排除ACI安全策略故障 — 合同

目录

简介

本文档介绍了解并排除ACI安全策略（称为合同）故障的步骤。

背景信息

本文档中的材料摘自《思科以应用为中心的基础设施故障排除》第二版书，特别是《安全策略 — 概述》、《安全策略 — 工具》、《安全策略 — **EPG至EPG》、《安全策略 — 首选组》**和《安全策略 — vzAny至EPG》章。

概述

ACI解决方案的基本安全架构遵循许可列表模式。除非在**unenforced**模式下配置VRF，否则会隐式丢弃所有EPG到EPG的流量。如开箱即用的许可列表模型所示，默认VRF设置处于**enforced**模式。通过在交换机节点上实施分区规则，可以允许或明确拒绝流量。根据终端组(EPG)之间的期望通信流和用于定义这些规则的方法，这些分区规则可以编程为各种不同的配置。请注意，分区规则条目不是有状态的，通常会在规则编程后根据端口/套接字允许或拒绝给定两个EPG。

分区规则的编程方法

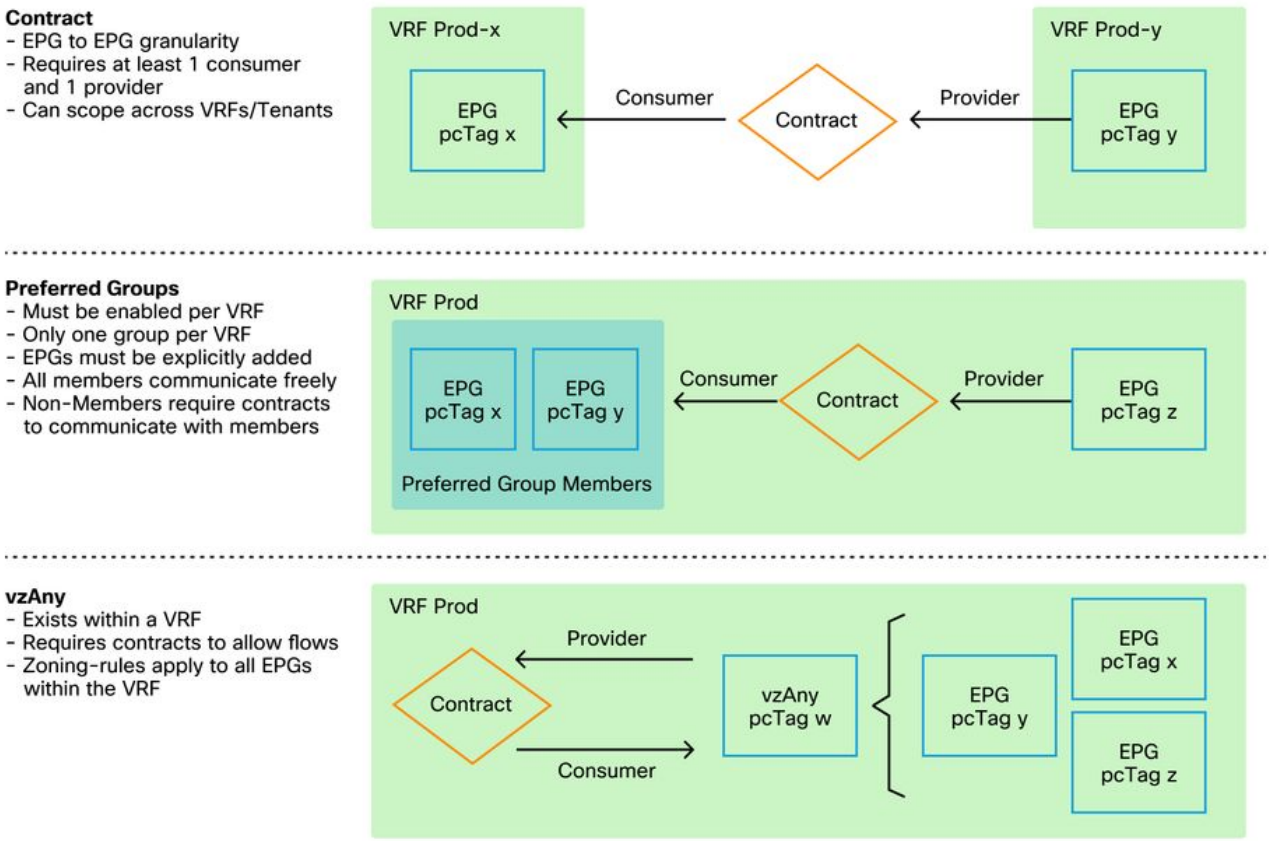
在ACI中规划分区规则的主要方法如下：

- **EPG到EPG合同：通常至少需要一个消费者和一个提供商对两个或多个不同终端组之间的分区规则进行编程。**

- **首选组**：需要在VRF级别启用分组；每个VRF只能存在一个组。小组所有成员均可自由交流。非成员需要合同以允许流向首选组。
- **vzAny:**在给定VRF下定义的“EPG集合”。vzAny表示VRF中的所有EPG。vzAny的使用允许一个EPG和VRF内的所有EPG之间通过一个合同连接进行流动。

下图可用于引用上述每种方法允许控制的分区规则的粒度：

分区规则方法的比较



在使用合同方法编写分区规则时，有一个选项用于定义合同范围。如果需要任何路由泄漏/共享服务设计，必须仔细考虑此选项。如果希望从ACI交换矩阵中的一个VRF连接到另一个VRF，则合同是执行此操作的方法。

范围值可以是以下值：

- **应用：合**同消费者/提供商关系将仅编程同一应用配置文件中定义的EPG之间的规则。在其他应用配置文件EPG之间重复使用同一合同不会允许它们之间的串扰。
- **VRF（默认）：合**同消费者/提供商关系将对同一VRF中定义的EPG之间的规则进行编程。在其他应用配置文件EPG之间重复使用同一合同将允许它们之间的串扰。请注意确保只允许所需的数据流，否则应定义新合同以防止无意串扰。
- **租户：合**同消费者/提供商关系将对同一租户内定义的EPG之间的规则进行编程。如果在单个租户内存在与多个VRF绑定的EPG，并且它们使用/提供相同的合同，则此范围可用于诱导路由泄漏，以允许VRF间通信。
- **全局：合**同消费者/提供商关系将对ACI交换矩阵内任何租户之间的EPG之间的规则进行编程。这是定义的最高范围，在先前定义的合同上启用此功能时应格外小心，以防止无意的流量泄漏。

读取分区规则条目

一旦对zoning-rule进行了编程，它将在枝叶上显示如下：

```
+---------+--------+--------+----------+----------------+---------+---------+----------------+-
---------+---------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope  |       Name     |
Action  |       Priority      |
+---------+--------+--------+----------+----------------+---------+---------+----------------+-
---------+---------------------+
```

- **规则ID**：规则条目的ID。除了充当唯一标识符之外，没有其他实际意义。
- **源EPG:**源终端组的每个VRF(pcTag)的唯一ID。
- **Dst EPG：目**标终端组的每个VRF(pcTag)的唯一ID。
- **FilterID:**规则尝试与之匹配的过滤器的ID。过滤器包含规则将匹配的协议信息。
- **Dir:**zoning-rule的方向性。
- **OperSt:**规则的运行状态。
- **范围**：规则匹配的VRF的唯一ID。
- **名称**：导致对该条目进行编程的合同的名称。
- **操作：**枝叶匹配该条目时将执行的操作。包括：[丢弃、允许、记录、重定向]。
- **优先级**：在给定匹配范围、SrcEPG、DstEPG和过滤器条目的情况下验证分区规则进行操作的顺序。

策略内容可寻址存储器(CAM)

随着每个分区规则的编程，根据过滤器条目映射的分区规则条目的矩阵将开始使用交换机上的**Policy CAM**。在设计通过ACI交换矩阵的允许流时，应特别注意重复使用合同，而不是根据最终设计创建新合同。在不了解所导致的分区规则的情况下，随意地跨多个EPG重复使用同一合同，可能会快速级联成意外允许的多个流。同时，这些无意的流量将继续消耗策略CAM。当策略CAM变满时，分区规则编程将开始失败，这可能会导致意外和间歇性丢失，具体取决于配置和终端行为。

共享L3Outs的VRF泄漏、全局pcTags和策略实施方向

这是需要配置合同的共享服务使用案例的特殊标注。共享服务通常意味着ACI交换矩阵内的VRF间流量，该流量依赖于使用“租户”或“全局”范围合同。要充分理解这一点，必须首先强调分配给EPG的典型pcTag值并非全球唯一的。pcTags的范围被限定在VRF中，同一个pcTag可能会在另一个VRF中重复使用。当讨论路由泄漏时，开始在ACI交换矩阵上实施要求，包括需要全局唯一值，包括子网和pcTags。

使这一点成为特殊考虑的是将EPG作为消费者与提供商相关联的方向性方面。在共享服务方案中，通常希望提供商驱动全局pcTag以获得交换矩阵唯一值。同时，消费者将保留其VRF范围的pcTag，使其处于特殊位置，以便现在能够编程和了解使用全局pcTag值执行策略。

作为参考，pcTag分配范围如下：

- 系统保留：1-15 的多播地址发送一次邻居消息。
- 全局范围：共享服务16384供商EPG的16-G。
- 本地范围：16385范围EPG的VRF-65535。

VRF策略控制实施方向

在每个VRF中，可以定义实施方向设置。

- 实施方向的默认设置为Ingress。
- 实施方向的另一个选项是出口。

了解策略的实施位置取决于几个不同的变量。

下表有助于了解在枝叶级别执行安全策略的位置。

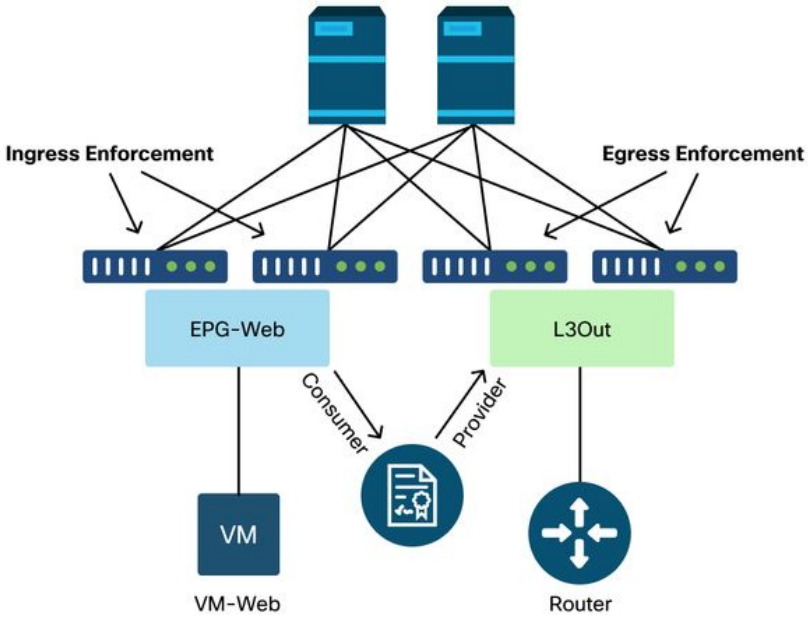
在哪里实施策略？

场景	VRF实施模式	消费者	提供商	策略实施于
VRF内	入口/出口	EPG	EPG	·如果获知目标终端：入口枝叶* ·如果未获知目标终端：出口枝叶
	入口	EPG	L3Out EPG	消费者枝叶（非边界枝叶）
	入口	L3Out EPG	EPG	提供商枝叶（非边界枝叶）
	出口	EPG	L3Out EPG	边界枝叶 — >非边界枝叶流量 ·如果获知目标终端：边界枝叶 ·如果未获知目标终端：非边界枝叶
	出口	L3Out EPG	EPG	非边界枝叶 — >边界枝叶流量 ·边界枝叶
	入口/出口	L3Out EPG	L3Out EPG	入口枝叶*
	入口/出口	EPG	EPG	消费者枝叶
	入口/出口	EPG	L3Out EPG	消费者枝叶（非边界枝叶）
VRF间	入口/出口	L3Out EPG	EPG	入口枝叶*
	入口/出口	L3Out EPG	L3Out EPG	入口枝叶*

*策略实施应用于数据包命中后的第一个枝叶。

下图显示了一个合同实施示例，其中EPG-Web作为消费者，L3Out EPG作为提供商具有VRF内合同。如果VRF设置为入口实施模式，策略由EPG-Web所在的枝叶节点实施。如果VRF设置为出口实施模式，则如果VM-Web终端是在边界枝叶上获取的，则由L3Out所在的边界枝叶节点实施策略。

入口实施和出口实施



工具

有许多工具和命令可用于帮助识别策**略丢弃**。策略丢弃可定义为由于合同配置或缺少合同配置而丢弃的数据包。

分区规则验证

以下工具和命令可用于明确验证由于已完成的合同消费者/提供商关系而在枝叶交换机上编程的分区规则。

'show zoning-rules'

显示所有分区规则的交换机级命令。

```
leaf# show zoning-rule
+---------+--------+--------+----------+----------------+---------+----------+----------------
+----------+----------------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope   |       Name
|  Action  |          Priority          |
+---------+--------+--------+----------+----------------+---------+----------+----------------
+----------+----------------------------+
|   4156  |   25   | 16410  |   425    | uni-dir-ignore | enabled | 2818048  | external_to_ntp
|  permit  |       fully_qual(7)        |
|   4131  | 16410  |   25   |   424    |     bi-dir     | enabled | 2818048  | external_to_ntp
|  permit  |       fully_qual(7)        |
+---------+--------+--------+----------+----------------+---------+----------+----------------
+----------+----------------------------+
```

'show zoning-filter'

包含分区规则正在执行的运动/数据流的过滤器。过滤器编程可以用此命令进行验证。

```
leaf# show zoning-filter
+----------+----------+-------------+-------------+-------------+----------+-------------+------
-------+-------------+-------------+----------+
| FilterId |   Name   |    EtherT   |     Prot    | ApplyToFrag | Stateful |  SFromPort  |
SToPort   |  DFromPort  |   DToPort   |   Prio   |
+----------+----------+-------------+-------------+-------------+----------+-------------+------
-------+-------------+-------------+----------+
| implarp  | implarp  |     arp     | unspecified |      no     |    no    | unspecified |
unspecified | unspecified | unspecified |  dport   |
| implicit | implicit | unspecified | unspecified |      no     |    no    | unspecified |
unspecified | unspecified | unspecified | implicit |
|   425    |  425_0   |      ip     |     tcp     |      no     |    no    |     123     |
123     | unspecified | unspecified |  sport   |
|   424    |  424_0   |      ip     |     tcp     |      no     |    no    | unspecified |
unspecified |     123     |     123     |  dport   |
+----------+----------+-------------+-------------+-------------+----------+-------------+------
-------+-------------+-------------+----------+
```

'显示系统内部策略管理器统计信息'

可以运行此命令来验证每个分区规则的命中数。这对于确定预期规则是否正在被命中而不是另一个预期规则非常有用，例如可能具有更高优先级的隐式丢弃规则。

```
leaf# show system internal policy-mgr stats
Requested Rule Statistics
Rule (4131) DN (sys/actrl/scope-2818048/rule-2818048-s-16410-d-25-f-424) Ingress: 0, Egress: 0,
Pkts: 0  RevPkts: 0
Rule (4156) DN (sys/actrl/scope-2818048/rule-2818048-s-25-d-16410-f-425) Ingress: 0, Egress: 0,
Pkts: 0  RevPkts: 0
```

'show logging ip access-list internal packet-log deny'

