配置并验证Catalyst 9000系列交换机上的Netflow、AVC和ETA

目录

简介

本文档介绍如何配置和验证NetFlow、应用可视性与可控性(AVC)和加密流量分析(ETA)。

先决条件

要求

Cisco 建议您了解以下主题：

- Netflow
- AVC
- 埃塔

使用的组件

本文档中的信息基于运行Cisco IOS XE软件16.12.4的Catalyst 9300交换机。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

相关产品

本文档也可用于以下硬件和软件版本：

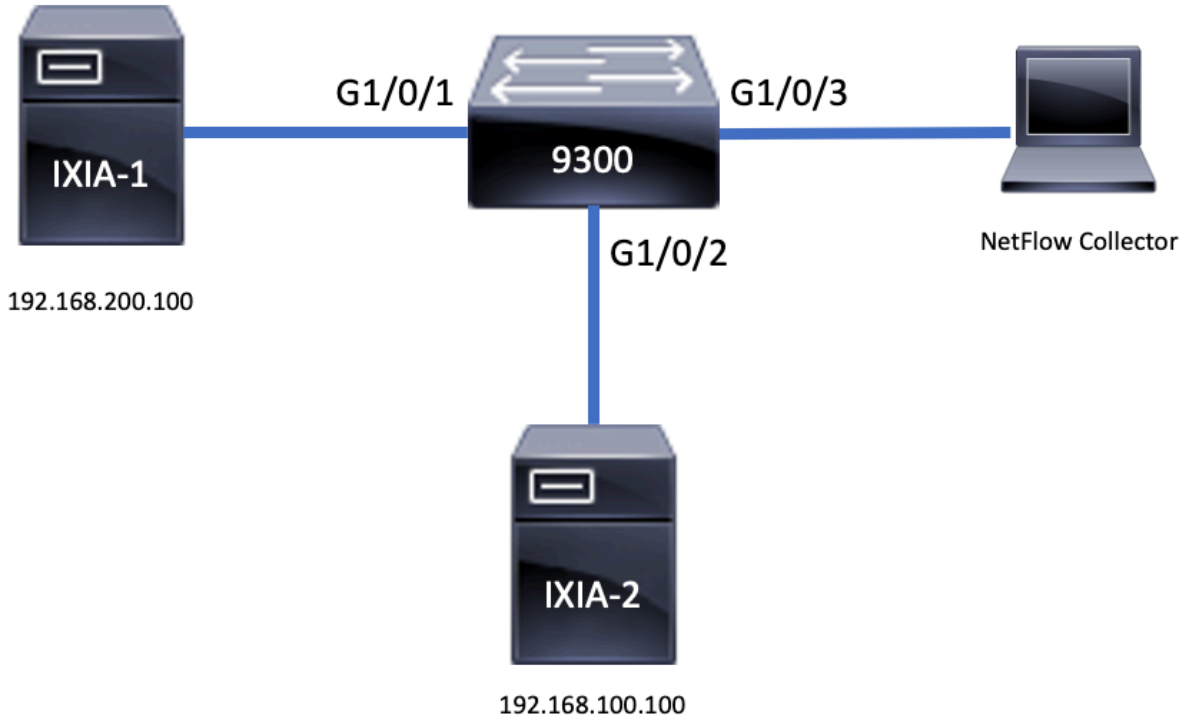
- 9200
- 9400
- 9500
- 9600
- Cisco IOS XE 16.12及更高版本

背景信息

- Flexible NetFlow是下一代流量技术，用于收集和测量数据，使网络中的所有路由器或交换机成为遥测源。
- Flexible NetFlow允许极其精细和准确的流量测量以及高级汇聚流量收集。
- Flexible NetFlow使用流为记帐、网络监控和网络规划提供统计信息。
- 流是到达源接口且具有相同的密钥值的单向数据包流。密钥是数据包中字段的标识值。通过流记录创建流，以定义流的唯一键。

注意：平台(fed)命令可能有所不同。命令可以是“**show platform fed <active|standby>**”和“**show platform fed switch <active|standby>**”。 如果示例中注明的语法未解析出，请尝试变体。

