Catalyst 9000シリーズスイッチでのNetflow、AVC、およびETAの設定と確認

内容

概要

このドキュメントでは、NetFlow、Application Visibility and Control(AVC)、および暗号化トラフィック分析(ETA)を設定および検証する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- NetFlow
- AVC
- エタ

使用するコンポーネント

このドキュメントの情報は、Cisco IOS XEソフトウェア16.12.4が稼働するCatalyst 9300スイッチに基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

関連製品

このドキュメントは、次のバージョンのハードウェアとソフトウェアにも使用できます。

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

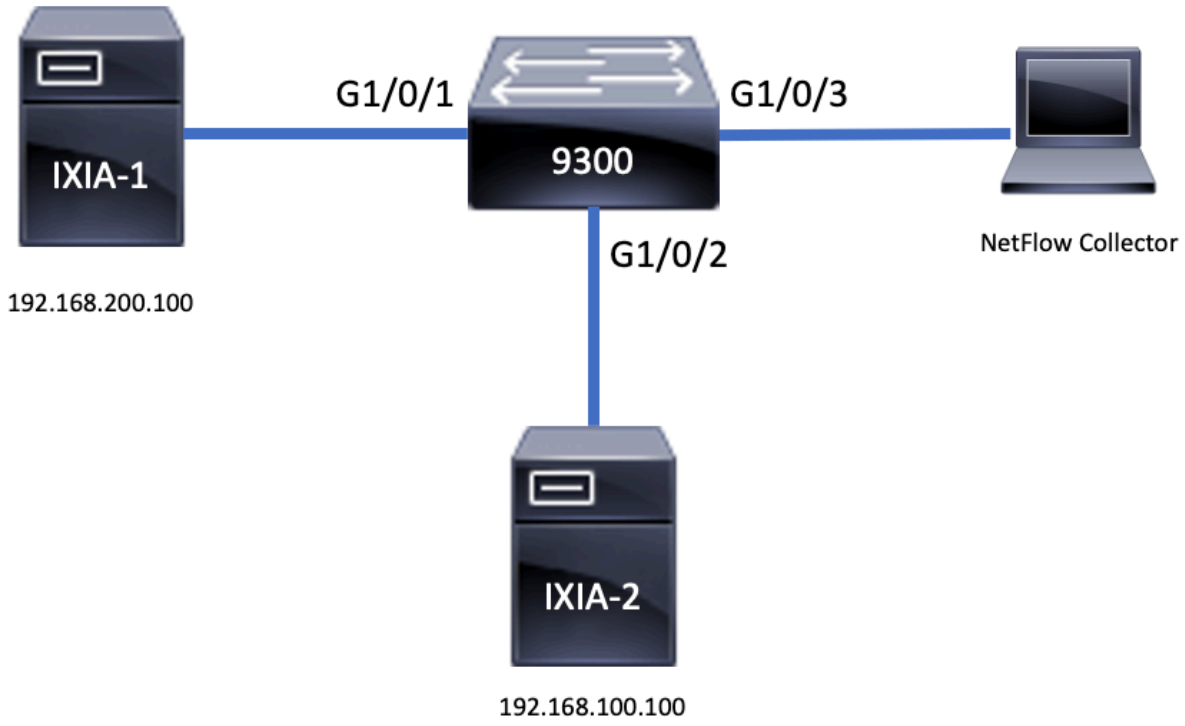
背景説明

- Flexible NetFlowは次世代のフローテクノロジーで、データを収集および測定して、ネットワーク内のすべてのルータまたはスイッチがテレメトリのソースになるようにします。
- Flexible NetFlowにより、非常にきめ細かく正確なトラフィック測定と高レベルの集約トラフィック収集が可能になります。
- Flexible NetFlowはフローを使用して、アカウンティング、ネットワークモニタリング、およびネットワーク計画の統計情報を提供します。
- フローとは、送信元インターフェイスに着信するパケットの単方向ストリームで、キーの値は同じです。キーは、パケット内のフィールドの識別値です。フローレコードを使用してフローを作成し、フローの一意のキーを定義します。

注：プラットフォーム(fed)コマンドは異なる場合があります。コマンドは、「**show platform fed <active|standby>**」と「**show platform fed switch <active|standby>**」です。 例

に示されている構文が解析できない場合は、バリアントを試してください。

ネットワーク図



設定

コンポーネント

NetFlowの設定は、3つの主要なコンポーネントで構成されます。これらのコンポーネントは、トラフィック分析とデータエクスポートを実行するためのさまざまなバリエーションとして使用できます。