可以在iBash级别运行的交换机级别命令，该命令报告ACL（合同）相关丢包和流相关信息，包括：

- VRF
- VLAN-ID
- 源MAC/目标MAC
- 源IP/目标IP
- 源端口/目的端口
- 来源接口

```
leaf# show logging ip access-list internal packet-log deny
[ Tue Oct  1 10:34:37 2019 377572 usecs]: CName: Prod1:VRF1(VXLAN: 2654209), VlanType: Unknown,
Vlan-Id: 0, SMac: 0x000c0c0c0c0c, DMac:0x000c0c0c0c0c, SIP: 192.168.21.11, DIP: 192.168.22.11,
SPort: 0, DPort: 0, Src Intf: Tunnel7, Proto: 1, PktLen: 98
[ Tue Oct  1 10:34:36 2019 377731 usecs]: CName: Prod1:VRF1(VXLAN: 2654209), VlanType: Unknown,
Vlan-Id: 0, SMac: 0x000c0c0c0c0c, DMac:0x000c0c0c0c0c, SIP: 192.168.21.11, DIP: 192.168.22.11,
SPort: 0, DPort: 0, Src Intf: Tunnel7, Proto: 1, PktLen: 98
```

contract_parser

设备上Python脚本，它根据ID执行名称查找时，生成一个输出，该输出将分区规则、过滤器和命中统计信息相关联。此脚本非常有用，因为它采用多步骤流程，并将其转换为单个命令，该命令可过滤为特定EPG/VRF或其他合同相关值。

```
leaf# contract_parser.py
```

```
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]

[7:4131] [vrf:common:default] permit ip tcp tn-Prod1/ap-Services/epg-NTP(16410) tn-Prod1/l3out-
L3Out1/instP-extEpg(25) eq 123  [contract:uni/tn-Prod1/brc-external_to_ntp] [hit=0]
[7:4156] [vrf:common:default] permit ip tcp tn-Prod1/l3out-L3Out1/instP-extEpg(25) eq 123 tn-
Prod1/ap-Services/epg-NTP(16410)  [contract:uni/tn-Prod1/brc-external_to_ntp] [hit=0]
[12:4169] [vrf:common:default] deny,log any tn-Prod1/l3out-L3Out1/instP-extEpg(25) epg:any
[contract:implicit] [hit=0]
[16:4167] [vrf:common:default] permit any epg:any tn-Prod1/bd-Services(32789)
[contract:implicit] [hit=0]
```

数据包分类验证

ELAM

用于检查转发详细信息的ASIC级别报告，在丢包的情况下，它指示丢弃原因。与此部分相关，原因可能是SECURITY_GROUP_DENY（合同策略丢弃）。

分类

APIC上基于Python的实用程序，可以通过ELAM跟踪端到端数据包流。

ELAM助理应用

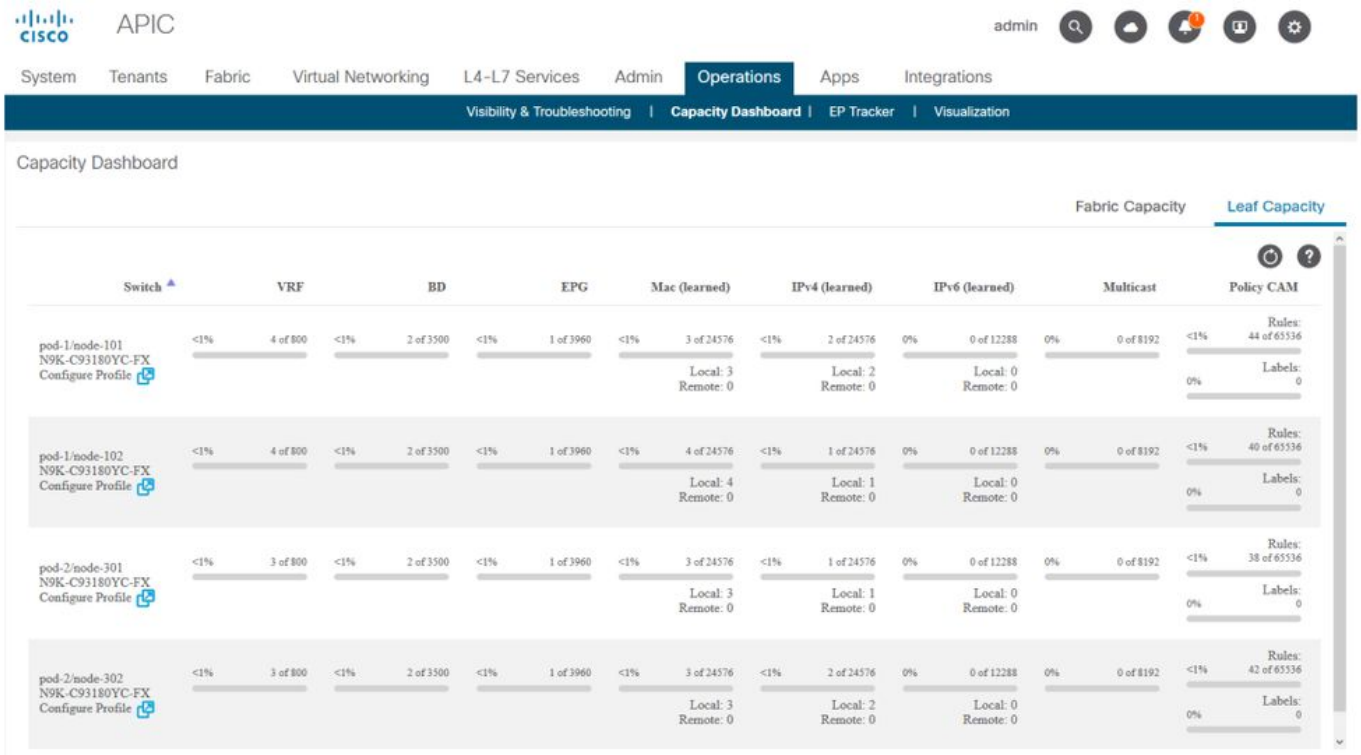
一个APIC应用，它抽象化各种ASIC的复杂性，使转发决策检查更加方便和用户友好。

有关ELAM、Triage和ELAM Assistant工具的其他详细信息，请参阅“交换矩阵内转发”部分

策略CAM使用情况

基于每个枝叶的策略CAM使用率是一个重要监控参数，用于确保交换矩阵处于正常状态。最快速监控方法是使用GUI中的“Capacity Dashboard”并明确检查“Policy Cam”列。

容量控制面板的“枝叶容量”视图



'show platform internal hal health-stats'

此命令可用于验证各种资源限制和使用情况，包括策略CAM。请注意，此命令只能在vsh_lc中运行，因此如果从iBash运行，请使用“ — c”标志传入此命令。

```
leaf8# vsh_lc -c "show platform internal hal health-stats"
|Sandbox_ID: 0  Asic Bitmap: 0x0
|-------------------------------------
 ...
Policy stats:
=============
policy_count                  : 96
max_policy_count              : 65536
policy_otcam_count            : 175
max_policy_otcam_count            : 8192
policy_label_count                : 0
max_policy_label_count            : 0
=============
```

EPG到EPG

通用策略丢弃注意事项

排除两个端点之间的连接问题的方法有很多。以下方法为快速有效地隔离连接问题是否为策略丢弃（合同引起）的结果提供了良好的起点。

在深入讨论之前，需要提出一些高级问题：

- 终端是否处于相同或不同的EPG中？ 位于不同EPG（EPG间）的两个终端之间的流量被隐式拒绝，需要联系才能通信。除非使用的是EPG内隔离，否则隐式允许同一EPG（EPG内）内的两个终端之间的流量。
- VRF是实施还是不实施？ 当VRF处于enforced模式时（在VRF内），两个不同EPG中的终端需

要合同才能通信。当VRF处于**unenforced**模式时（在VRF内），ACI交换矩阵将允许所有流量通过属于未实施VRF的多个EPG，而不管应用的ACI合同如何。

方法

有了各种可用的工具，有一些工具比其他工具更适合和更便于使用，这取决于已知的有关受影响流量的信息级别。

ACI交换矩阵中数据包的完整路径是否已知（入口枝叶、出口枝叶……）？

- 如果答案为是，应使用ELAM Assistant确定源或目标交换机上的丢弃原因。
- 如果答案为否，则Visibility & Troubleshooting、fTriage、contract_parser、Tenant视图中的Operational选项卡和iBash命令将有助于缩小数据包的路径或更好地了解丢弃原因。

请注意，本部分不会详细讨论fTriage工具。有关使用此工具的更多详细信息，请参阅“交换矩阵内转发”一章。

请注意，虽然可视性和故障排除有助于快速了解两个端点之间的数据包丢弃位置，但fTriage显示更多深入信息，以便进一步进行故障排除。即fTriage将帮助识别接口、丢弃原因和有关受影响流量的其他低级详细信息

.

此示例场景将展示如何排除两个终端之间的策略丢弃故障：192.168.21.11 和 192.168.23.11

假设在这两个端点之间发生丢包，将使用以下故障排除工作流程确定问题的根本原因：

确定流量流中涉及的src/dst枝叶：

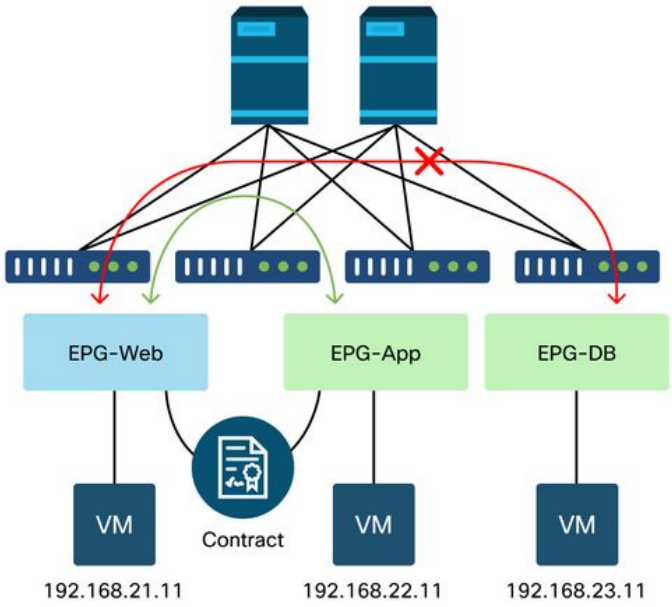
1. 使用**可视性和故障排除**跟踪数据包流并确定哪个设备正在丢弃数据包。
2. 在所选设备上运行命令show logging ip access-list internal packet-log deny。 如果拒绝并记录具有其中一个相关IP地址的数据包，**packet-log**将按点击次数打印相关终端和合同名称。
3. 在源枝叶和目标枝叶上使用命令“contract_parser.py —vrf <tenant>:<VRF>”来观察已配置合同的命中计数： 如果数据包到达源交换机或目标交换机上的合同，相关合同的计数器将会增加在许多流可能达到同一规则（两个相关的EPG之间的多个终端/流）的情况下，此方法比IP访问列表内部数据包日志的粒度更小。