网络图



配置

组件

NetFlow配置由三个主要**组件组成，**这些组件可以一起使用多种变体，以执行流量分析和数据导出。

流记录

- 记录是键字段和非键字段的组合。将Flexible NetFlow记录分配给Flexible NetFlow流监控器，以定义用于存储流数据的缓存。
- Flexible NetFlow包括可用于监控流量的多个预定义记录。
- Flexible NetFlow还允许通过指定键和非键字段为Flexible NetFlow流监控器缓存定义自定义记录，以根据您的特定要求自定义数据收集。

如示例所示，流记录配置详细信息：

```
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
```

流导出器

- 流导出器用于将流监控器缓存中的数据导出到远程系统（充当NetFlow收集器的服务器）进行分析和存储。
- 流导出器分配给流监控器，以便为流监控器提供数据导出功能。

如示例所示，流导出器配置详细信息：

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

流量监控器

- 流监控器是Flexible NetFlow组件，应用于接口以执行网络流量监控。
- 流数据从网络流量收集，并在进程运行时添加到流监控器缓存。该流程基于流记录中的关键字段和非关键字段。

如示例所示，流量监控器配置详细信息：

```
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
```

流采样器（可选）

- 流量采样器作为路由器配置中的独立组件创建。
- 流采样器限制选择用于分析的数据包数量，以减少使用Flexible NetFlow的设备上的负载。
- 流量采样器用于减少使用通过限制选择用于分析的数据包数量实现的Flexible NetFlow的设备上的负载。
- 流量采样器可以交换路由器性能的准确性。如果流监控器分析的数据包数量减少，则存储在流

监控器缓存中的信息的准确性可能会受到影响。
如示例所示，流量采样器配置示例：

```
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
```

限制

- 完整的Flexible NetFlow需要DNA插件许可证，否则采样NetFlow仅可用。
- 流导出器无法使用管理端口作为源。

这不是包含列表，请查阅相应平台和代码的配置指南。

验证

独立于平台的验证

验证配置并确认所需的NetFlow组件存在：

1. **流记录**
2. **流导出器**
3. **流量监控器**
4. **流采样器（可选）**

   提示：若要在一个命令中查看流记录、流导出器和流监控器输出，请运行“**show running-config flow monitor <flow monitor name> expand**”

如示例所示，与输入方向相关的流量监控器及其关联组件：

```
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
!
```

如示例所示，与输出方向相关的流量监控器及其关联组件：

```
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
!
```

运行命令“show flow monitor <flow monitor name>” statistics。此输出有助于确认已记录数据：

```
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
```

运行命令show flow monitor <flow monitor name> cache以确认NetFlow缓存有输出：

```
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
```

运行命令show flow exporter <exporter name> statistics以确认导出器发送了数据包：

```
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
```

平台相关验证

NetFlow初始化 — NFL分区表

- NetFlow分区针对不同功能进行初始化，每个方向有16个分区（输入与输出）。
- NetFlow分区表配置分为全局银行分配，进一步细分为入口和出口流银行。

关键字段

- 参加者人数
- 分区启用状态
- 分区限制
- 当前分区使用情况

要查看NetFlow分区表，可以运行命令“show platform software fed switch active|standby|member| fnf sw-table-sizes asic <asic number> shadow 0”

> **注意**：创建的流特定于交换机和asic核心。需要相应地指定交换机编号（主用、备用等）。输入的ASIC编号绑定到各个接口，使用“show platform software fed switch active|standby|member ifm mappings”确定与接口对应的ASIC。对于阴影选项，请始终使用“0”。

```
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
```

流量监控器

流监控器配置包括：

1. NetFlow ACL配置，这会导致在ACL TCAM表中创建条目。

ACL TCAM条目包括：

- 查找匹配密钥
- 用于NetFlow查找的结果参数，其中包括：
  配置文件IDNetFlow ID

2.流掩码配置，这会导致NflLookupTable和NflFlowMaskTable中创建一个条目。

- 通过NetFlow ACL结果参数索引，以查找netflow查找的流掩码

NetFlow ACL

要查看NetFlow ACL配置，请运行命令“show platform hardware fed switch active fwd-asic resource tcam table nfl_acl asic <asic number>