フローレコード

- レコードは、キーフィールドと非キーフィールドを組み合わせたものです。Flexible NetFlowレコードはFlexible NetFlowフローモニタに割り当てられ、フローデータの保存に使用されるキャッシュを定義します。
- Flexible NetFlowには、トラフィックの監視に使用できる定義済みのレコードがいくつか用意されています。
- Flexible NetFlowでは、キーおよび非キーフィールドを指定してFlexible NetFlowフローモニタキャッシュ用にカスタムレコードを定義し、データ収集を特定の要件に合わせてカスタマイズすることもできます。

例に示すように、フローレコード設定の詳細は次のとおりです。

```
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
```

フローエクスポータ

- フローエクスポータは、フローモニタキャッシュ内のデータをリモートシステム（NetFlowコレクタとして機能するサーバ）にエクスポートして、分析と保存に使用します。
- フローモニタには、フローモニタのデータエクスポート機能を提供するフローエクスポータが割り当てられます。

例に示すように、フローエクスポータの設定の詳細は次のとおりです。

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

フローモニタ

- フローモニタは、ネットワークトラフィックのモニタリングを実行するためにインターフェイスに適用されるFlexible NetFlowコンポーネントです。
- プロセスの実行中に、ネットワークトラフィックからフローデータが収集され、フローモニタキャッシュに追加されます。このプロセスは、フローレコードのキーフィールドと非キーフィールドに基づいています。

例に示すように、フローモニタの設定の詳細は次のとおりです。

```
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
```

フローサンプラ (オプション)

- フローサンプラは、ルータの設定で個別のコンポーネントとして作成されます。
- フローサンプラは、Flexible NetFlowを使用するデバイスの負荷を軽減するために、分析用に選択するパケットの数を制限します。
- フローサンプラは、分析用に選択されるパケット数の制限によって達成されるFlexible NetFlowを使用するデバイスの負荷を軽減するために使用されます。
- フローサンプラは、ルータのパフォーマンスに対する精度を交換します。フローモニタで分析されるパケット数が減少すると、フローモニタのキャッシュに格納されている情報の精度に影響が及ぶ可能性があります。

例に示すように、フローサンプラの設定例は次のとおりです。

```
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
```

制約事項

- Flexible NetFlowを完全に使用するには、DNAアドオンライセンスが必要です。それ以外の場合は、Sampled NetFlowのみを使用できます。
- フローエクスポータは、管理ポートを送信元として使用できません。

これは包括的なリストではありません。適切なプラットフォームとコードの設定ガイドを参照してください。

確認

プラットフォームに依存しない検証

設定を確認し、必要なNetFlowコンポーネントが存在することを確認します。

1. **フローレコード**
2. **フローエクスポータ**
3. **フローモニタ**
4. **フローサンプラ (オプション)**

   ヒント：フローレコード、フローエクスポータ、およびフローモニタの出力を1つのコマンドで表示するには、「**show running-config flow monitor <flow monitor name> expand**」を実行します

例に示すように、フローモニタは入力方向とその関連コンポーネントに関連付けられています。

```
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
!
```

例に示すように、フローモニタは出力方向および関連コンポーネントに関連付けられています。

```
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
!
```

コマンド「show flow monitor <flow monitor name>」の統計情報を実行します。次の出力は、データが記録されていることを確認するのに役立ちます。

```
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
```

"show flow monitor <flow monitor name> cache コマンドを実行して、NetFlowキャッシュに出力

があることを確認します。

```
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」コマンドを実行して、エクスポータがパケットを送信したことを確認します。

```
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
```

プラットフォーム依存の検証

NetFlow初期化 – NFLパーティションテーブル

- NetFlowパーティションは、方向（入力と出力）ごとに16個のパーティションを持つ異なる機能用に初期化されます。
- NetFlowパーティションテーブルの設定は、グローバルバンクの割り当てに分割され、さらに入力および出力フローバンクに分割されます。

キーフィールド

- パーティション数
- パーティションの有効化ステータス
- パーティション制限
- 現在のパーティション使用状況

NetFlowパーティションテーブルを表示するには、コマンド「show platform software fed switch active|standby|member| fnf sw-table-sizes asic <asic number> shadow 0」を実行します。

注：作成されるフローは、作成時にスイッチとASICコアに固有です。スイッチ番号（アクティブ、スタンバイなど）を適宜指定する必要があります。入力されるASIC番号はそれぞれのインターフェイスに関連付けられています。「**show platform software fed switch active|standby|member ifm mappings**」を使用して、インターフェイスに対応するASICを判別します。shadowオプションには、常に「0」を使用します。