上述步骤将在下一段进一步介绍。

EPG到EPG的故障排除示例场景

此示例场景将展示如何排除两个终端之间的策略丢弃故障：EPG-Web中的192.168.21.11和EPG-DB中的192.168.23.11。

拓扑

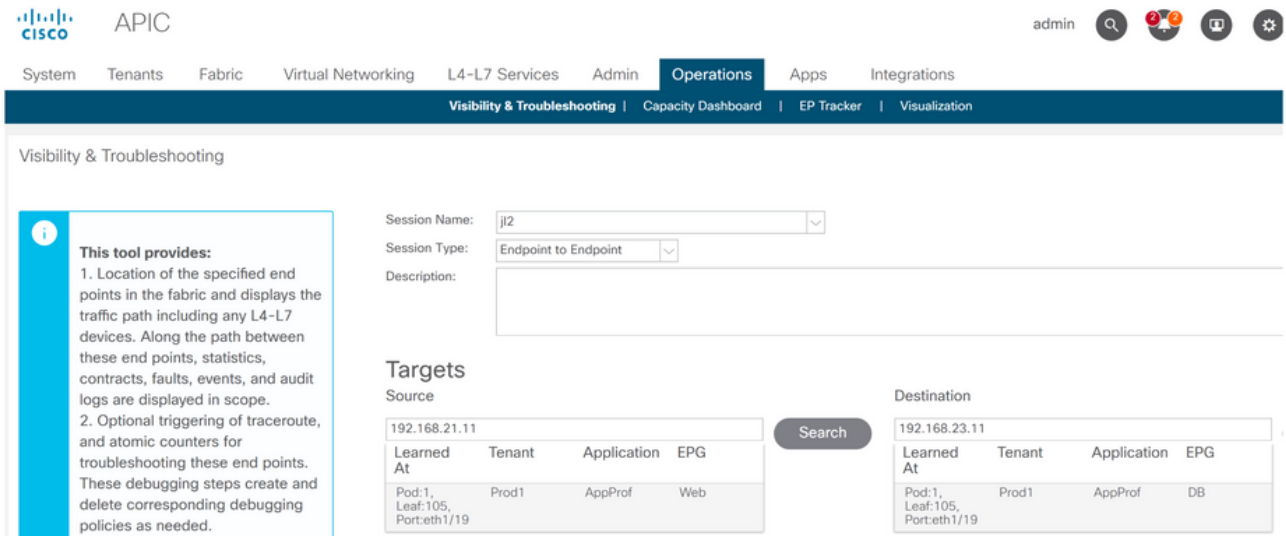


确定数据包丢弃中涉及的源枝叶交换机和目标枝叶交换机

可视性和故障排除

可视性和故障排除工具将帮助可视化特定EP到EP流发生数据包丢弃的交换机，并确定可能丢弃数据包的位置。

可视性与故障排除的配置

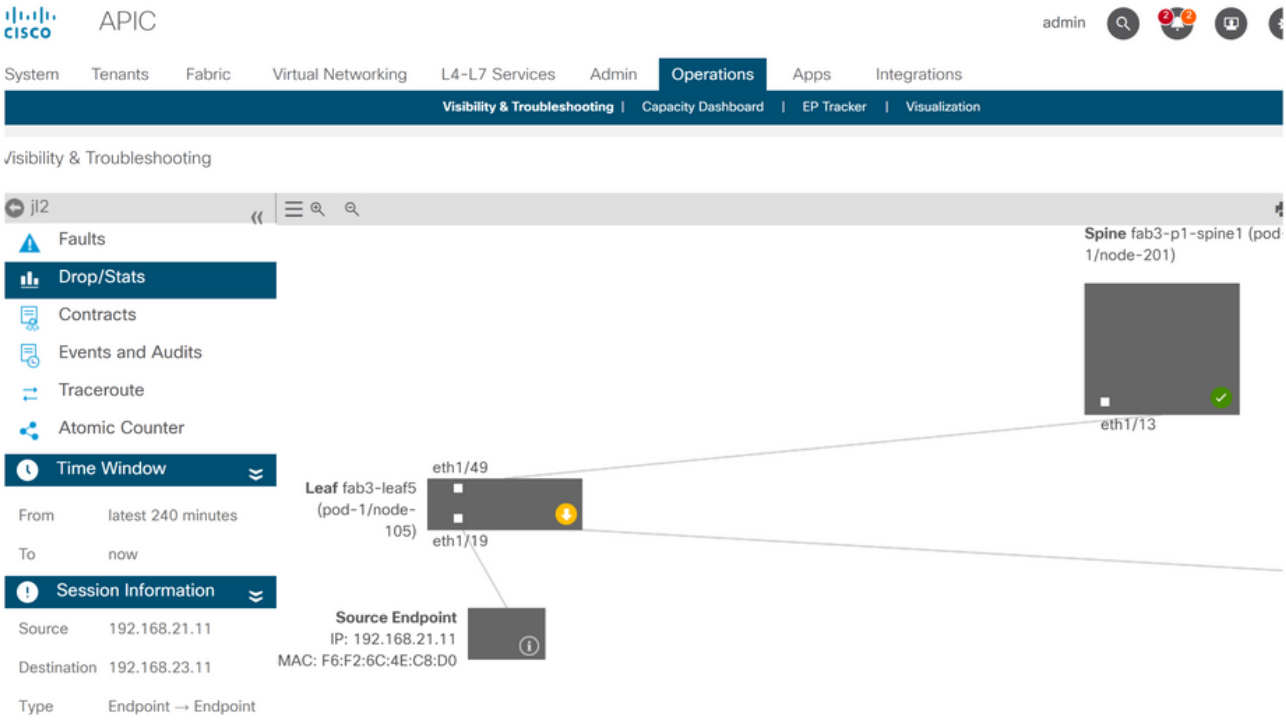


配置会话名称、源和目标终端。然后单击“提交”或“生成报告”。

该工具将自动查找交换矩阵中的终端，并提供有关租户、应用配置文件和EP所属的EPG的信息。

在本例中，它将发现EP属于租户Prod1，它们属于同一应用配置文件“AppProf”，并被分配到不同的EPG:“Web”和“DB”。

丢弃标识

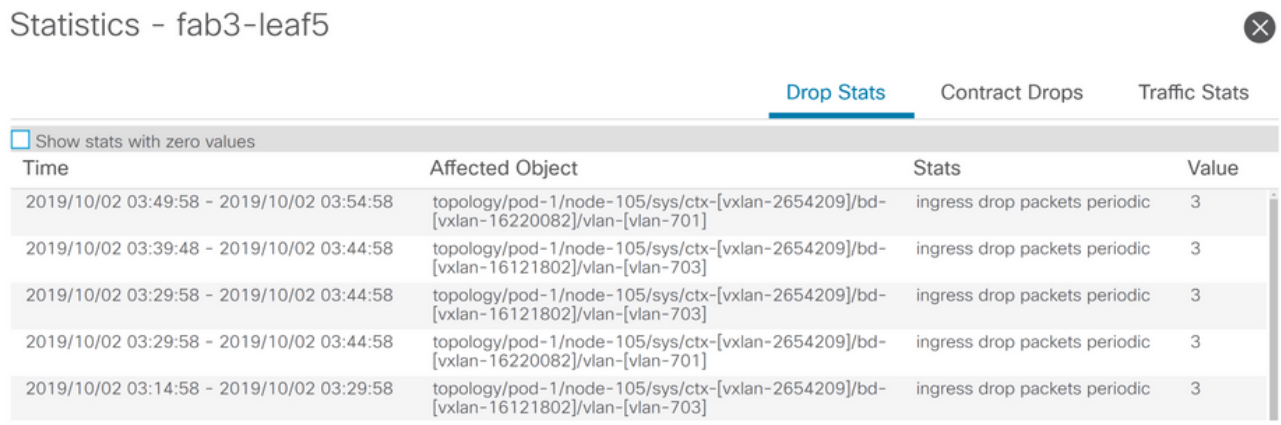


该工具将自动显示故障排除方案的拓扑。在这种情况下，两个终端恰好连接到同一枝叶交换机。

通过导航到Drop/Stats子菜单，用户可以查看有关枝叶或主干上的常规丢包。请参阅本书“交换矩阵内转发”一章中的“接口丢弃”部分，了解更多有关了解哪些丢弃相关的信息。

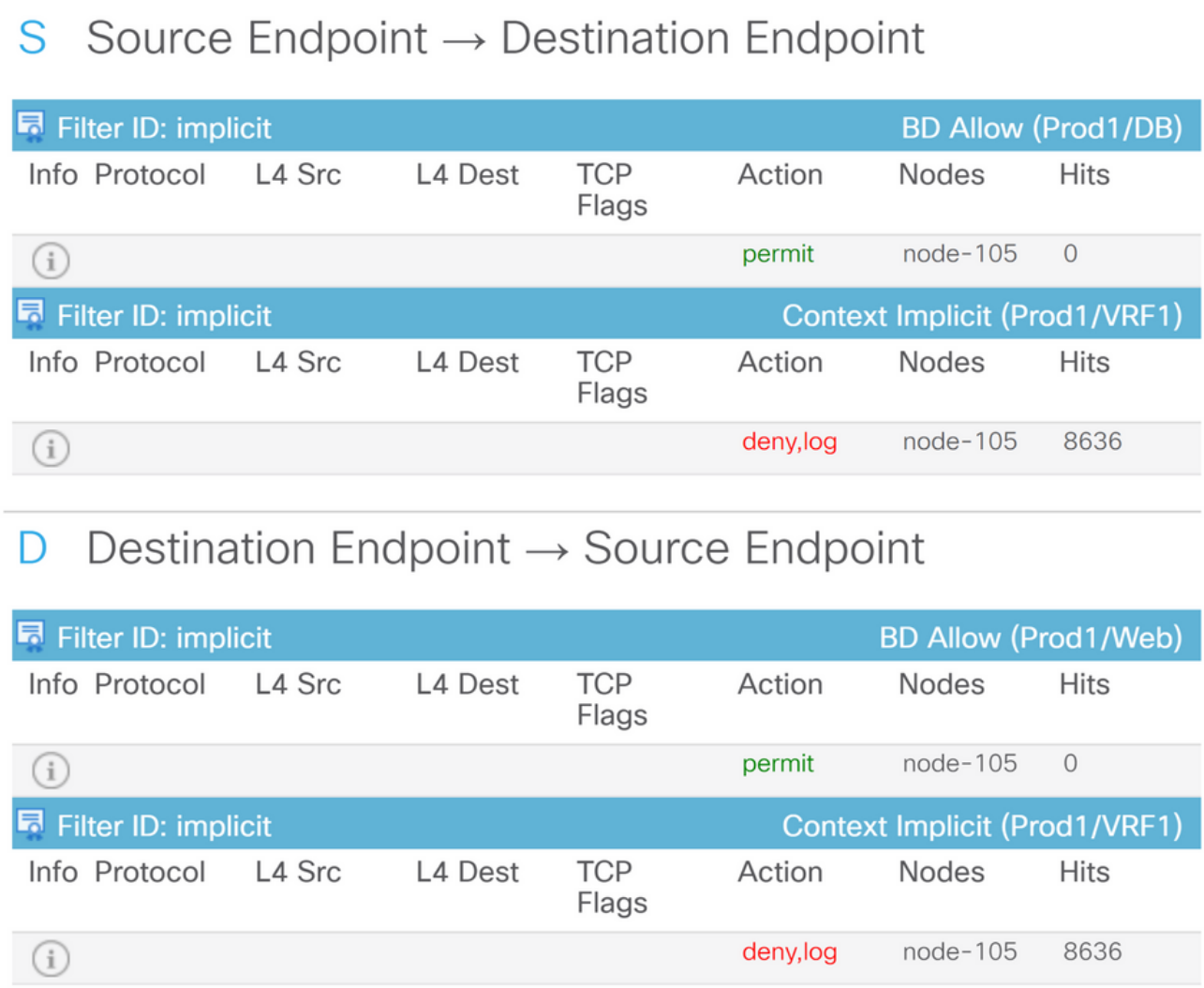
其中许多丢弃是预期行为，可以忽略。

删除详细信息



通过使用交换机图上的黄色“丢弃的数据包”按钮向下钻取以丢弃详细信息，用户可以查看有关丢弃的数据流的详细信息。

合同详细信息



通过导航到Contracts（合同）子菜单，用户可以确定哪个合同导致EPG之间的策略下拉。在本例中，“隐式”(Implicit)表示拒绝Prod1/VRF1，它显示一些命中数。这并不一定意味着指定的流（192.168.21.11和192.168.23.11）达到此隐式拒绝。如果Context Implicit deny规则的Hits数增加，则表明Prod1/DB和Prod1/Web之间存在未命中任何合同的流量，因此被Implicit deny规则丢弃。

在Tenant（租户）>的Application Profile Topology（应用配置文件拓扑）视图中，选择左侧的Application Profile name（应用配置文件名称）> Topology（拓扑），可以验证将哪些合同应用于数据库EPG。在这种情况下，没有将合同分配给EPG:

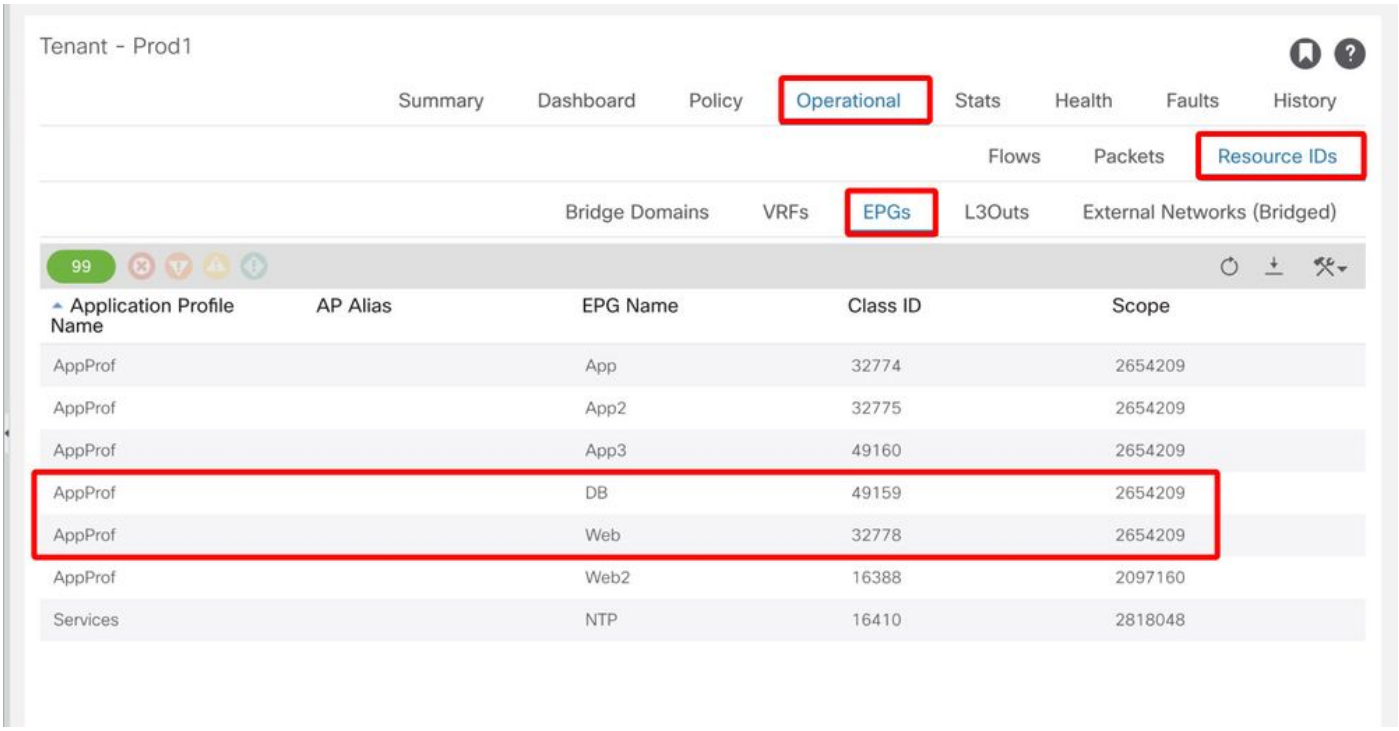
合同可视化

由于源和目的EPG已知，因此还可以确定其他相关信息，例如：

- 受影响终端的src/dst **EPG pcTag**。pcTag是用于使用分区规则标识EPG的类ID。
- 受影响终端的src/dst **VRFVNID**，也称为**范围**。

通过打开租户>在左侧选择租户名称>操作>资源ID > EPG，可以从APIC GUI中轻松检索类ID和范围

用于查找EPG pcTag和范围的租户资源ID



在本例中，类ID和范围是：

- Web EPG pcTag 32778
- Web EPG作用域2654209
- DB EPG pcTag 49159
- DB EPG作用域2654209

验证应用于正在故障排除的流量的策略

iBash

验证ACI枝叶上丢弃的数据包的有趣工具是iBash命令行：'show logging ip access-list internal packet-log deny':

```
leaf5# show logging ip access-list internal packet-log deny | grep 192.168.21.11
[2019-10-01T14:25:44.746528000+09:00]: CName: Prod1:VRF1(VXLAN: 2654209), VlanType: FD_VLAN,
Vlan-Id: 114, SMac: 0xf6f26c4ec8d0, DMac:0x0022bdf819ff, SIP: 192.168.21.11, DIP: 192.168.23.11,
SPort: 0, DPort: 0, Src Intf: Ethernet1/19, Proto: 1, PktLen: 126
[2019-10-01T14:25:44.288653000+09:00]: CName: Prod1:VRF1(VXLAN: 2654209), VlanType: FD_VLAN,
Vlan-Id: 116, SMac: 0x3e2593f0eded, DMac:0x0022bdf819ff, SIP: 192.168.23.11, DIP: 192.168.21.11,
SPort: 0, DPort: 0, Src Intf: Ethernet1/19, Proto: 1, PktLen: 126
```

根据前面的输出，可以看到，在枝叶交换机上，从EP 192.168.23.11到192.168.21.11的许多ICMP数据包已丢弃。

contract_parser工具将帮助验证应用于终端关联的VRF的实际策略：

```
leaf5# contract_parser.py --vrf Prod1:VRF1
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]

[7:5159] [vrf:Prod1:VRF1] permit ip tcp tn-Prod1/ap-App1/epg-App(32771) eq 5000 tn-Prod1/ap-
```

```
App1/epg-Web(32772)  [contract:uni/tn-Prod1/brc-web_to_app] [hit=0]
[7:5156] [vrf:Prod1:VRF1] permit ip tcp tn-Prod1/ap-App1/epg-Web(32772) tn-Prod1/ap-App1/epg-
App(32771) eq 5000  [contract:uni/tn-Prod1/brc-web_to_app] [hit=0]
[16:5152] [vrf:Prod1:VRF1] permit any epg:any tn-Prod1/bd-Web(49154) [contract:implicit] [hit=0]
[16:5154] [vrf:Prod1:VRF1] permit arp epg:any epg:any [contract:implicit] [hit=0]
[21:5155] [vrf:Prod1:VRF1] deny,log any epg:any epg:any [contract:implicit] [hit=38,+10]
[22:5153] [vrf:Prod1:VRF1] deny,log any epg:any pfx-0.0.0.0/0(15) [contract:implicit] [hit=0]
```

这也可以通过在枝叶中编程的分区规则验证交换机实施的策略。

```
leaf5# show zoning-rule scope 2654209
+---------+--------+--------+----------+----------------+---------+----------+------------+-----
-----+-------------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope   |    Name    |
Action  |         Priority        |
+---------+--------+--------+----------+----------------+---------+----------+------------+-----
-----+-------------------------+
|   5155  |   0    |   0    | implicit |    uni-dir     | enabled | 2654209  |            |
deny,log |     any_any_any(21)     |
|   5159  | 32771  | 32772  |   411    | uni-dir-ignore | enabled | 2654209  | web_to_app |
permit  |      fully_qual(7)      |
|   5156  | 32772  | 32771  |   410    |     bi-dir     | enabled | 2654209  | web_to_app |
permit  |      fully_qual(7)      |
+---------+--------+--------+----------+----------------+---------+----------+------------+-----
-----+-------------------------+
```

如可视性与故障排除工具、contract_parser工具和分区规则所示，输出确认在故障排除过程中，源和目标EPG之间没有合同。很容易假设丢弃的数据包与隐式拒绝规则5155匹配。

ELAM捕获

ELAM捕获提供用于检查转发详细信息的ASIC级别报告，在数据包被丢弃的情况下，该报告指示丢弃原因。当丢弃原因为策略丢弃时（如本场景所示），ELAM捕获的输出将如下所示。