提示：如果存在端口ACL(PACL)，则在接口映射到的ASIC上创建条目。对于路由器ACL(RACL)，条目存在于所有ASIC上。

- 在此输出中，有NFCMD0和NFCMD1，它们是4位值。要计算配置文件ID，请将值转换为二进制。
- 在此输出中，NFCMD0为1,NFCMD1为2。转换为二进制时：000100010
- 在Cisco IOS-XE 16.12中，在合并的8位内，前4位是配置文件ID，第7位表示查找已启用。因此，在示例**0001** 0010中，配置文件ID为1。
- 在Cisco IOS XE 16.11及更早版本的代码中，在组合的8位内，前6位是配置文件ID，第7位表示查找已启用。在本例中，**000100**10，配置文件ID为4。

```
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

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

流掩码

运行命令“show platform software fed switch active|standby|member fnf fmask-entry asic <asic number> entry 1”以查看流掩码是否安装在硬件中。关键字段列表的数量也可在此处找到。

```
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)
```

流统计数据和时间戳卸载数据

运行命令show platform software fed switch active fnf flow-record asic <asic number> start-index <index number> num-flows <number of flows> 以查看netflow统计信息和时间戳

```
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
```

应用可视性与可控性(AVC)

背景信息

- 应用可视性与可控性(AVC)解决方案利用基于网络的识别版本2(**NBAR2**)、**NetFlow V9**以及各种报告和管理工具(**Cisco Prime**)来帮助通过深度数据包检测(DPI)对应用进行分类。
- AVC可在独立交换机或交换机堆叠的有线接入端口上配置。
- AVC也可用于思科无线控制器，以根据DPI识别应用，然后将其标记为特定DSCP值。它还可以收集各种无线性能指标，例如应用和客户端方面的带宽使用情况。

性能和扩展

性能：每个交换机成员能够以低于50%的CPU利用率每秒处理500个连接(CPS)。在此速率之外，AVC服务没有保证。

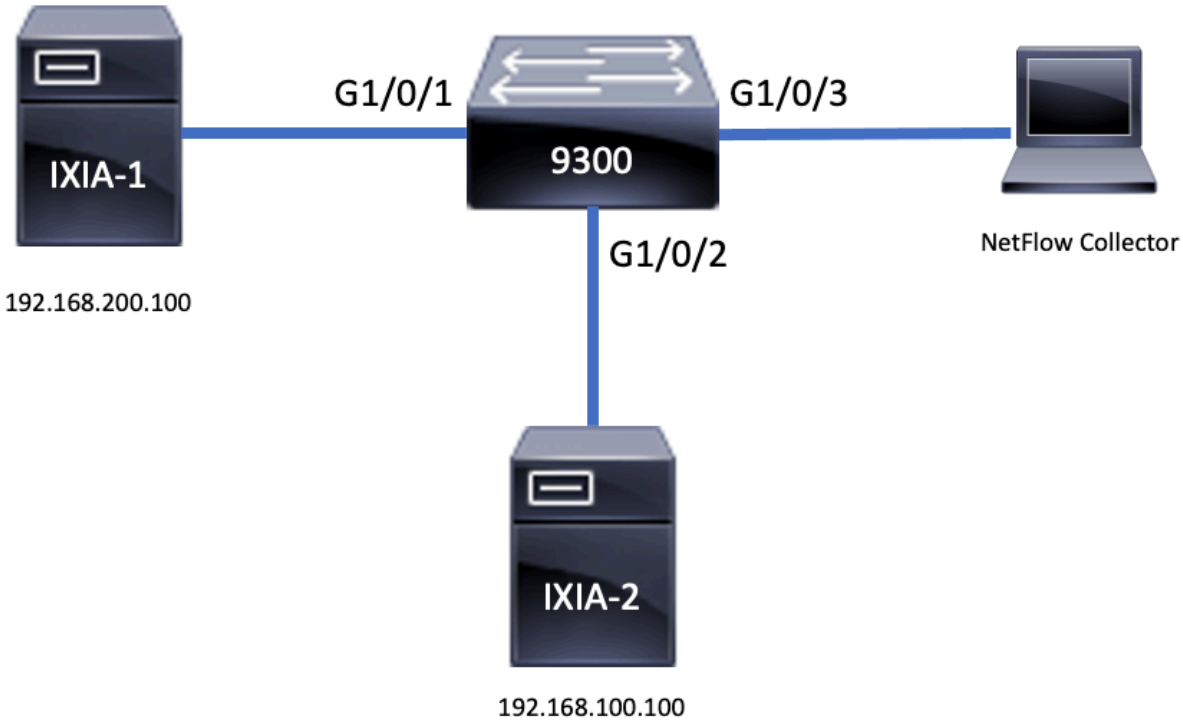
扩展：能够处理每24个接入端口多达5000个双向流（每个接入端口约200个流）。

有线AVC限制

- AVC和加密流量分析(ETA)不能在同一接口上同时配置。
- 仅单播IPv4(TCP/UDP)流量支持数据包分类。
- 只有有线物理端口支持基于NBAR的QoS策略配置。这包括第2层接入和中继端口以及第3层路由端口。
- 端口通道成员、交换机虚拟接口(SVI)或子接口不支持基于NBAR的QoS策略配置。
- 基于NBAR2的分**类器(匹配协议)**，仅支持标记和管制的QoS操作。
- 在所有策略中，“Match protocol”限制为255个不同的协议（8位硬件限制）

注意：这不是所有限制的详尽列表，请参考适用于您的平台和代码版本的相应AVC配置指南。

网络图



组件

AVC配置由构成解决方案的三个主要组件组成：

可视性：协议发现

- 协议发现通过NBAR实现，NBAR提供每个接口、方向和应用字节/数据包统计信息。
- 通过接口配置为特定接口启用协议发现:**ip nbar protocol-discovery**

如输出所示，如何启用协议发现：