```
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
```

フローモニタ

フローモニタの設定には、次のものが含まれます。

1. NetFlow ACLの設定。これにより、ACL TCAMテーブル内にエントリが作成されます。

ACL TCAMエントリは次の要素で構成されます。

- 一致キーの検索
- NetFlowルックアップに使用される結果パラメータ。次のものが含まれます。
  プロファイルIDNetFlow ID

2.フローマスクの設定。これにより、NflLookupTableおよびNflFlowMaskTableにエントリが作成されます。

- NetFlow ACLの結果パラメータによってインデックスが作成され、NetFlowルックアップ用のフローマスクが検索されます。

NetFlow ACL

NetFlow ACL設定を表示するには、コマンド「**show platform hardware fed switch active fwd-asic resource tcam table nfl_acl asic <asic number>**

> **ヒント**：ポートACL(PACL)がある場合、インターフェイスがマッピングされているASIC上にエントリが作成されます。ルータACL(RACL)の場合、エントリはすべてのASICに存在します。

- この出力には、4ビット値のNFCMD0とNFCMD1があります。プロファイルIDを計算するには、値を2進数に変換します。
- この出力では、NFCMD0は1、NFCMD1は2です。バイナリに変換すると、次のようになります。000100010
- Cisco IOS-XE 16.12以降では、結合された8ビットの最初の4ビットがプロファイルIDで、7番目のビットがルックアップが有効であることを示します。したがって、例の**0001**では、プロファイルIDは1です。
- Cisco IOS XE 16.11以前のバージョンのコードでは、合計8ビット内の最初の6ビットがプロファイルIDで、7番目のビットがルックアップが有効であることを示します。したがって、この例の**000100**では、プロファイルIDは4です。

```
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

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

フローマスク

「show platform software fed switch active|standby|member fnf fmask-entry asic <asic number> entry 1」コマンドを実行して、フローマスクがハードウェアにインストールされていることを確認します。キーフィールドのリストの数は、こちらにも記載されています。

```
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)
```

フロー統計とタイムスタンプオフロードデータ

コマンド"show platform software fed switch active fnf flow-record asic <asic number> start-index <index number> num-flows <number of flows> を実行して、netflowの統計情報とタイムスタンプを表示します

```
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
```

アプリケーションの可視性と制御（AVC）

背景説明

- Application Visibility and Control(AVC)は、Network-Based Recognition Version 2(NBAR2)、NetFlow V9、およびさまざまなレポートと管理ツール(Cisco Prime)を利用するソリューションで、ディープパケットインスペクション(DPI)によってアプリケーションを分類できます。
- AVCは、スタンドアロンスイッチまたはスイッチスタックの有線アクセスポートで設定できます。
- AVCは、Ciscoワイヤレスコントローラで使用して、DPIに基づいてアプリケーションを識別し、特定のDSCP値でマーキングすることもできます。また、アプリケーションやクライアントに関する帯域幅使用量など、さまざまなワイヤレスパフォーマンスメトリックを収集す

ることもできます。

パフォーマンスと拡張性

パフォーマンス：各スイッチメンバーは、500 CPU使用率50 %未満で500 CPS（接続/秒）を処理できます。このレートを超えると、AVCサービスは保証されません。