请注意，本章不讨论设置ELAM捕获的详细信息，请参阅“交换矩阵内转发”一章。

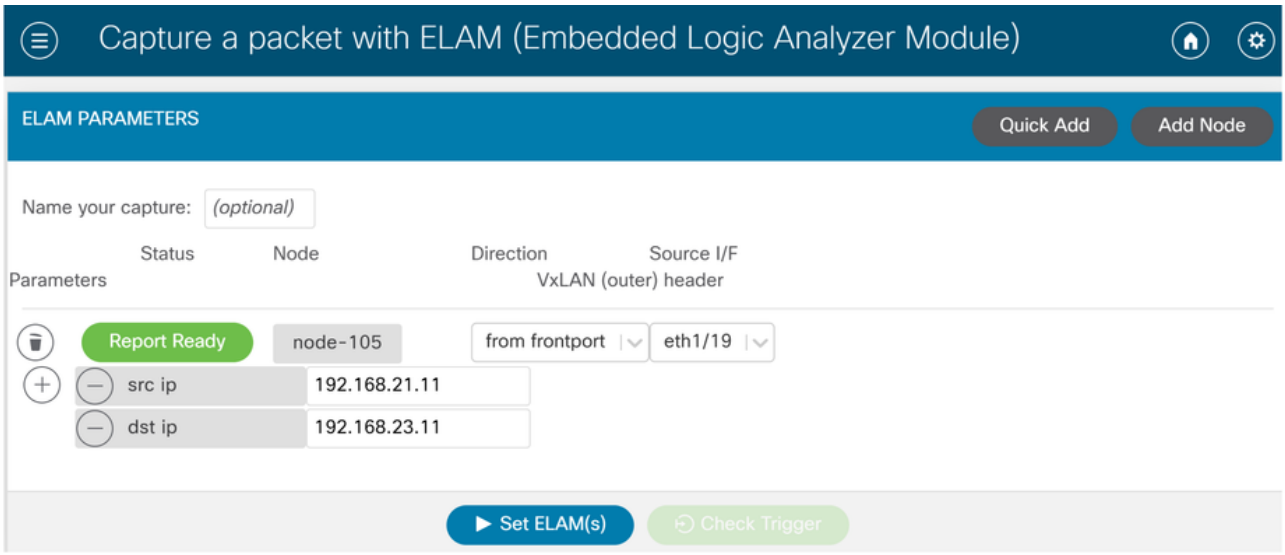
```
leaf5# vsh_lc
module-1# debug platform internal tah elam asic 0
module-1(DBG-elam)# trigger init in-select 6 out-select 0
module-1(DBG-elam)# trigger reset
module-1(DBG-elam-insel6)# set outer ipv4 src_ip 192.168.21.11 dst_ip 192.168.23.11
module-1(DBG-elam-insel6)# start
module-1(DBG-elam-insel6)# status

ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed

module-1(DBG-elam-insel6)# ereport | grep reason
RW drop reason                          : SECURITY_GROUP_DENY
LU drop reason                          : SECURITY_GROUP_DENY
   pkt.lu_drop_reason: 0x2D
```

上述ELAM报告清楚地显示由于策略丢弃而丢弃了数据包：'SECURITY_GROUP_DENY'

ELAM助理：

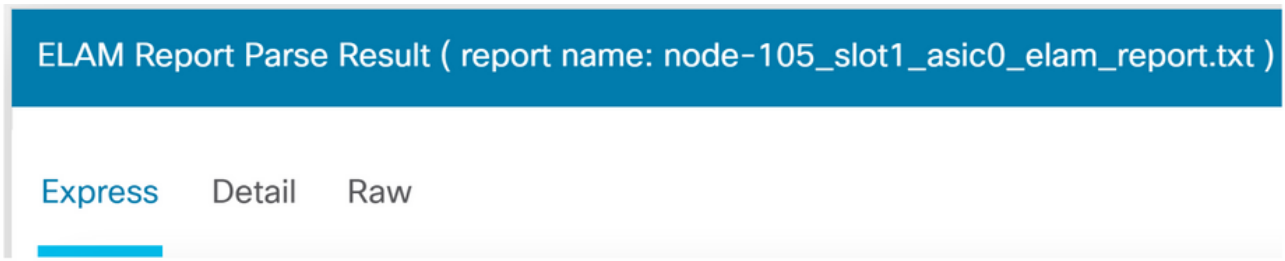
通过APIC GUI上的ELAM Assistant应用可显示与ELAM捕获完全相同的结果。

配置



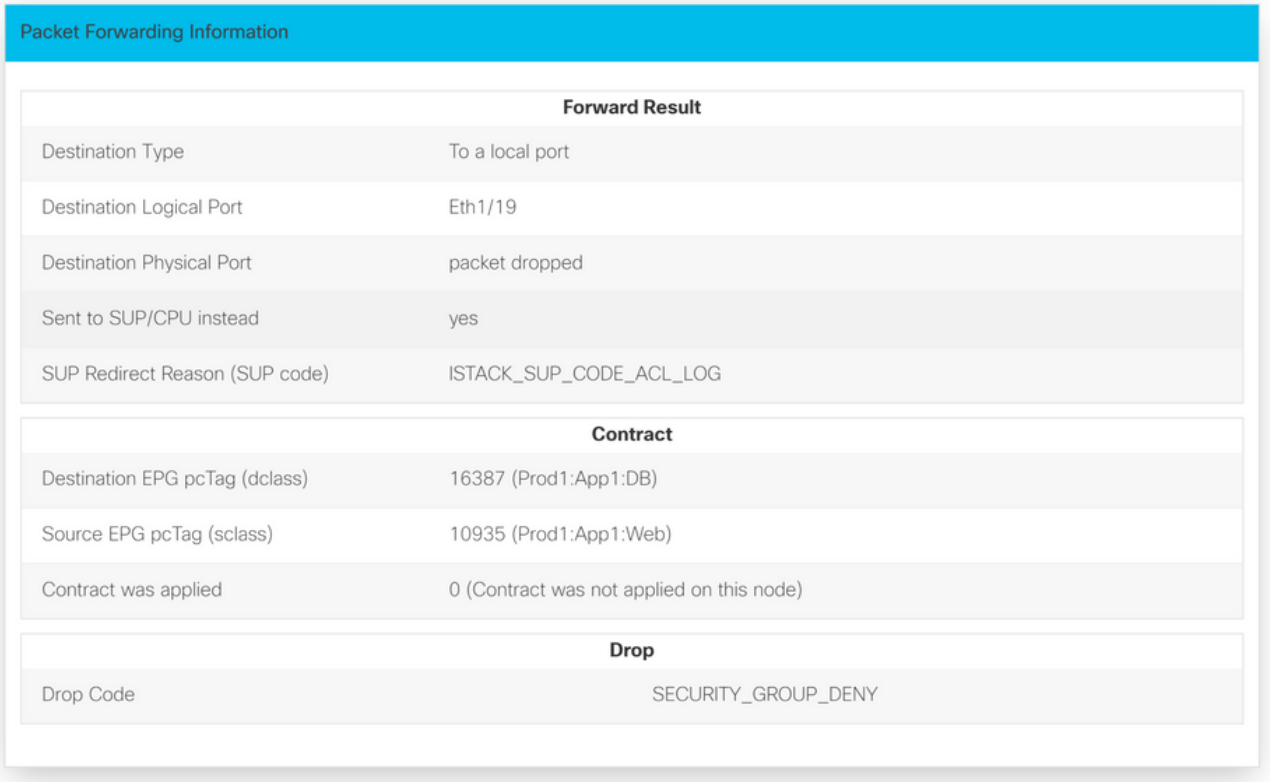
通常，用户将配置相关流的源和目标详细信息。在本示例中，源IP用于捕获目标EPG中流向与源EPG没有合同关系的终端的流量。

Elam Assistant Express报告



通过ELAM Assistant可以查看三个级别的输出。这些是Express、Detail和Raw。

Elam Assistant Express报告（续）



在Express Result（快速结果）下，丢弃代码原因SECURITY_GROUP_DENY表示丢弃是合同命中的结果。

首选组

关于合同首选组

在配置了合同首选组的VRF中，有两种类型的策略实施可用于EPG:

- 包含的EPG:如果EPG拥有合同首选组的成员资格，则无需合同即可自由相互通信。这基于source-any-destination-any-permit默认规则。
- 排除的EPG:非首选组成员的EPG需要合同才能相互通信。否则，将应用排除EPG和任何EPG之间的拒绝规则。

合同首选组功能可更好地控制VRF中EPG之间的通信。如果VRF中的大部分EPG应该具有开放式通信，但少数应仅与其他EPG具有有限的通信，请配置合同首选组和合同与过滤器的组合，以更加精确地控制EPG间通信。

如果存在覆盖source-any-destination-any-deny默认规则的合同，则从首选组排除的EPG只能与其他EPG通信。

合同首选组编程

从本质上讲，合同首选组与常规合同相反。对于常规合同，显式permit zoning-rules使用VRF范围的隐式deny zoning-rule进行编程。对于首选组，使用最高数值优先级值编程隐式PERMIT分区规则，而编程特定DENY分区规则以禁止来自非首选组成员的EPG的流量。因此，首先评估拒绝规则，如果流与这些规则不匹配，则隐式允许流。

对于首选组之外的每个EPG，始终有一对明确的deny zoning-rules:

- 从非首选组成员到任意pcTag的一个（值0）。
- 另一个从任何pcTag（值0）到非首选组成员。

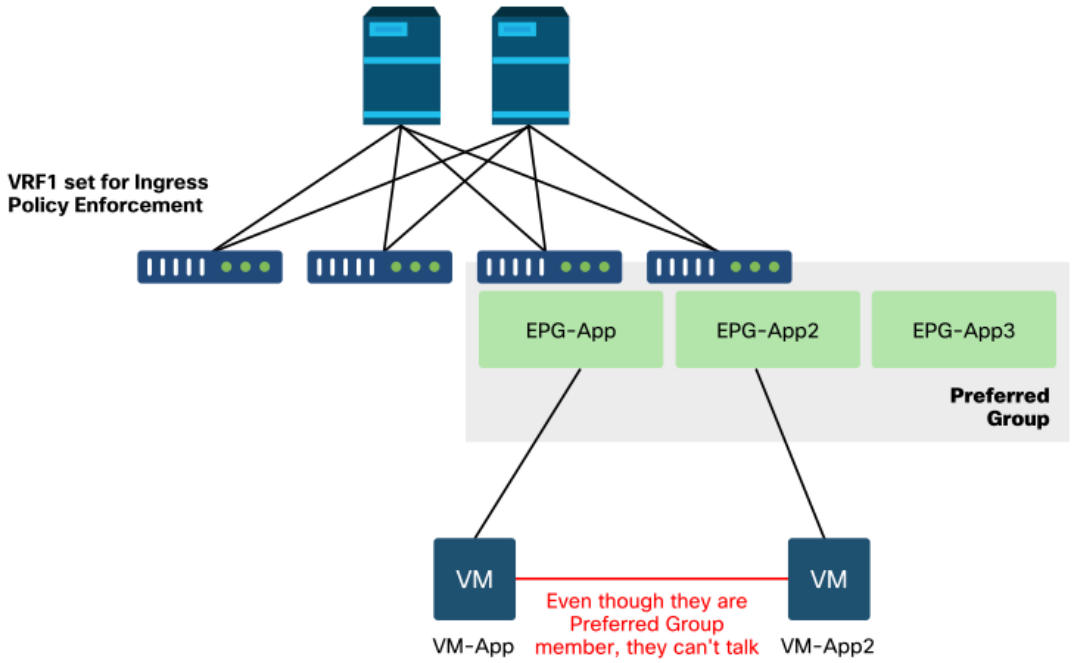
首选组故障排除场景

下图显示了一个逻辑拓扑，其中EPG应用、应用2和应用3均配置为首选组成员。

VM-App是EPG-App的一部分，VM-App2是EPG-App2的一部分。App和App2 EPG都应是首选的一部分，因此可以自由通信。

VM-App在TCP端口6000上向VM-App2发起流量。EPG-App和EPG-App2都是VRF1的首选组成员。VM-App2从TCP端口6000上从未收到任何数据包。

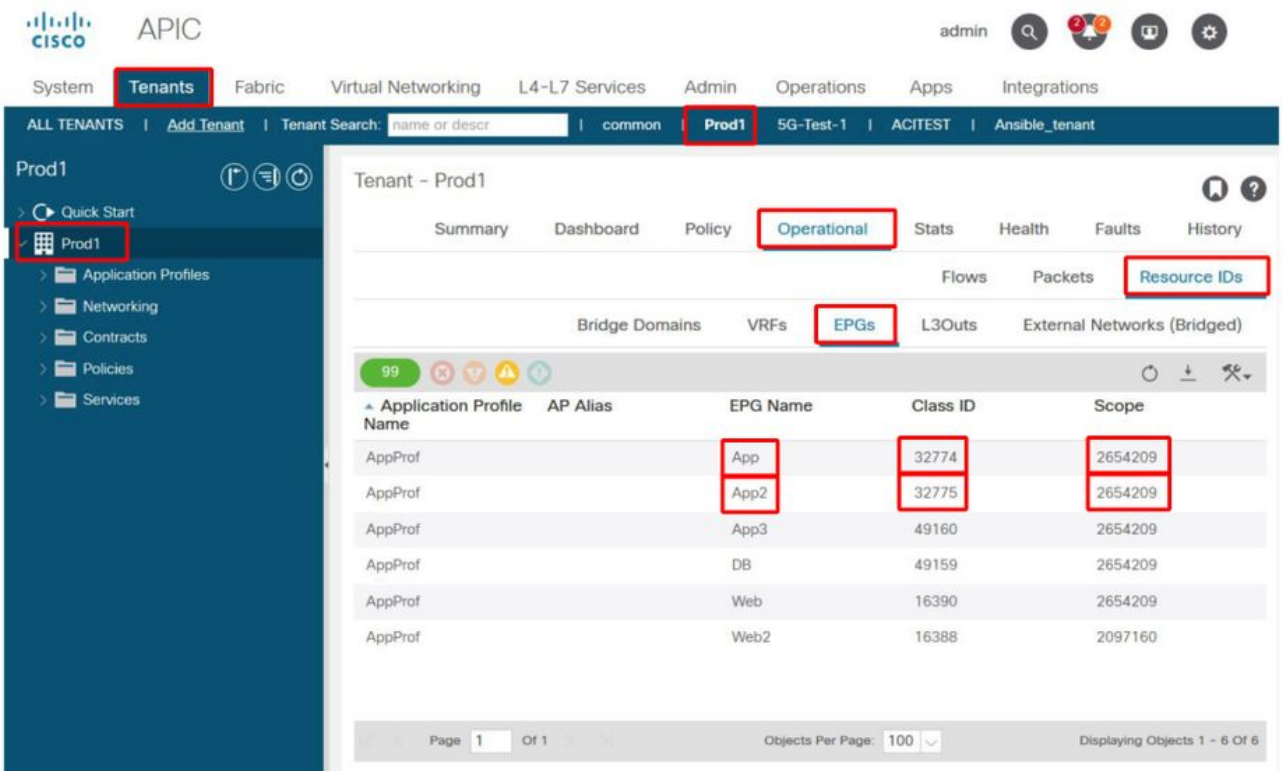
拓扑



工作流程

1.查找EPG APP的pcTag及其VRF VNID/Scope

EPG和VRF pcTags



2.在入口枝叶上使用contract_parser.py验证合同编程

使用contract_parser.py和/或“show zoning-rule”命令并指定VRF

```
fab3-leaf8# show zoning-rule scope 2654209
+---------+--------+--------+----------+---------+---------+---------+------+----------+--------
--------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |   Dir   |  operSt |  Scope  | Name |  Action
|          Priority          |
+---------+--------+--------+----------+---------+---------+---------+------+----------+--------
--------------------+
|   4165  |   0    |   0    | implicit | uni-dir | enabled | 2654209 |      |  permit  |
grp_any_any_any_permit(20) |
|   4160  |   0    |   0    | implarp  | uni-dir | enabled | 2654209 |      |  permit  |
any_any_filter(17)     |
|   4164  |   0    |   15   | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_any_dest_any_deny(19)  |
|   4176  |   0    | 16386  | implicit | uni-dir | enabled | 2654209 |      |  permit  |
any_dest_any(16)       |
|   4130  | 32770  |   0    | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_src_any_any_deny(18)   |
|   4175  | 49159  |   0    | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_src_any_any_deny(18)   |
|   4129  |   0    | 49159  | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_any_dest_any_deny(19)  |
|   4177  | 32778  |   0    | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_src_any_any_deny(18)   |
|   4128  |   0    | 32778  | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_any_dest_any_deny(19)  |
|   4178  | 32775  |   0    | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_src_any_any_deny(18)   |
|   4179  |   0    | 32775  | implicit | uni-dir | enabled | 2654209 |      | deny,log |
grp_any_dest_any_deny(19)  |
+---------+--------+--------+----------+---------+---------+---------+------+----------+--------
--------------------+

fab3-leaf8# contract_parser.py --vrf Prod1:VRF1
```

```
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]
[16:4176] [vrf:Prod1:VRF1] permit any epg:any tn-Prod1/bd-App(16386) [contract:implicit] [hit=0]
[16:4160] [vrf:Prod1:VRF1] permit arp epg:any epg:any [contract:implicit] [hit=0]
[18:4130] [vrf:Prod1:VRF1] deny,log any tn-Prod1/vrf-VRF1(32770) epg:any [contract:implicit]
[hit=?]
[18:4178] [vrf:Prod1:VRF1] deny,log any epg:32775 epg:any [contract:implicit] [hit=?]
[18:4177] [vrf:Prod1:VRF1] deny,log any epg:32778 epg:any [contract:implicit] [hit=?]
[18:4175] [vrf:Prod1:VRF1] deny,log any epg:49159 epg:any [contract:implicit] [hit=?]
[19:4164] [vrf:Prod1:VRF1] deny,log any epg:any pfx-0.0.0.0/0(15) [contract:implicit] [hit=0]
[19:4179] [vrf:Prod1:VRF1] deny,log any epg:any epg:32775 [contract:implicit] [hit=?]
[19:4128] [vrf:Prod1:VRF1] deny,log any epg:any epg:32778 [contract:implicit] [hit=?]
[19:4129] [vrf:Prod1:VRF1] deny,log any epg:any epg:49159 [contract:implicit] [hit=?]
[20:4165] [vrf:Prod1:VRF1] permit any epg:any epg:any [contract:implicit] [hit=65]
```