```
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
```

控制：基于应用的QoS

与在IP地址和UDP/TCP端口上匹配的传统QoS相比，AVC通过基于应用的QoS实现更精细的控制，从而允许您在应用上进行匹配，并通过标记和管制等QoS操作提供更精细的控制。

- 对聚合流量（而不是每个流）执行操作
- 基于应用的QoS通过创建类映射、匹配协议以及创建策略映射来实现。
- 基于应用的QoS策略附加到接口。

如输出所示，基于应用的QoS的配置示例：

```
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
```

基于应用的Flexible NetFlow

有线AVC FNF支持两种类型的预定义流记录：**传统双向流记录**和新的方**向流记录**。

双向流记录用于跟踪客户端/服务器应用统计信息。

如输出所示，双向流记录的配置示例。

```
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
```

方向记录是用于输入/输出的应用统计信息。

如输出所示，输入和输出方向记录的配置示例：

> **注意**：命令“**match interface input**”指定与输入接口的匹配。命令“**match interface output**”指定输出接口的匹配项。命令“**match application name**”对于AVC支持是必需的。

```
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
```

流导出器

创建流导出器以定义导出参数。

如输出所示，流导出器的示例配置：

```
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
!
```

流量监控器

创建流监控器，将其关联到流记录。

如输出所示，流量监控器的配置示例：

```
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
```

将流监控器关联到接口

您最多可以将具有不同预定义记录的两个不同AVC监控器同时连接到一个接口。

如输出所示，流量监控器的配置示例：

```
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动态无中断协议包升级

协议包是更新设备上的NBAR2协议支持的软件包，无需替换设备上的Cisco软件。协议包包含由NBAR2正式支持的应用程序的信息，这些应用程序经过编译和打包。对于每个应用程序，协议包都包含有关应用程序签名和应用程序属性的信息。每个软件版本都随附一个内置协议包。

- NBAR2提供了一种更新协议数据包的方法，无需任何流量或服务中断，也无需修改设备上的软件映像
- NBAR2协议数据包可从以下URL下载到思科软件中心：https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html

NBAR2协议包升级

在安装新的协议包之前，必须将协议数据包复制到所有交换机的闪存中。 要加载新的协议包，请使用命令“ip nbar protocol-pack flash:<Pack Name>

无需重新加载交换机即可进行NBAR2升级。

如输出所示，有关如何加载NBAR2协议包的示例配置：

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

要恢复为内置协议包，请使用命令“default ip nbar protocol-pack”