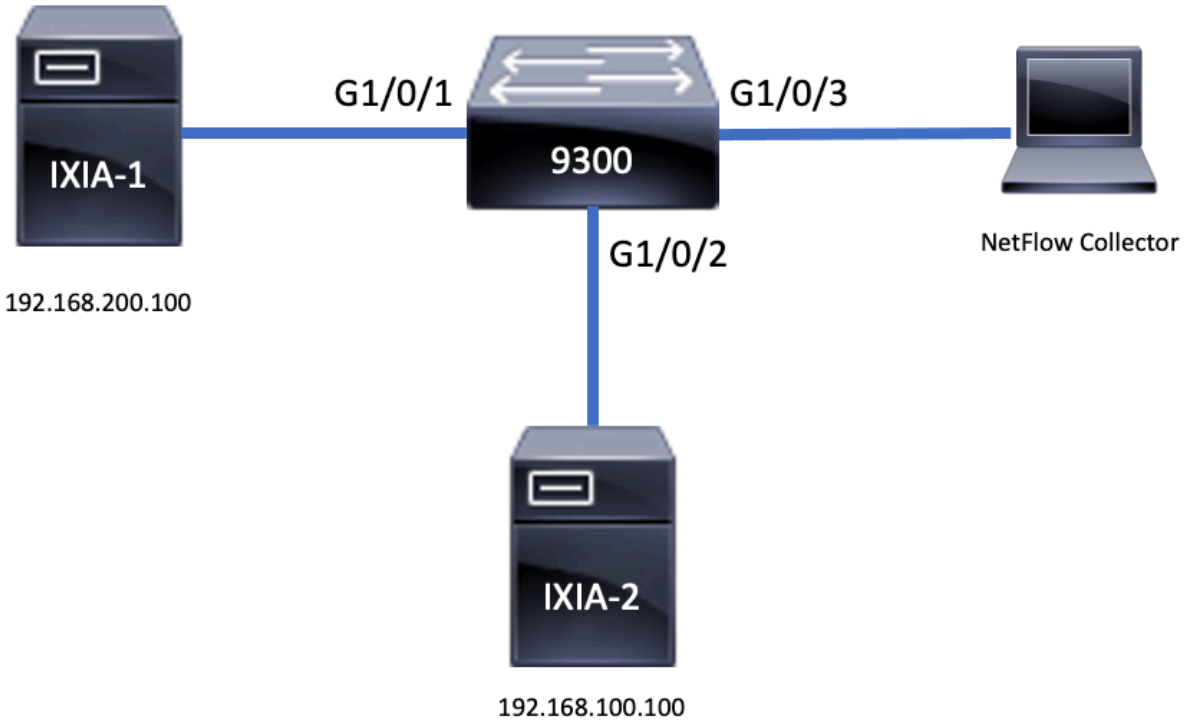
拡張性：24のアクセスポートあたり最大5000の双方向フローを処理できます（アクセスポートあたり約200のフロー）。

有線AVCの制限

- AVCとEncrypted Traffic Analytics(ETA)を同じインターフェイスで同時に設定することはできません。
- パケット分類は、ユニキャストIPv4(TCP/UDP)トラフィックでのみサポートされます。
- NBARベースのQoSポリシー設定は、有線物理ポートでのみサポートされます。これには、レイヤ2アクセスポートとトランクポート、およびレイヤ3ルーテッドポートが含まれます。
- NBARベースのQoSポリシー設定は、ポートチャネルメンバー、スイッチ仮想インターフェイス(SVI)、またはサブインターフェイスではサポートされません。
- NBAR2ベースの分類子(**match protocol)は、マーキングとポリシング**のQoSアクションのみをサポートしています。
- 「match protocol」は、すべてのポリシーで255の異なるプロトコルに制限されています（8ビットハードウェアの制限）

  注：これはすべての制限の完全なリストではありません。ご使用のプラットフォームとコードのバージョンに適したAVC設定ガイドを参照してください。

ネットワーク図



コンポーネント

AVC構成は、ソリューションを構成**する3つの主要**コンポーネントで構成されています。

可視性：プロトコル ディスカバリ

- プロトコル検出はNBARによって実現されます。NBARは、インターフェイス単位、方向、およびアプリケーションのバイト/パケットの統計情報を提供します。
- 特定のインターフェイスに対して、インターフェイス設定**ip nbar protocol-discovery**を使用してプロトコル検出を有効にします。

出力に示されているように、プロトコルディスカバリを有効にする方法は次のとおりです。

```
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
```

Control:アプリケーションベースのQoS

IPアドレスとUDP/TCPポートで一致する従来のQoSと比較すると、AVCはアプリケーションベースのQoSを通じてより細かい制御を実現します。これにより、アプリケーションで一致し、マーキングやポリシングなどのQoSアクションを通じてより細かい制御が可能になります。

- 集約されたトラフィックに対してアクションが実行される（フローごとではない）
- アプリケーションベースのQoSは、クラスマップの作成、プロトコルの照合、ポリシーマップの作成によって実現されます。
- アプリケーションベースのQoSポリシーは、インターフェイスに適用されます。

出力に示されているように、アプリケーションベースのQoSの設定例は次のとおりです。

```
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
```

アプリケーションベースのFlexible NetFlow

有線AVC FNFは、2種類の定義済みフローレコードをサポートします。**従来の双方向フローレコード**と新しい方向**フローレコードです**。

双方向フローレコードにより、クライアント/サーバアプリケーションの統計情報を追跡します。

出力に示されているように、双方向フローレコードの設定例を参照してください。

```
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
```

ディレクショナルレコードは、入力/出力のアプリケーション統計です。

出力に示すように、入力および出力方向レコードの設定例は次のとおりです。

> **注**：コマンド「match interface input」は、入力インターフェイスへの一致を指定します。コマンド「match interface output」は、出力インターフェイスとの一致を指定します。AVCのサポートには、「match application name」コマンドが必須です。

```
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
```