检查上述输出，观察到具有最高优先级20的隐式permit条目 — ruleId 4165。除非有具有较低优先级的显式拒绝规则禁止流量，否则此隐式允许规则将导致允许所有流量。

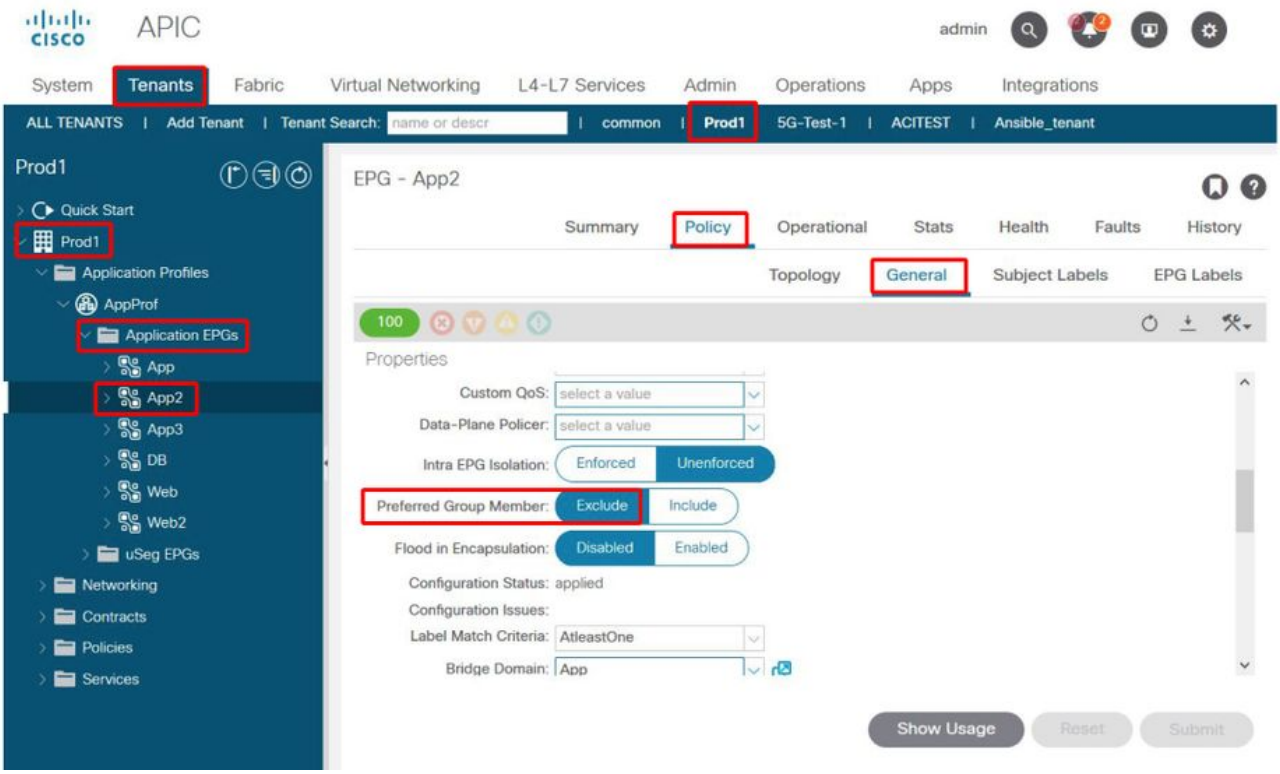
此外，对于pcTag 32775（即EPG App2的pcTag），观察到两个显式拒绝规则。这两个显式拒绝分区规则禁止从任何EPG到EPG App2的流量，反之亦然。这些规则的优先级为18和19，因此它们将优先于默认的允许规则。

结论是，EPG App2不是首选组成员，因为已观察到显式拒绝规则。

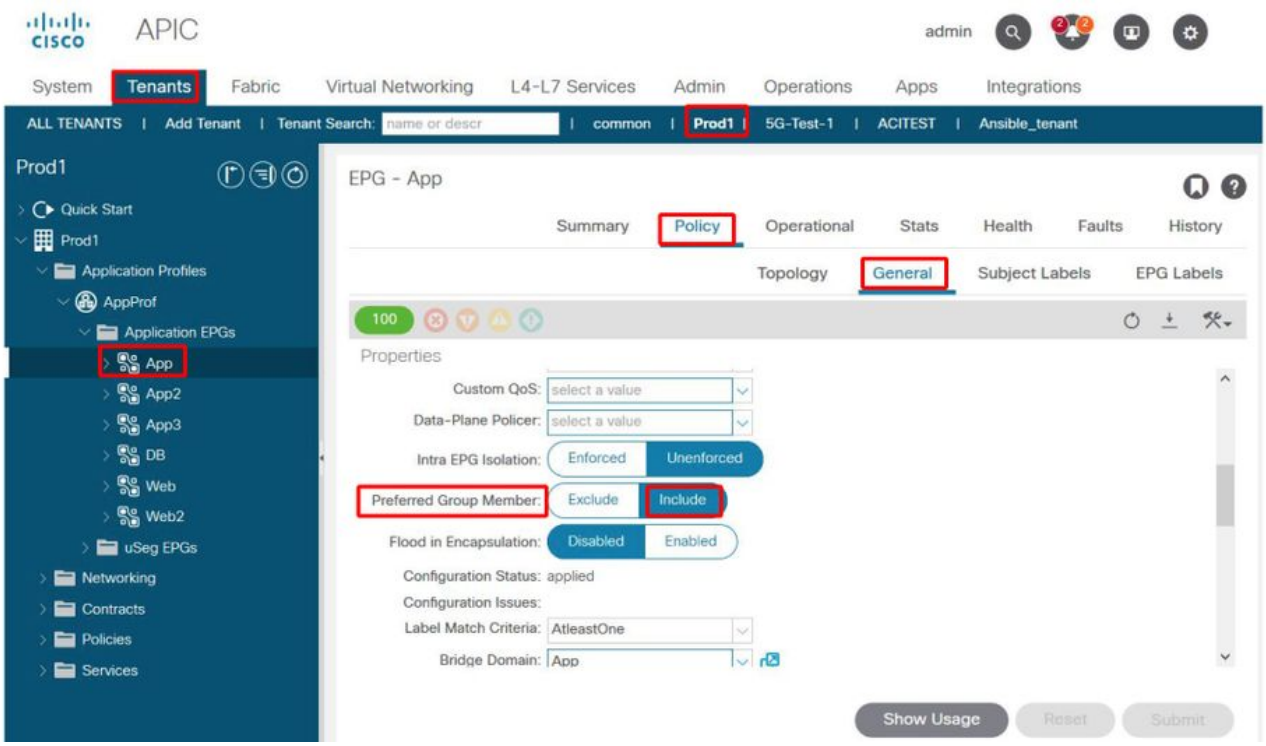
3.检验EPG首选组成员配置

导航APIC GUI并检查EPG App2和EPG应用首选组成员配置，在下图中，请参阅EPG App2未配置为首选组成员。

EPG App2 — 排除首选组成员设置



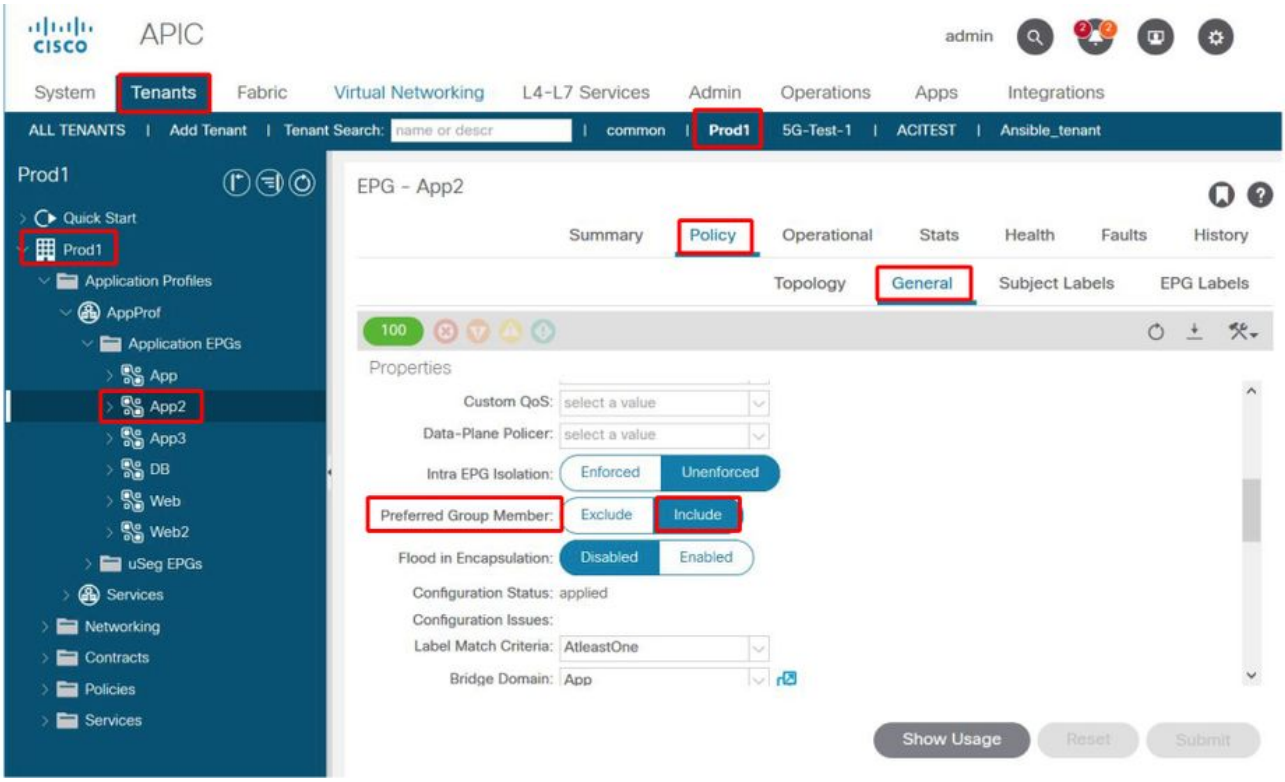
EPG应用 — 包括首选组成员设置



4.将EPG App2设置为首选组成员

更改App2 EPG配置可使首选组作为首选组的一部分自由通信。

EPG App2 — 包括首选组成员设置



5.使用src EP所在的枝叶上的contract_parser.py重新验证合同编程

再次使用contract_parser.py并指定VRF名称以验证EPG App2的显式拒绝规则是否已消失。

```
fab3-leaf8# contract_parser.py --vrf Prod1:VRF1
```

```
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]
[16:4176] [vrf:Prod1:VRF1] permit any epg:any tn-Prod1/bd-App(16386) [contract:implicit] [hit=0]
[16:4160] [vrf:Prod1:VRF1] permit arp epg:any epg:any [contract:implicit] [hit=0]
[18:4175] [vrf:Prod1:VRF1] deny,log any epg:16390 epg:any [contract:implicit] [hit=0]
[18:4167] [vrf:Prod1:VRF1] deny,log any epg:23 epg:any [contract:implicit] [hit=0]
[18:4156] [vrf:Prod1:VRF1] deny,log any tn-Prod1/vrf-VRF1(32770) epg:any [contract:implicit]
[hit=0]
[18:4168] [vrf:Prod1:VRF1] deny,log any epg:49159 epg:any [contract:implicit] [hit=0]
[19:4164] [vrf:Prod1:VRF1] deny,log any epg:any pfx-0.0.0.0/0(15) [contract:implicit] [hit=0]
[19:4169] [vrf:Prod1:VRF1] deny,log any epg:any epg:16390 [contract:implicit] [hit=0]
[19:4159] [vrf:Prod1:VRF1] deny,log any epg:any epg:23 [contract:implicit] [hit=0]
[19:4174] [vrf:Prod1:VRF1] deny,log any epg:any epg:49159 [contract:implicit] [hit=0]
[20:4165] [vrf:Prod1:VRF1] permit any epg:any epg:any [contract:implicit] [hit=65]
```

上述输出中不再显示EPG App2及其pcTag32775的显式拒绝规则。这意味着EPG应用和EPG应用2中的EP之间的流量现在将匹配隐式允许规则(ruleId 4165)，最高优先级为20。

vzAny到EPG

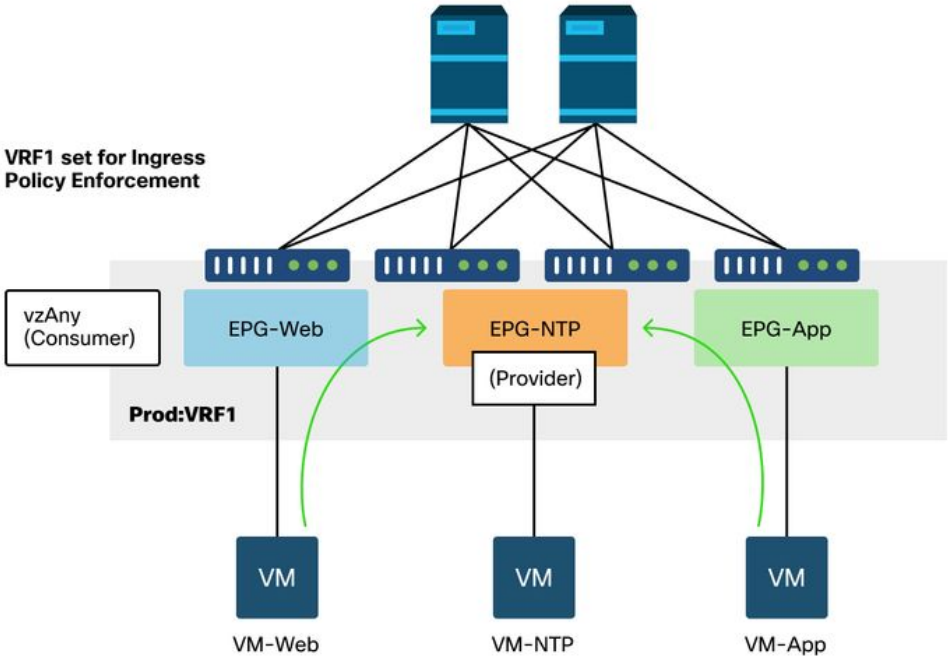
关于vzAny

在一个或多个EPG之间配置合同时，可以将合同配置为消耗关系或提供的关系。当EPG数量增加时，它们之间的合同关系量也会增加。某些常见使用案例要求所有EPG与另一个特定EPG交换流量。此类用例可以是EPG，其中包含需要由相同VRF（例如NTP或DNS）内的所有其他EPG使用的服务的EP。vzAny可降低在配置所有EPG与提供所有其他EPG要使用的服务的特定EPG之间的合同关系时的运营开销。此外，由于每个vzAny合同关系仅添加了2个分区规则，因此vzAny允许在枝叶交换机上更有效地使用安全策略CAM。