如输出所示，有关如何恢复到内置协议包的示例配置：

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

显示NBAR2协议包信息

要显示协议包信息，请使用下列命令：

- show ip nbar version
- show ip nbar protocol-pack active detail

如输出所示，这些命令的输出示例：

```
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自定义应用

NBAR2支持使用自定义协议识别自定义应用。自定义协议支持NBAR2当前不支持的协议和应用程序。

这些可能包括：

- 组织特定应用
- 特定地理位置的应用

NBAR2通过命令ip nbar custom<*myappname*>提供手动自定义应用的方法。

> **注意**：自定义应用优先于内置协议

应用定制有多种类型：

通用协议定制

- HTTP
- SSL
- DNS

复合：基于多个协议的定制 — server-name

第3层/第4层定制

- IPv4地址
- DSCP 值
- TCP/UDP端口
- 流源或目标方向

字节偏移：根据负载中的特定字节值进行自定义

HTTP自定义

HTTP自定义可以基于以下来源的HTTP字段组合：

- **cookie** - HTTP Cookie
- **host** — 包含资源的源服务器的主机名
- **方法**- HTTP方法
- **referrer** — 从中获取资源请求的地址
- **url** — 统一资源定位器路径
- **user-agent** — 代理发送请求的软件
- **版本**- HTTP版本
- **via** - HTTP via字段

名为MYHTTP的自定义应用示例，使用选择器ID为10的HTTP主机“*mydomain.com”。

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

SSL自定义

通过从SSL服务器名称指示(SNI)或公用名称(CN)提取的信息，可以对SSL加密流量进行自定义。

名为MYSSL的自定义应用示例，该应用使用具有选择器ID 11的SSL唯一名称“mydomain.com”。

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

DNS自定义

NBAR2会检查DNS请求和响应流量，并且可以将DNS响应关联到应用。从DNS响应返回的IP地址将缓存并用于与该特定应用相关的后续数据包流。

命令dip nbar *customapplication-namednsdomain-nameidapplication*-idis用于DNS自定义。要扩展应用程序，请使用command dip nbar *customapplication*-nameds domain-*namedomain-nameextenddsexisting-application*。

名为MYDNS的自定义应用示例，该应用使用具有选择器ID 12的DNS域名“mydomain.com”。

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

复合定制

NBAR2提供了一种根据HTTP、SSL或DNS中显示的域名自定义应用的方法。

名为MYDOMAIN的自定义应用示例，它使用选择器ID为13的HTTP、SSL或DNS域名“mydomain.com”。

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

L3/L4定制

第3层/第4层自定义基于数据包元组，并且始终在流的第一个数据包上进行匹配。

示例custom application LAYER4CUSTOM，使用选择器ID 14匹配IP地址10.56.1.10和10.56.1.11、TCP和DSCP ef。

```
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
```

监控自定义应用

要监控自定义应用，请使用列出的show命令：

show ip nbar protocol-id | inc自定义

```
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
```

检验AVC

验证AVC功能需要多个步骤，本节提供命令和示例输出。

要验证NBAR是否处于活动状态，可以运行命令“show ip nbar control-plane”

关键领域：

- NBAR状态必须在**正确**方案中激活
- NBAR配置状态必须在正确的**方案**中就绪

```
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>
```

使用show platform software fed switch active|standby|member wdavc function wdavc_stile_cp_show_info_ui命令验证每个交换机成员是否具有活动数据平面:

DP是否激活在正确场景中必须为TRUE

```
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
```

使用“show platform software fed switch active|standby|member wdavc flows命令显示关键信息：

```
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
```

关键字段：

CurrFlows：演示AVC跟踪的活动流数量

水印:显示AVC历来跟踪的最大流数

超时秒:非活动超时取决于已确定的应用程序

应用名称:确定的应用程序

流类型:Real Flow表示这是由入站数据创建的。Pre Flow表示此流是作为入站数据的结果创建的。预流用于预期的媒体流

元组类型:实际流始终为完整元组，预流为完整元组或半元组

旁路:如果设置为TRUE，则表示软件不需要更多数据包来标识此流

最终:如果设置为TRUE，则表示此流不再更改应用程序

绕过包:需要多少数据包才能达到最终分类

#PKTS:此数据流实际上将多少数据包传送到软件

查看有关当前流的其他详细信息，您可以使用命令“**show platform software fed switch active wdavc function wdavc_ft_show_all_flows_seg_ui**”

```
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
```

关键字段

I/F ID:指定接口ID

SEG DIR：指定出口方向的入口

FIF DIR:确定这是否是流发起方方向

NFL HDL：硬件中的流ID

要查看硬件中的条目，请运行命令“**show platform software fed switch active fnf flow-record asic <number> start-index <number> num-flows <number of flows>**

> **注意：**要选择ASIC，它是端口映射到的ASIC实例。要标识ASIC，请使用命令“**show platform software fed switch active|standby|member ifm mappings**”。如果您对特定流不感兴趣，可以将start-index设置为“0”。否则，需要指定start-index。对于num-flows，指定可以查看的流数，最多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
```