フローエクスポータ

フローエクスポータを作成して、エクスポートパラメータを定義します。

出力に示されているように、フローエクスポータの設定例は次のとおりです。

```
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
!
```

フローモニタ

フローモニタを作成して、フローレコードに関連付けます。

出力に示すように、フローモニタの設定例は次のとおりです。

```
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
```

フローモニタのインターフェイスへの関連付け

1つのインターフェイスに、事前定義されたレコードが異なる最大2つの異なるAVCモニタを**同時に接続**できます。

出力に示すように、フローモニタの設定例は次のとおりです。

```
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アップグレード

プロトコルパックは、デバイスのシスコソフトウェアを交換することなく、デバイスのNBAR2プロトコルサポートを更新するソフトウェアパッケージです。プロトコル・パックには、NBAR2によって正式にサポートされるアプリケーションに関する情報がまとめてコンパイルおよびパックされます。アプリケーションごとに、プロトコル・パックにはアプリケーション・シグネチャおよびアプリケーション属性に関する情報が含まれます。各ソフトウェアリリースには、組み込みのプロトコルパックがバンドルされています。

- NBAR2は、トラフィックやサービスを中断することなく、またデバイスのソフトウェアイメージを変更することなく、プロトコルパケットを更新する方法を提供します
- NBAR2プロトコルパケットは、次のURLからCisco Software Centerにダウンロードできます。https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html

NBAR2プロトコルパックアップグレード

新しいプロトコルパックをインストールする前に、プロトコルパケットをすべてのスイッチのフラッシュにコピーする必要があります。 新しいプロトコルパックをロードするには、コマンド「ip nbar protocol-pack flash:<Pack Name>

NBAR2のアップグレードを実行するためにスイッチをリロードする必要はありません。

出力に示すように、NBAR2プロトコルパックのロード方法の設定例は次のとおりです。

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

組み込みプロトコルパックに戻すには、コマンド「default ip nbar protocol-pack」を使用します。

出力に示すように、組み込みプロトコルパックに戻す方法の設定例は次のとおりです。

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

NBAR2プロトコルパック情報の表示

プロトコルパック情報を表示するには、次に示すコマンドを使用します。

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

これらのコマンドの出力例を次に示します。

```
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カスタムアプリケーション

NBAR2は、カスタムプロトコルを使用してカスタムアプリケーションを識別する機能をサポートしています。カスタムプロトコルは、NBAR2が現在サポートしていないプロトコルとアプリケーションをサポートします。

これには次のようなものがあります。

- 組織への特定の適用
- 地域固有のアプリケーション

NBAR2では、**ip nbar custom<*myappname*>**コマンドを使用して、アプリケーションを手動でカスタマイズできます。

注：カスタムアプリケーションは、組み込みプロトコルよりも優先されます

アプリケーションのカスタマイズには、さまざまなタイプがあります。

汎用プロトコルカスタマイズ

- HTTP
- SSL
- DNS

コンポジット：複数のプロトコルに基づくカスタ**マイズ – server-name**

レイヤ3/レイヤ4のカスタマイズ

- IPv4 アドレス
- DSCP値
- TCP/UDPポート
- フローの送信元または宛先の方向

バイトオフセット：ペイロードの特定のバイト値に基づくカスタマイズ

HTTPカスタマイズ

HTTPカスタマイズは、次のHTTPフィールドの組み合わせに基づいて行うことができます。

- **cookie**:HTTPクッキー
- **host**：リソースを含むオリジンサーバのホスト名
- **method**:HTTPメソッド
- **referrer**：リソース要求の取得元アドレス
- **url**:Uniform Resource Locatorのパス
- **user-agent**：要求を送信するエージェントによって使用されるソフトウェア
- **version**:HTTPバージョン
- **via**:HTTP viaフィールド

HTTPホスト"*mydomain.com"とセレクタID 10を使用するMYHTTPという名前のカスタムアプリケーションの例。

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

SSLカスタマイズ

SSLサーバ名表示(SNI)または共通名(CN)から抽出された情報を使用して、SSL暗号化トラフィックのカスタマイズを行うことができます。

セレクタID 11のSSL一意の名前「mydomain.com」を使用するMYSLという名前のカスタムアプリケーションの例。

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

DNSのカスタマイズ

NBAR2はDNS要求および応答トラフィックを調べ、DNS応答をアプリケーションに関連付けることができます。DNS応答から返されたIPアドレスはキャッシュされ、その特定のアプリケーションに関連付けられた後のパケットフローに使用されます。

