전역적으로 ETA를 활성화하고 흐름 내보내기 대상을 정의합니다.

```
C9300(config)#et-analytics
C9300(config-et-analytics)#ip flow-export destination 172.16.18.1 2055
```

팁: 포트 2055를 사용해야 하며 다른 포트 번호를 사용하지 마십시오.

그런 다음 출력에 표시된 대로 Flexible NetFlow를 구성합니다.

플로우 레코드 구성

```
C9300(config)#flow record FNF-RECORD
C9300(config-flow-record)#match ipv4 protocol
C9300(config-flow-record)#match ipv4 source address
C9300(config-flow-record)#match ipv4 destination address
C9300(config-flow-record)#match transport source-port
C9300(config-flow-record)#match transport destination-port
C9300(config-flow-record)#collect counter bytes long
C9300(config-flow-record)#collect counter packets long
C9300(config-flow-record)#collect timestamp absolute first
C9300(config-flow-record)#collect timestamp absolute last
```

플로우 모니터 구성

```
C9300(config)#flow exporter FNF-EXPORTER
C9300(config-flow-exporter)#destination 172.16.18.1
C9300(config-flow-exporter)#transport udp 2055
C9300(config-flow-exporter)#template data timeout 30
C9300(config-flow-exporter)#option interface-table
C9300(config-flow-exporter)#option application-table timeout 10
C9300(config-flow-exporter)#exit
```

플로우 레코드 구성

```
C9300(config)#flow monitor FNF-MONITOR
C9300(config-flow-monitor)#exporter FNF-EXPORTER
C9300(config-flow-monitor)#record FNF-RECORD
C9300(config-flow-monitor)#end
```

플로우 모니터 적용

```
C9300(config)#int range g1/0/3-4
C9300(config-if-range)#ip flow mon FNF-MONITOR in
C9300(config-if-range)#ip flow mon FNF-MONITOR out
```

```
C9300(config-if-range)#end
```

스위치 인터페이스에서 ETA 활성화

```
C9300(config)#interface range g1/0/3-4
C9300(config-if-range)#et-analytics enable
```

다음을 확인합니다.

ETA 모니터 "eta-mon" 모니터가 활성 상태인지 확인합니다. **"show flow monitor eta-mon" 명령을** 통해 상태가 할당되었는지 확인합니다.

```
C9300#show flow monitor eta-mon
Flow Monitor eta-mon:
Description: User defined
Flow Record: eta-rec
Flow Exporter: eta-exp
Cache:
Type: normal (Platform cache)
Status: allocated
Size: 10000 entries
Inactive Timeout: 15 secs
Active Timeout: 1800 secs
```

ETA 캐시가 채워져 있는지 확인합니다. NetFlow와 ETA가 동일한 인터페이스에 구성된 경우 "show flow monitor eta**-mon cache**"의 출력이 비어 있으므로 "**show flow monitor eta-mon cache**" 대신 "**show flow monitor** <monitor name**> cache"를** 사용합니다.

```
C9300#show flow monitor FNF-MONITOR cache
Cache type: Normal (Platform cache)
Cache size: 10000
Current entries: 4

Flows added: 8
Flows aged: 4
- Inactive timeout ( 15 secs) 4

IPV4 SOURCE ADDRESS: 192.168.10.2
IPV4 DESTINATION ADDRESS: 192.168.20.2
TRNS SOURCE PORT: 0
TRNS DESTINATION PORT: 0
IP PROTOCOL: 1
counter bytes long: 500
counter packets long: 5
timestamp abs first: 21:53:23.390
timestamp abs last: 21:53:23.390

IPV4 SOURCE ADDRESS: 192.168.20.2
IPV4 DESTINATION ADDRESS: 192.168.10.2
TRNS SOURCE PORT: 0
TRNS DESTINATION PORT: 0
IP PROTOCOL: 1
counter bytes long: 500
counter packets long: 5
timestamp abs first: 21:53:23.390
timestamp abs last: 21:53:23.390

IPV4 SOURCE ADDRESS: 192.168.20.2
IPV4 DESTINATION ADDRESS: 192.168.10.2
TRNS SOURCE PORT: 0
```

```
TRNS DESTINATION PORT: 0
IP PROTOCOL: 1
counter bytes long: 500
counter packets long: 5
timestamp abs first: 21:53:23.390
timestamp abs last: 21:53:23.390

IPV4 SOURCE ADDRESS: 192.168.10.2
IPV4 DESTINATION ADDRESS: 192.168.20.2
TRNS SOURCE PORT: 0
TRNS DESTINATION PORT: 0
IP PROTOCOL: 1
counter bytes long: 500
counter packets long: 5
timestamp abs first: 21:53:23.390
timestamp abs last: 21:53:23.390
```

"show flow exporter eta-exp statistics" 명령을 사용하여 SMC 및 FC로 **플로우를 내보내는지 확인합니다**.

```
C9300#show flow exporter eta-exp statistics
Flow Exporter eta-exp:
Packet send statistics (last cleared 03:05:32 ago):
Successfully sent: 3 (3266 bytes)

Client send statistics:
Client: Flow Monitor eta-mon
Records added: 4
- sent: 4
Bytes added: 3266
- sent: 3266
```

"show platform software fed switch active fnf et-analytics-flows" 명령을 사용하여 SPLT **및 IDP가 FC로 내보내지는지 확인합니다.**

```
C9300#show platform software fed switch active fnf et-analytics-flows

ET Analytics Flow dump

=================
Total packets received : 20
Excess packets received : 0
Excess syn received : 0
Total eta records added : 4
Current eta records : 0
Total eta splt exported : 2
Total eta IDP exported : 2
```

"show platform software et-analytics interfaces" 명령을 사용하여 **et-analytics에 대해 구성된 인터페이스를 확인합니다.**

```
C9300#show platform software et-analytics interfaces
ET-Analytics interfaces
GigabitEthernet1/0/3
GigabitEthernet1/0/4

ET-Analytics VLANs
```

ETA의 전역 상태를 보려면 **"show platform software et-analytics global"** 명령을 사용합니다.

```
C9300#show plat soft et-analytics global
ET-Analytics Global state
=========================
All Interfaces : Off
IP Flow-record Destination : 10.31.126.233 : 2055
Inactive timer : 15

ET-Analytics interfaces
GigabitEthernet1/0/3
GigabitEthernet1/0/4

ET-Analytics VLANs
```

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.