使用案例示例

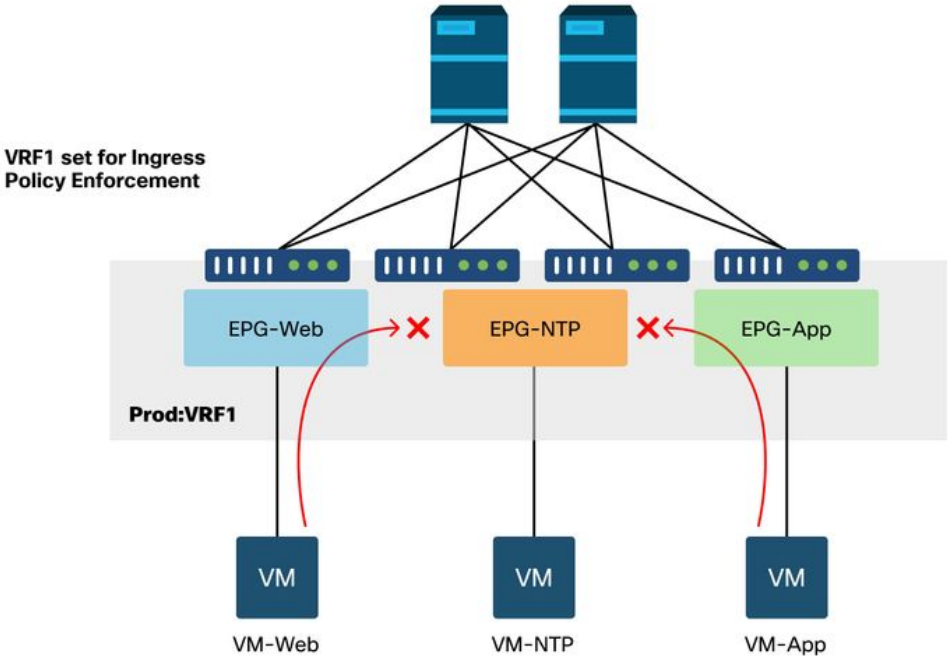
下图描述了这样的使用案例，其中EPG Web中的VM-Web和VM-App分别需要使用EPG-NTP中的VM-NTP的NTP服务。vzAny允许VRF Prod:VRF1中的每个EPG从EPG NTP使用NTP服务，而不是在EPG NTP上配置提供的合同，并且随后将该合同与EPG Web和App上使用的合同相同。

vzAny - VRF中的任何EPG Prod:VRF1都可以使用EPG NTP中的NTP服务



考虑这样一个场景：当使用NTP服务的EPG之间没有合同时，它们之间会观察到丢弃。

故障排除场景 — 如果没有合同，则丢弃流量

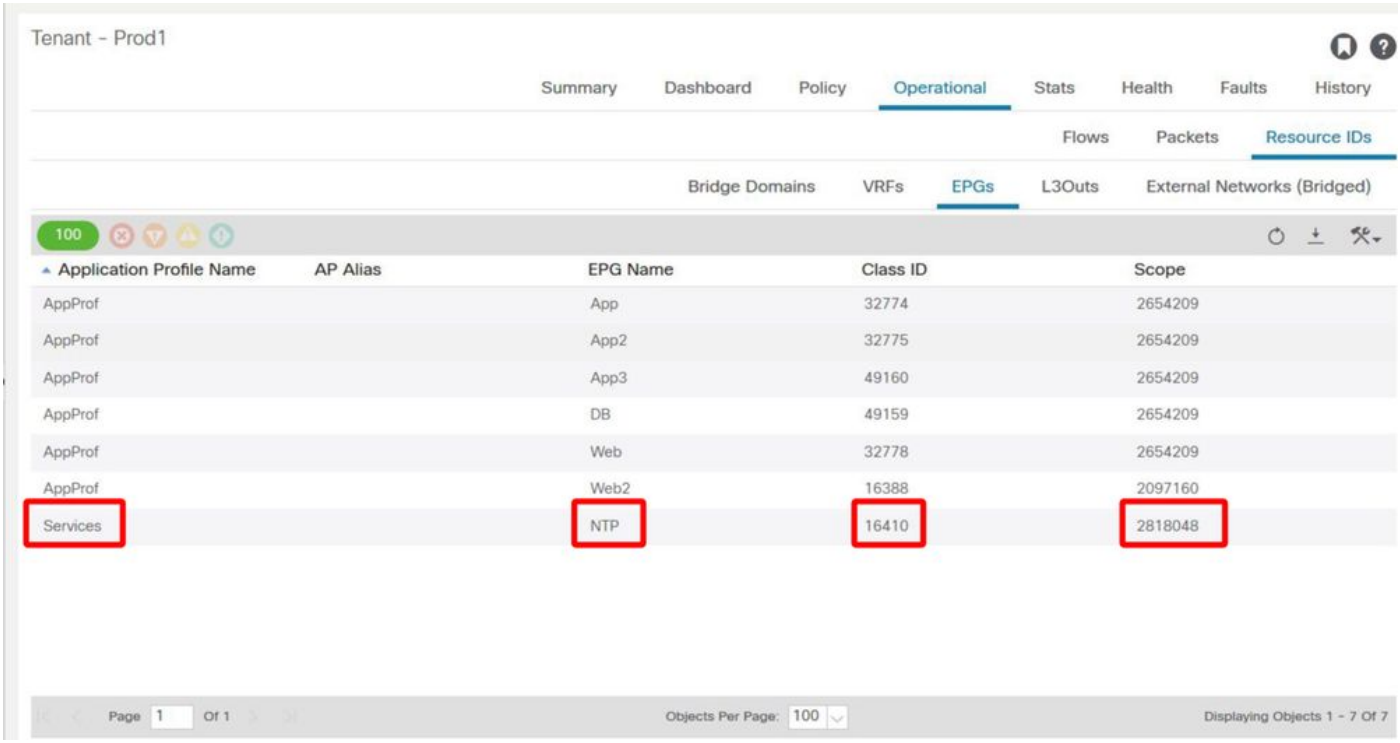


工作流程

1.查找EPG NTP的pcTag及其VRF VNID/Scope

“Tenant > Operational > Resource IDs > EPGs”允许查找pcTag和范围

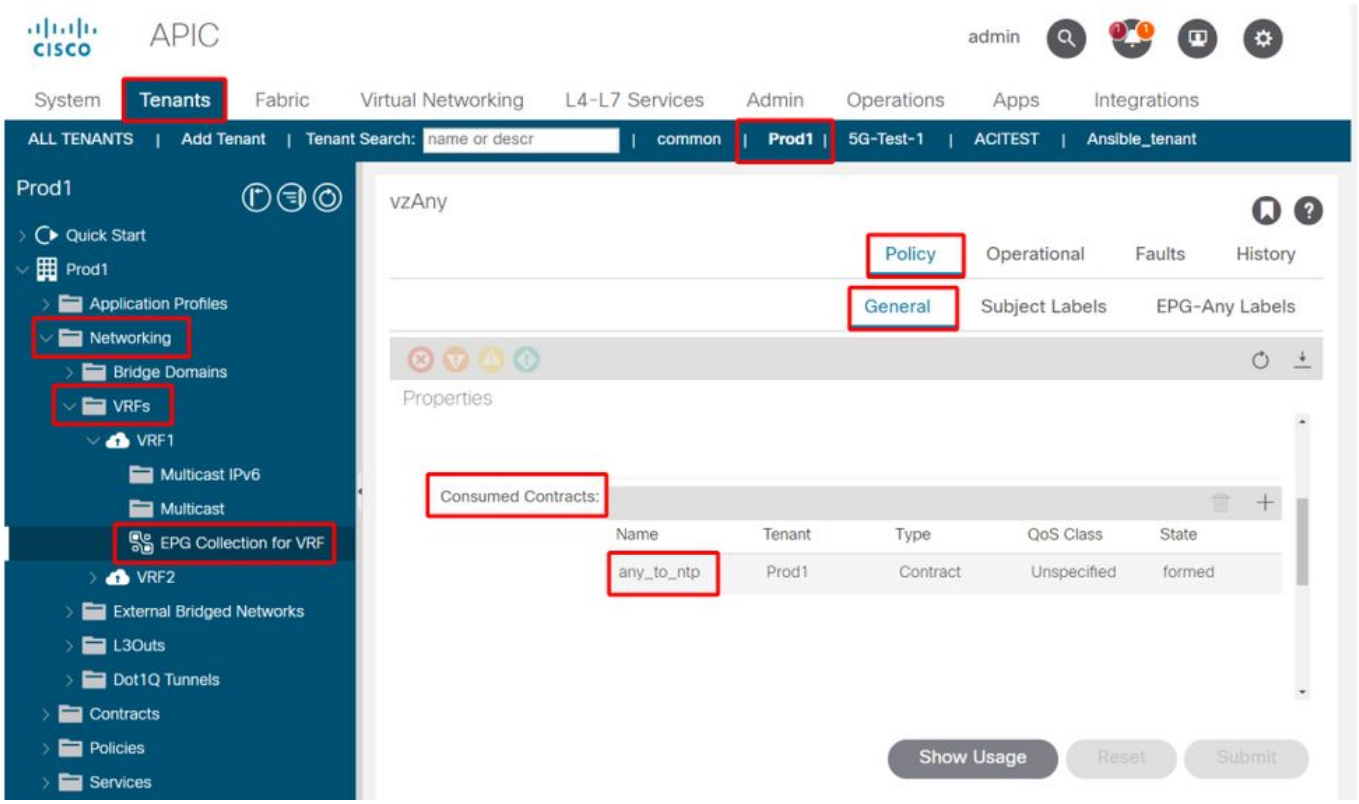
EPG NTP pcTag及其VRF VNID/范围



2.验证合同是否配置为vz作为VRF一部分的任何使用合同

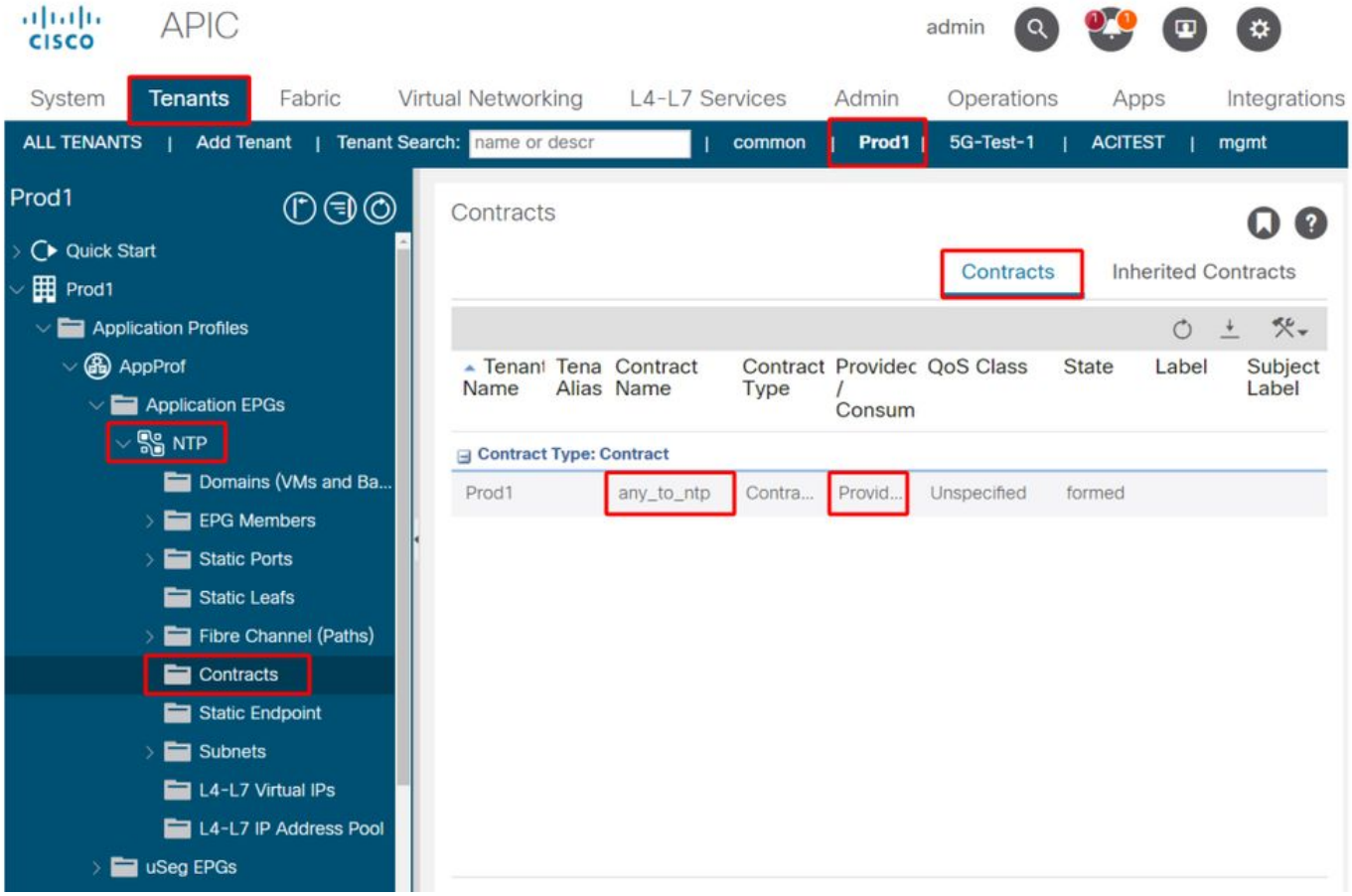
导航到VRF，并检查“EPG Collection for VRF”下是否有已使用的合同配置为vzAny。

配置为已消耗vz的合同VRF上的任何合同



3.验证相同合同是否应用为EPG NTP上提供的合同

为了建立合同关系，需要将同一合同应用为向其VRF中的其他EPG提供NTP服务的EPG NTP上提供的合同。



4.使用contract_parser.py或“show zoning-rule”对入口枝叶进行分区规则验证

入口枝叶应该具有2个分区规则，以允许任何EPG和EPG NTP之间的双向流量传输（如果合同主题设置为允许两个方向）。“任何EPG”在分区规则编程中表示为pcTag 0。

在指定VRF时在入口枝叶上使用contract_parser.py或“show zoning-rule”命令可确保对zoning-rule进行编程。

允许流量从VRF中的其他EPG传入/传出EPG NTP的分区规则

使用contract_parser.py和“show zoning-rule”检查是否存在基于vzAny的分区规则。

这里显然有两种规则：

1. 规则4156和规则4168允许Any对NTP，反之亦然。它们的优先级为13和14: 允许流量从任何EPG(pcTag 0)流到EPG NTP(pcTag 49161)的分区规则。允许流量从EPG NTP(pcTag 46161)流向任何其他EPG(pcTag 0)的分区规则。
2. 规则4165，它是优先级为21的any to any deny规则（默认）。