コマンドdip nbar *customapplication-namednsdomain-namedapplication*-idは、DNSカスタマイズに使用されます。アプリケーションを拡張するには、commandip nbar *customapplication-*namedns domain-*namedomain-nameextendsexisting-application*コマンドを使用します。

セレクタID 12のDNSドメイン名「mydomain.com」を使用するMYDNSという名前のカスタムアプリケーションの例。

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

複合カスタマイズ

NBAR2では、HTTP、SSL、またはDNSに表示されるドメイン名に基づいてアプリケーションをカスタマイズできます。

HTTP、SSL、またはDNSドメイン名「mydomain.com」（セレクタID 13）を使用するMYDOMAINという名前のカスタムアプリケーションの例。

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

L3/L4カスタマイズ

レイヤ3/レイヤ4のカスタマイズはパケットタプルに基づいており、常にフローの最初のパケットに一致します。

IPアドレス10.56.1.10と10.56.1.11、セレクタID 14のTCPおよびDSCP efに一致するカスタムアプリケーションLAYER4CUSTOMの例。

```
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
```

カスタムアプリケーションの監視

カスタムアプリケーションをモニタするには、次に示すshowコマンドを使用します。

show ip nbar protocol-id | incカスタム

```
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の確認

AVCの機能を検証するには複数の手順があります。このセクションでは、コマンドと出力例を示します。

NBARがアクティブであることを確認するには、コマンド「show ip nbar control-plane」を実行します。

主な分野：

- 正しいシナリオでNBAR状態を**アクティブ**にする必要があります
- NBARの設定状態は、正しいシナリオで**ready**である必要があります

```
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コマンドを使用して、各スイッチメンバにアクティブなデータプレーンがあることを確認します。

DPが有効になっているかどうかは、正しいシナリオでは**TRUE**である必要があります

```
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」コマンドを使用して、重要な情報を表示します。

```
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
```

キーフィールド：

CurrFlows:AVCによって追跡されるアクティブフローの数を示します。

透かし:AVCが過去に追跡したフローの最大数を示す

TIMEOUT SEC:特定されたアプリケーションに基づく非アクティビティタイムアウト

アプリケーション名:特定されたアプリケーション

フロータイプ：Real Flowは、これがインバウンドデータの結果として作成されたことを示します。Pre Flowは、このフローがインバウンドデータの結果として作成されることを示します。プレフローは、予想されるメディアフローに使用されます

組の種類:実際のフローは常に完全なタプルであり、事前フローは完全なタプルか半分のタプルです

バイパス:TRUEに設定すると、このフローを識別するためにソフトウェアでこれ以上パケットが必要ないことを示します

最終版:TRUEに設定すると、アプリケーションはこのフローに対してこれ以上変更されないことを示します

BYPASS PKT:最終的な分類に到達するために必要なパケット数

#PKTS:このフローでソフトウェアに実際にパントされたパケットの数

現在のフローの詳細を表示するには、コマンド「**show platform software fed switch active wdavc function wdavc_ft_show_all_flows_seg_ui」を使用します。**

```
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
```

キーフィールド

I/F ID:インターフェイスIDを指定します

SEG DIR：出力方向の入力を指定します。

FIF DIR：これがフローイニシエータの方向かどうかを決定します

NFL HDL：ハードウェアのフローID

ハードウェアでエントリを表示するには、コマンド「**show platform software fed switch active fnf flow-record asic <number> start-index <number> num-flows <number of flows>**

> **注：**ASICを選択するには、ポートがマッピングされているASICインスタンスを選択します。ASICを識別するには、コマンド「**show platform software fed switch active|standby|member ifm mappings**」を使用します。特定のフローを使用しない場合は、start-indexを「0」に設定できます。そうでない場合は、start-indexを指定する必要があります。num-flowsでは、表示可能なフローの数を指定します。最大10です。

```
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
```

データパスでさまざまなエラーや警告を探す

コマンド「show platform software fed switch active|standby|member wdavc function wdavc_ft_show_stats_ui |inc err|warn|潜在的なフローテーブルエラーを表示できません：

```
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
```

コマンド「show platform software fed switch active wdavc function wdavc_stile_stats_show_ui | inc err"を発行して、NBARエラーの可能性を確認します。