在数据路径中查找各种错误和警告

使用命令“show platform software fed switch active|standby|member wdavc function wdavc_ft_show_stats_ui | inc err|warn|无法查看潜在的流表错误：

```
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
```

使用“show platform software fed switch active wdavc function wdavc_stile_stats_show_ui”命令 | inc err”查看任何潜在的NBAR错误：

```
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
```

检验数据包是否已克隆到CPU

使用show platform software fed switch active punt cpuq 21命令 | inc received”验证数据包被克隆到CPU以进行NBAR处理：

> **注意**：在实验中，此数字没有增加。

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

识别CPU拥塞

在拥塞时，数据包可以在发送到WDAVC进程之前被丢弃。使用命令show platform software fed switch active wdavc function fed_wdavc_show_ots_stats_ui进行验证：

```
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
```

> **提示**：要清除punt drop计数器，请使用命令“show platform software fed switch active wdavc function fed_wdavc_clear_ots_stats_ui”

确定扩展问题

如果硬件中没有可用的FNF条目，流量不受NBAR2分类限制。使用命令**show platform software fed switch active fnf sw-table-sizes asic <number> shadow 0**确认：

> **注意：**创建的流特定于交换机和asic核心。需要相应地指定交换机编号（主用、备用等）。输入的ASIC编号绑定到各个接口，使用“**show platform software fed switch active|standby|member ifm mappings**”确定与接口对应的ASIC。对于阴影选项，请始终使用“0”。