鉴于最低优先级具有优先级，VRF的所有EPG将有权访问NTP EPG。

```
fab3-leaf8# contract_parser.py --vrf Prod1:VRF
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]
```

```
[13:4156] [vrf:Prod1:VRF1] permit ip tcp tn-Prod1/ap-Services/epg-NTP(49161) eq 123 epg:any
[contract:uni/tn-Prod1/brc-any_to_ntp] [hit=0]
[14:4168] [vrf:Prod1:VRF1] permit ip tcp epg:any tn-Prod1/ap-Services/epg-NTP(49161) eq 123
[contract:uni/tn-Prod1/brc-any_to_ntp] [hit=0]
[16:4176] [vrf:Prod1:VRF1] permit any epg:any tn-Prod1/bd-App(16386) [contract:implicit] [hit=0]
[16:4174] [vrf:Prod1:VRF1] permit any epg:any tn-Prod1/bd-Services(32776) [contract:implicit]
[hit=0]
[16:4160] [vrf:Prod1:VRF1] permit arp epg:any epg:any [contract:implicit] [hit=0]
[21:4165] [vrf:Prod1:VRF1] deny,log any epg:any epg:any [contract:implicit] [hit=65]
[22:4164] [vrf:Prod1:VRF1] deny,log any epg:any pfx-0.0.0.0/0(15) [contract:implicit] [hit=0]

fab3-leaf8# show zoning-rule scope 2654209
+---------+--------+--------+----------+---------+---------+---------+------------+----------+--
--------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |   Dir   |  operSt |  Scope  |    Name    |  Action
|       Priority       |
+---------+--------+--------+----------+---------+---------+---------+------------+----------+--
--------------------+
|   4165  |   0    |   0    | implicit | uni-dir | enabled | 2654209 |            | deny,log |
any_any_any(21)    |
|   4160  |   0    |   0    | implarp  | uni-dir | enabled | 2654209 |            |  permit  |
any_any_filter(17)  |
|   4164  |   0    |   15   | implicit | uni-dir | enabled | 2654209 |            | deny,log |
any_vrf_any_deny(22) |
|   4176  |   0    | 16386  | implicit | uni-dir | enabled | 2654209 |            |  permit  |
any_dest_any(16)   |
|   4174  |   0    | 32776  | implicit | uni-dir | enabled | 2654209 |            |  permit  |
any_dest_any(16)   |
|   4168  |   0    | 49161  |   424    | uni-dir | enabled | 2654209 | any_to_ntp |  permit  |
any_dest_filter(14)  |
|   4156  | 49161  |   0    |   425    | uni-dir | enabled | 2654209 | any_to_ntp |  permit  |
src_any_filter(13)  |
+---------+--------+--------+----------+---------+---------+---------+------------+----------+--
--------------------+
```

共享L3Out到EPG

关于共享L3Out

共享第3层出站是一种配置，它允许在一个VRF中有一个L3Out提供一些服务（外部访问），并且一个或多个其他VRF使用此L3Out。有关共享L3Out的详细信息，请参阅“外部路由”一章。

执行共享L3Out时，建议让合同的提供商成为共享L3Out，EPG成为合同的消费者。此场景将在本节中说明。

建议不要执行相反操作，即L3Out使用EPG提供的服务。原因与可扩展性有关，因为对于共享服务，zoning-rules仅安装在消费者VRF上。消费和提供原则表示流量从何处开始。使用默认入口策略实施时，这意味着策略实施将应用于消费者端，更具体地应用于入口枝叶（非边界枝叶）。 入口枝叶要执行策略，需要目标的pcTag。在此场景中，目标是外部EPG pcTag。入口枝叶因此执行策略实施并将数据包转发到边界枝叶。边界枝叶在其交换矩阵链路上接收数据包，该交换矩阵链路执行路由查找(LPM)并将数据包转发到目的前缀的邻接关系上。

但是，边界枝叶在将流量发送到目标EP时不会执行任何策略实施，也不会在将返回流量流回源EP时执行任何策略实施。

因此，只有入口非BL枝叶的策略CAM安装了条目（在消费者VRF中），BL的策略CAM不会受到影响。

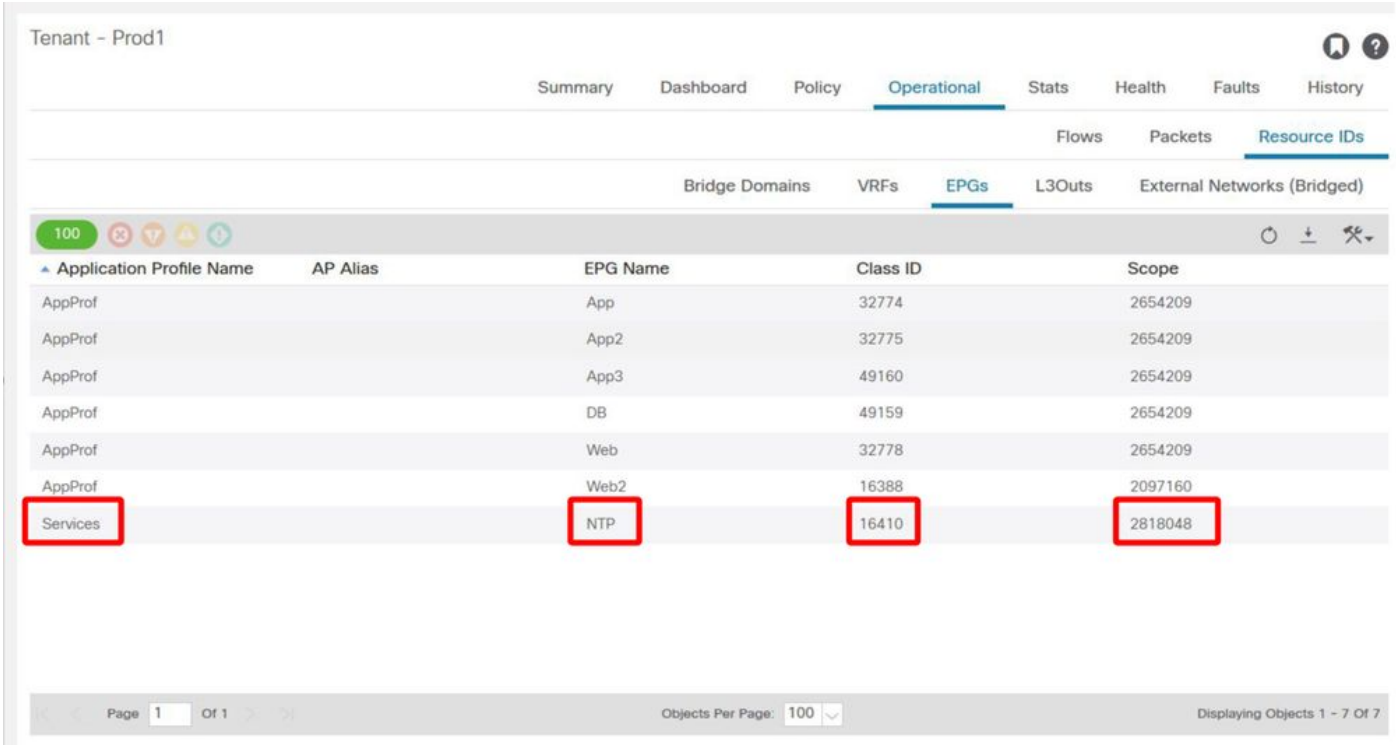
排除共享L3out故障

工作流程

1.验证消费者EPG的EPG pcTag和VRF VNID/范围

对于共享L3Out，分区规则仅安装在使用者VRF中。提供商必须拥有允许此pcTag用于所有消费者VRF的全局pcTag（低于16k）。在我们的场景中，提供商是外部EPG，将拥有全局pcTag。与往常一样，消费者EPG将使用本地pcTag。

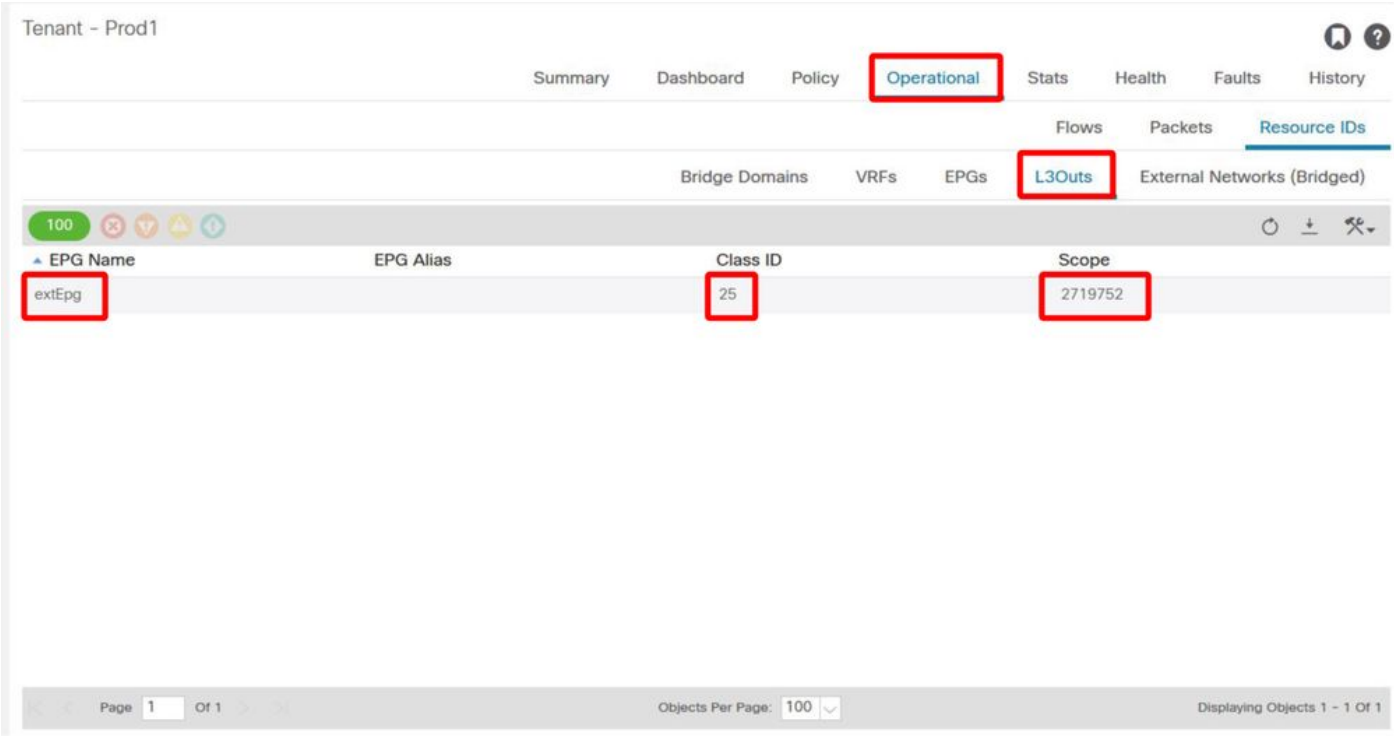
消费者EPG的pcTag



2.验证提供商L3Out EPG的pcTag和VRF VNID/范围

如步骤1所述，提供商L3Out EPG具有全局范围pcTag，作为L3Out的前缀，这些前缀会泄漏到消费者VRF中。因此，L3Out EPG pcTag需要与消费者VRF中的pcTag不重叠，因此它在全局pcTag范围内。