```
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
```

パケットがCPUに複製されていることの確認

コマンド「show platform software fed switch active punt cpuq 21 | inc received」を参照してください。

注：ラボでは、この数値は増加しませんでした。

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

CPU輻輳の特定

輻輳時には、パケットはWDAVCプロセスに送信される前にドロップされる可能性があります。コマンド「show platform software fed switch active wdavc function fed_wdavc_show_ots_stats_ui」を使用して検証します。

```
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
```

> **ヒント：**パントドロップカウンタをクリアするには、コマンド「show platform software fed switch active wdavc function fed_wdavc_clear_ots_stats_ui」を使用します。

スケールの問題の特定

ハードウェアに空きFNFエントリがない場合、トラフィックはNBAR2分類の対象になりません。コマンド「show platform software fed switch active fnf sw-table-sizes asic <number> shadow 0」を使用して確認します。

> **注：**作成されるフローは、作成時にスイッチとASICコアに固有です。スイッチ番号（アクティブ、スタンバイなど）を適宜指定する必要があります。入力されるASIC番号はそれぞれのインターフェイスに関連付けられています。「show platform software fed switch active|standby|member ifm mappings」を使用して、インターフェイスに対応するASICを判別します。shadowオプションには、常に「0」を使用します。

```
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
```

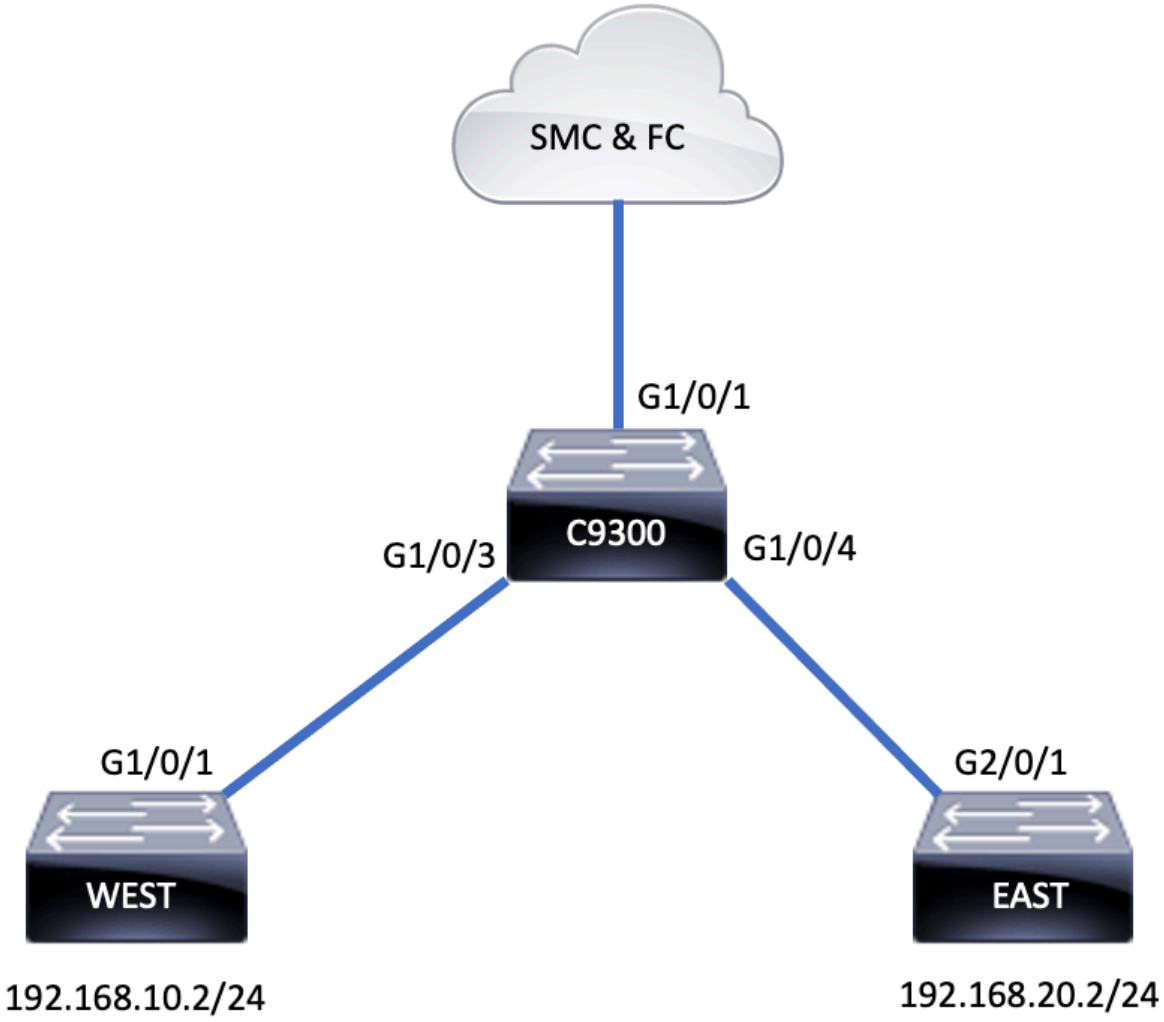
```
EGRESS usedBankEntry 0 usedOvfTcamEntry 0 <-- 256 means TCAM entries are full
--------------------------------
Flows Statistics
INGRESS TotalSeen=1 MaxEntries=1 MaxOverflow=0
EGRESS TotalSeen=0 MaxEntries=0 MaxOverflow=0