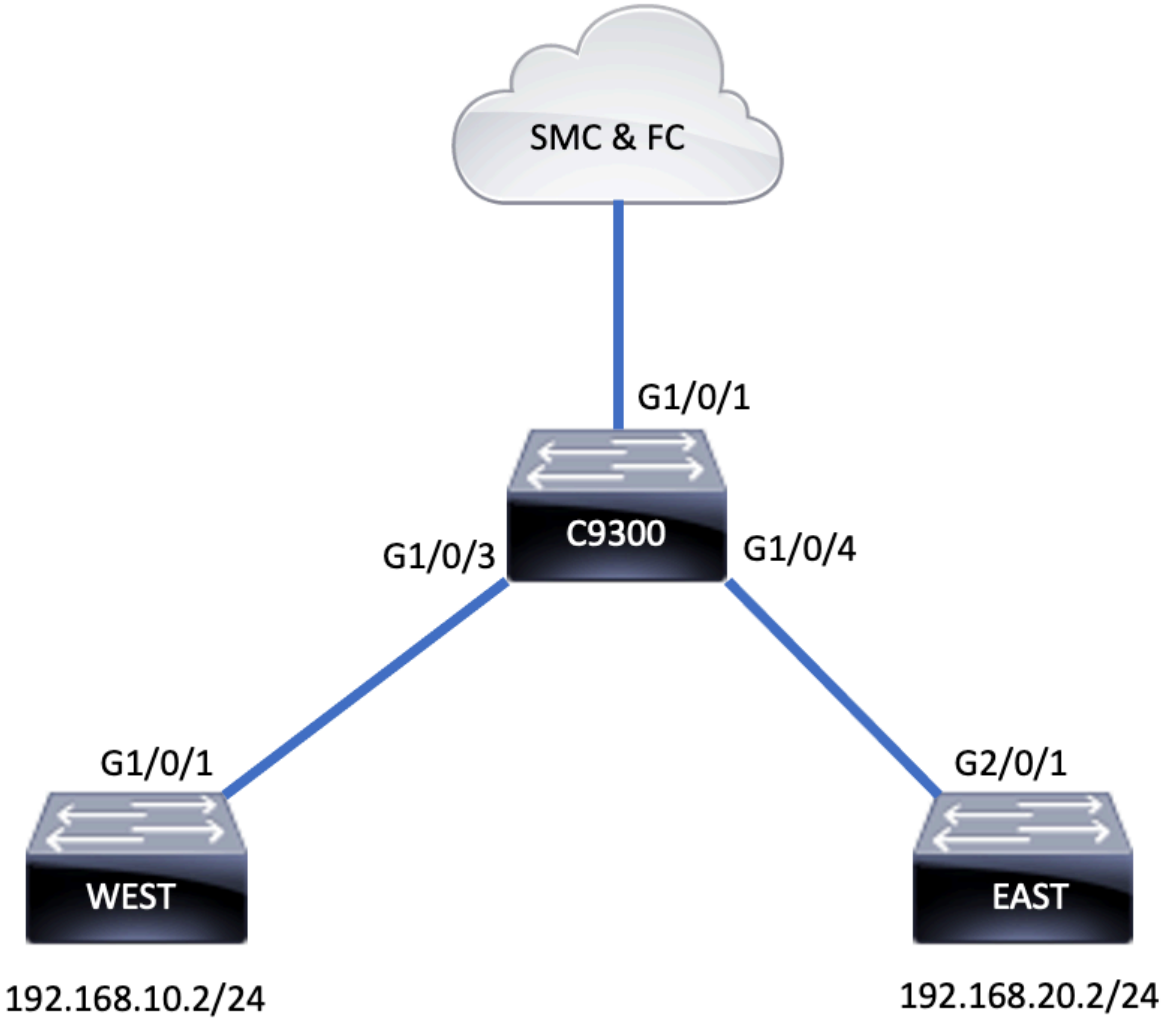
```
Switch#show platform software fed switch active fnf sw-table-sizes asic 3 shadow 0

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

加密流量分析(ETA)

背景信息

- ETA专注于通过被动监控、提取相关数据元素，以及行为建模和机器学习与基于云的全球安全的结合，识别加密流量中的恶意软件通信。
- ETA利用来自NetFlow的遥感勘测数据以及加密的恶意软件检测和加密合规性，并将这些数据发送到Cisco Stealthwatch。
- ETA提取两个主要的数据元素：初始数据包(IDP)和数据包长度和时间序列(SPLT)。

网络图



组件

ETA由多个不同的组件组成，这些组件用于创建ETA解决方案：

- NetFlow — 一种标准，用于定义网络设备导出的数据元素，这些元素描述网络中的流量。
- Cisco Stealthwatch — 利用网络遥感勘测的强大功能（包括NetFlow、IPFIX、代理日志和原始数据包的深度数据包检测），提供高级网络可视性、安全情报和分析。
- 思科认知情报 — 查找绕过安全控制或通过未受监控的渠道进入组织环境内的恶意活动。
- 加密流量分析 — Cisco IOS XE功能使用高级行为算法，通过分析加密流量的传入元数据识别恶

意流量模式，检测加密流量中隐藏的潜在威胁。

注意：本文档的这一部分仅重点介绍在Catalyst 9000系列交换机上配置和验证ETA和NetFlow，但不包括到Cognitive Intelligence Cloud的Stealthwatch管理控制台(SMC)和流量收集器(FC)部署。

限制

- 部署ETA需要DNA优势发挥作用
- 同一接口不支持ETA和传输(TX)交换端口分析器(SPAN)。

这不是包含列表，请查阅交换机的相应配置指南以及适用于所有限制的代码版本。

配置

如输出所示，在交换机上全局启用ETA并定义流导出目标：

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

提示：您必须使用端口2055，请勿使用其他端口号。

接下来，按照输出所示配置Flexible NetFlow:

配置流记录

```
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
```

配置流监控

```
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
```

配置流记录

```
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
```

应用流监控

```
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
```

在交换机接口上启用ETA

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

验证

验证ETA监控器“eta-mon”监控器是否处于活动状态。确认已通过**show flow monitor eta-mon**命令分配状态

```
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
```

验证ETA缓存已填充。当在同一接口上配置NetFlow和ETA时，使用“**show flow monitor <monitor name> cache**”而不是“**show flow monitor eta-mon cache**”，因为“**show flow monitor eta-mon cache”**的输出为空：

```
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
```

使用“show flow exporter eta-exp statistics”命令验证流是否导出到SMC和FC。

```
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
```

使用命令“show platform software fed switch active fnf et-analytics-flows”确认SPLT和IDP已导出到FC。

```
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
```

使用“show platform software et-analytics interfaces”命令验证为et-analytics配置的接口

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

```
ET-Analytics VLANs
```

使用命令“show platform software et-analytics global”查看ETA的全局状态：

```
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
```

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。