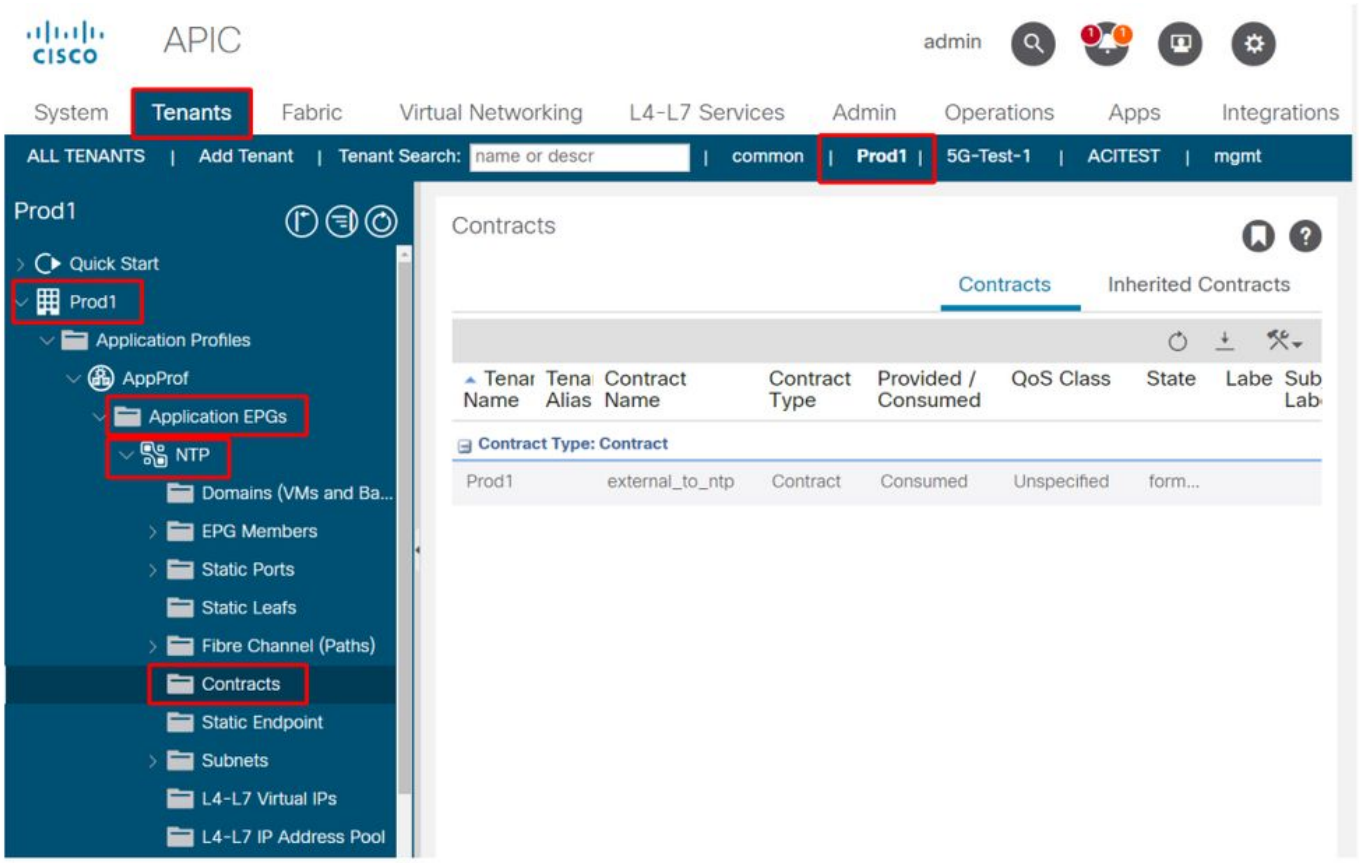
提供商外部EPG的pcTag



3.验证消费者EPG是否配置了导入的租户范围合同或全局合同

在EPG/BD下定义子网的消费者EPG NTP使用“租户”或“全局”范围合同

EPG使用的合同



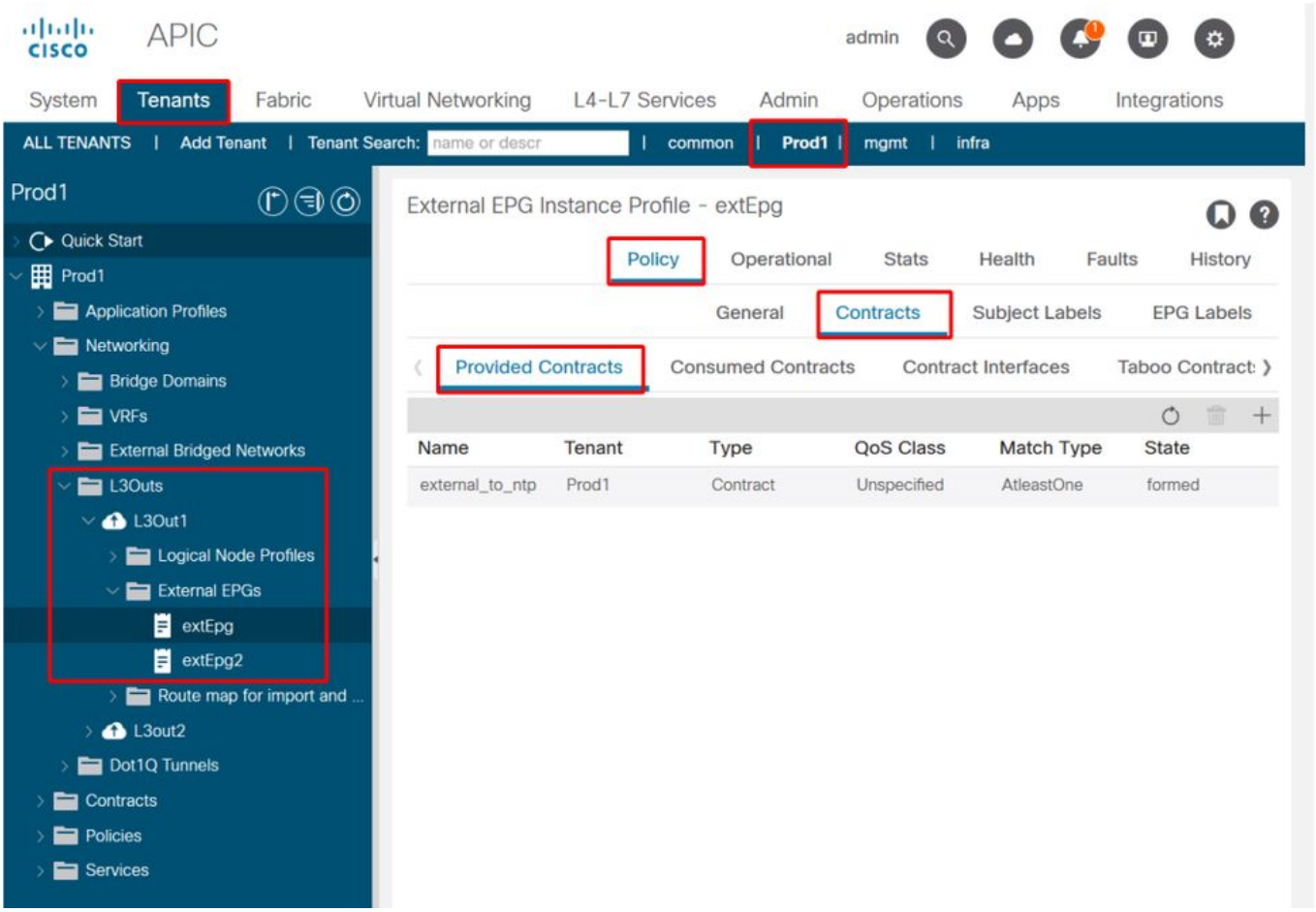
4.验证消费者EPG的BD是否配置了作用域设置为“在VRF之间共享”的子网

EPG的子网在桥接域下配置，但必须具有“在VRF之间共享”标志（允许路由泄漏）和“通告外部”标志（允许通告到L3Out）

5.验证提供商L3Out EPG是否配置了导入的租户范围合同或全局合同

L3Out EPG应具有租户范围合同或配置为所提供合同的全局合同。

提供商L3Out的合同



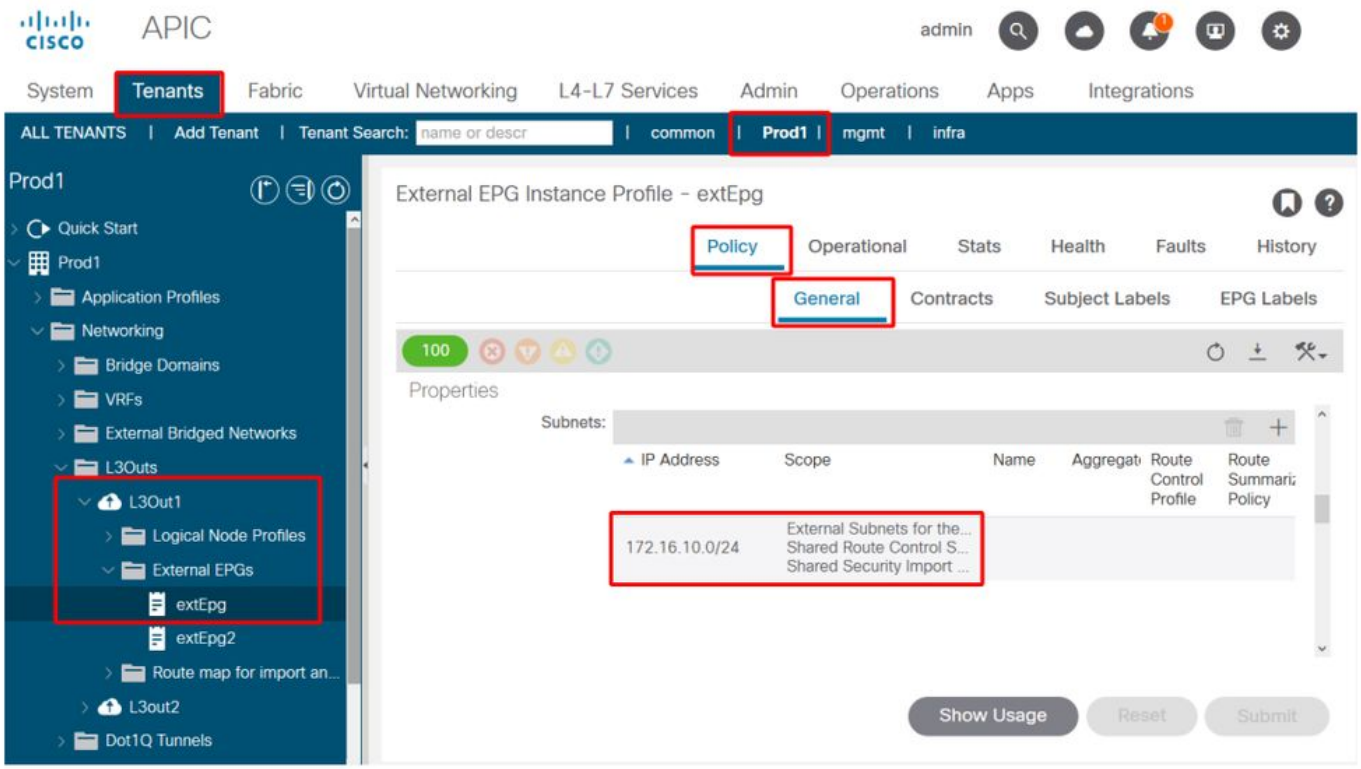
6.验证提供商L3Out EPG是否配置了已检查必要范围的子网

提供商L3Out EPG应使用以下范围配置要泄漏的前缀：

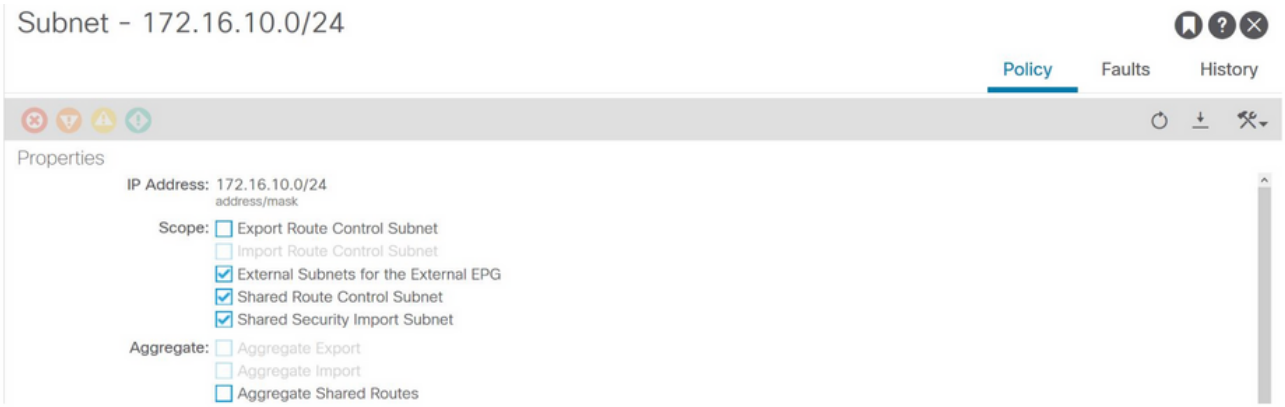
- 外部EPG的外部子网。
- 共享路由控制子网。
- 共享安全导入子网。

有关L3Out EPG中的子网标志的详细信息，请参阅“外部转发”一章。

外部EPG子网设置



外部EPG子网设置已展开



7.验证消费者VRF的非BL上L3Out EPG子网的pcTag

当发往外部EPG子网的流量进入非BL时，会根据目标前缀执行查找以确定pcTag。这可以在非BL上使用以下命令来检查。

请注意，此输出在VNI范围2818048即消费者VRF VNID。通过查看该表，消费者可以找到目标的pcTag，即使它不在同一VRF中。

```
fab3-leaf8# vsh -c 'show system internal policy-mgr prefix' | egrep 'Vrf-Vni|==|common:default'
Vrf-Vni VRF-Id Table-Id Table-State  VRF-Name
Addr                                Class Shared Remote Complete
======= ======  ========== =======  ============================
================================ ====== ====== ====== ========
2818048 19     0x13          Up     common:default
0.0.0.0/0   15      False  False  False
2818048 19     0x80000013    Up     common:default
::/0   15      False  False  False
2818048 19     0x13          Up     common:default
172.16.10.0/24  25      True   True   False
```

以上输出显示了L3Out EPG子网及其全局pcTag 25的组合。

8.验证消费者VRF的非BL上的编程分区规则

使用“contract_parser.py”或“show zoning-rule”命令并指定VRF。

以下命令输出显示了两个分区规则，以允许从使用者EPG本地pcTag 16410到L3Out EPG全局pcTag 25的流量。这属于范围2818048，即使用者VRF的范围。

```
fab3-leaf8# show zoning-rule scope 2818048
+---------+--------+--------+----------+----------------+---------+---------+----------------+-
---------+----------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope  |       Name     |
Action  |        Priority      |
+---------+--------+--------+----------+----------------+---------+---------+----------------+-
---------+----------------------+
|   4174  |   0    |   0    | implarp  |    uni-dir     | enabled | 2818048 |                |
permit  |   any_any_filter(17)  |
|   4168  |   0    |   15   | implicit |    uni-dir     | enabled | 2818048 |                |
deny,log |  any_vrf_any_deny(22) |
|   4167  |   0    | 32789  | implicit |    uni-dir     | enabled | 2818048 |                |
permit  |    any_dest_any(16)   |
|   4159  |   0    |   0    | implicit |    uni-dir     | enabled | 2818048 |                |
deny,log |    any_any_any(21)    |
|   4169  |   25   |   0    | implicit |    uni-dir     | enabled | 2818048 |                |
deny,log | shsrc_any_any_deny(12)|
|   4156  |   25   | 16410  |   425    | uni-dir-ignore | enabled | 2818048 | external_to_ntp |
permit  |     fully_qual(7)     |
|   4131  | 16410  |   25   |   424    |     bi-dir     | enabled | 2818048 | external_to_ntp |
permit  |     fully_qual(7)     |
+---------+--------+--------+----------+----------------+---------+---------+----------------+-
---------+----------------------+

fab3-leaf8# contract_parser.py  --vrf common:default
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]

[7:4131] [vrf:common:default] permit ip tcp tn-Prod1/ap-Services/epg-NTP(16410) tn-Prod1/l3out-
L3Out1/instP-extEpg(25) eq 123  [contract:uni/tn-Prod1/brc-external_to_ntp] [hit=0]
[7:4156] [vrf:common:default] permit ip tcp tn-Prod1/l3out-L3Out1/instP-extEpg(25) eq 123 tn-
Prod1/ap-Services/epg-NTP(16410)  [contract:uni/tn-Prod1/brc-external_to_ntp] [hit=0]
[12:4169] [vrf:common:default] deny,log any tn-Prod1/l3out-L3Out1/instP-extEpg(25) epg:any
[contract:implicit] [hit=0]
[16:4167] [vrf:common:default] permit any epg:any tn-Prod1/bd-Services(32789)
[contract:implicit] [hit=0]
[16:4174] [vrf:common:default] permit arp epg:any epg:any [contract:implicit] [hit=0]
[21:4159] [vrf:common:default] deny,log any epg:any epg:any [contract:implicit] [hit=0]
[22:4168] [vrf:common:default] deny,log any epg:any pfx-0.0.0.0/0(15) [contract:implicit]
[hit=0]
```

9.验证提供商VRF的BL上的编程分区规则

使用“contract_parser.py”或“show zoning-rule”命令并指定VRF。以下命令输出显示提供商VRF中没有之前多次概述的NO specific zoning-rules。

它处于提供商2719752的作用域范围内。

```
border-leaf# show zoning-rule scope 2719752
```

```
+---------+--------+--------+----------+----------------+---------+---------+-------------+----
------+----------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope  |     Name    |
Action  |       Priority       |
+---------+--------+--------+----------+----------------+---------+---------+-------------+----
------+----------------------+
|   4134  | 10937  |   24   | default  | uni-dir-ignore | enabled | 2719752 | vrf1_to_vrf2 |
permit  |    src_dst_any(9)    |
|   4135  |   24   | 10937  | default  |     bi-dir     | enabled | 2719752 | vrf1_to_vrf2 |
permit  |    src_dst_any(9)    |
|   4131  |   0    |   0    | implicit |    uni-dir     | enabled | 2719752 |              |
deny,log |   any_any_any(21)    |
|   4130  |   0    |   0    | implarp  |    uni-dir     | enabled | 2719752 |              |
permit  |  any_any_filter(17)  |
|   4132  |   0    |   15   | implicit |    uni-dir     | enabled | 2719752 |              |
deny,log | any_vrf_any_deny(22) |
+---------+--------+--------+----------+----------------+---------+---------+-------------+----
------+----------------------+

border-leaf# contract_parser.py --vrf Prod1:VRF3
Key:
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4]
[flags][contract:{str}] [hit=count]

[9:4134] [vrf:Prod1:VRF3] permit any tn-Prod1/l3out-L3Out1/instP-extEpg2(10937) tn-Prod1/l3out-
L3Out2/instP-extEpg2(24) [contract:uni/tn-Prod1/brc-vrf1_to_vrf2] [hit=0]
[9:4135] [vrf:Prod1:VRF3] permit any tn-Prod1/l3out-L3Out2/instP-extEpg2(24) tn-Prod1/l3out-
L3Out1/instP-extEpg2(10937) [contract:uni/tn-Prod1/brc-vrf1_to_vrf2] [hit=0]
[16:4130] [vrf:Prod1:VRF3] permit arp epg:any epg:any [contract:implicit] [hit=0]
[21:4131] [vrf:Prod1:VRF3] deny,log any epg:any epg:any [contract:implicit] [hit=0]
[22:4132] [vrf:Prod1:VRF3] deny,log any epg:any pfx-0.0.0.0/0(15) [contract:implicit] [hit=0]
```

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）。