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

暗号化トラフィック分析(ETA)

背景説明

- ETAは、パッシブモニタリング、関連データ要素の抽出、および行動モデリングと機械学習とクラウドベースのグローバルセキュリティの組み合わせによる、暗号化されたトラフィック内のマルウェア通信の特定に重点を置いています。
- ETAは、NetFlowからのテレメトリに加え、暗号化されたマルウェア検出と暗号化コンプライアンスを活用し、このデータをCisco Stealthwatchに送信します。
- ETAは、次の2つの主要なデータ要素を抽出します。初期データパケット(IDP)とパケット長と時間のシーケンス(SPLT)。

ネットワーク図



コンポーネント

ETAは、ETAソリューションの作成に使用される複数の異なるコンポーネントで構成されています。

- NetFlow：ネットワークデバイスによってエクスポートされるデータ要素を定義し、ネットワーク上のフローを記述する標準。
- Cisco Stealthwatch:NetFlow、IPFIX、プロキシログ、未加工パケットのディープパケットインスペクションなどのネットワークテレメトリの機能を利用して、高度なネットワークの可視性、セキュリティインテリジェンス、分析を提供します。
- Cisco Cognitive Intelligence：セキュリティ制御をバイパスした悪意のあるアクティビティ、または監視されていないチャネルを介して組織の環境内に入った悪意のあるアクティビティを検出します。
- 暗号化トラフィック分析：高度な動作アルゴリズムを使用して、暗号化トラフィックのインフラストラクチャメタデータの分析を通じて悪意のあるトラフィックパターンを特定し、暗号化トラフィックに隠れている潜在的な脅威を検出するCisco IOS XE機能。

注：このドキュメントでは、Catalyst 9000シリーズスイッチでのETAおよびNetFlowの設定と検証のみを対象とし、Cognitive Intelligence CloudへのStealthWatch Management Console(SMC)とFlow Collector(FC)の導入については取り上げていません。

制約事項

- ETAの導入にはDNAの優位性が必要である
- ETAと送信(TX)スイッチドポートアナライザ(SPAN)は、同じインターフェイスではサポートされていません。

これは包括的なリストではありません。すべての制限について、スイッチとコードのバージョンに対応するコンフィギュレーションガイドを参照してください。

コンフィギュレーション

出力に示されているように、スイッチでETAをグローバルに有効にし、フローエクスポートの宛先を定義します。

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

> **ヒント**：ポート2055を使用する必要があります。別のポート番号を使用しないでください。

次に、出力に示すようにFlexible NetFlowを設定します。

フローレコードの設定

```
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
```

フローモニタの設定

```
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
```

フローレコードの設定

```
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
```

Apply Flow Monitor

```
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
```

スイッチのインターフェイスでETAを有効にする

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

確認

ETAモニタ「eta-mon」モニタがアクティブであることを確認します。show flow monitor eta-monコマンドを使用して、ステータスが割り当てられていることを確認します。

```
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キャッシュにデータが入力されていることを確認します。NetFlowとETAが同じインターフェイス上で設定されている場合は、「show flow monitor eta-mon cache」の出力が空であるため、「show flow monitor eta-mon cache」ではなく「show flow monitor <monitor name> cache」を使用します。

```
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
```

コマンド「show flow exporter eta-exp statistics」を使用して、フローがSMCおよびFCにエクスポートされることを検証します。

```
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
```

コマンド「show platform software fed switch active fnf et-analytics-flows」を使用して、SPLTとIDPがFCにエクスポートされていることを確認します。

```
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コマンドを使用して、どのインターフェイスがet-analytics用に設定されているかを検証します。

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

ET-Analytics VLANs
```

ETAのグローバル状態を表示するには、コマンド「show platform software et-analytics global」を

使用します。

```
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
```

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。