排除ACI中意外路由泄漏故障

目录

概述

ACI通过部署简单策略来处理许多传统上复杂的路由和交换配置。这些功能包括泄漏vrf之间的路由以便于共享服务。传统上，这涉及许多步骤，例如定义路由目标、创建BGP地址系列、路由区别器，以及在许多设备上复制此配置。

在ACI中，路由泄漏通过合同和在子网上设置特定共享标志的组合进行处理。执行路由泄漏工作所需的所有传统配置都在后端通过合同和共享子网配置进行处理。

但是，抽象此配置后，确定哪个合同实际上导致路由泄露将变得更加困难。在具有大量epg、vrf和合同的环境中，尤其如此。如果路由在vrfs之间意外泄漏，管理员如何确定导致这种情况的配置（合同）？

本文档旨在演示如何确定哪种合同关系导致ACI中的路由在VRF之间泄露。它有助于您熟悉传统的路由泄漏概念，如路由目标和BGP VPNv4。

使用的软件

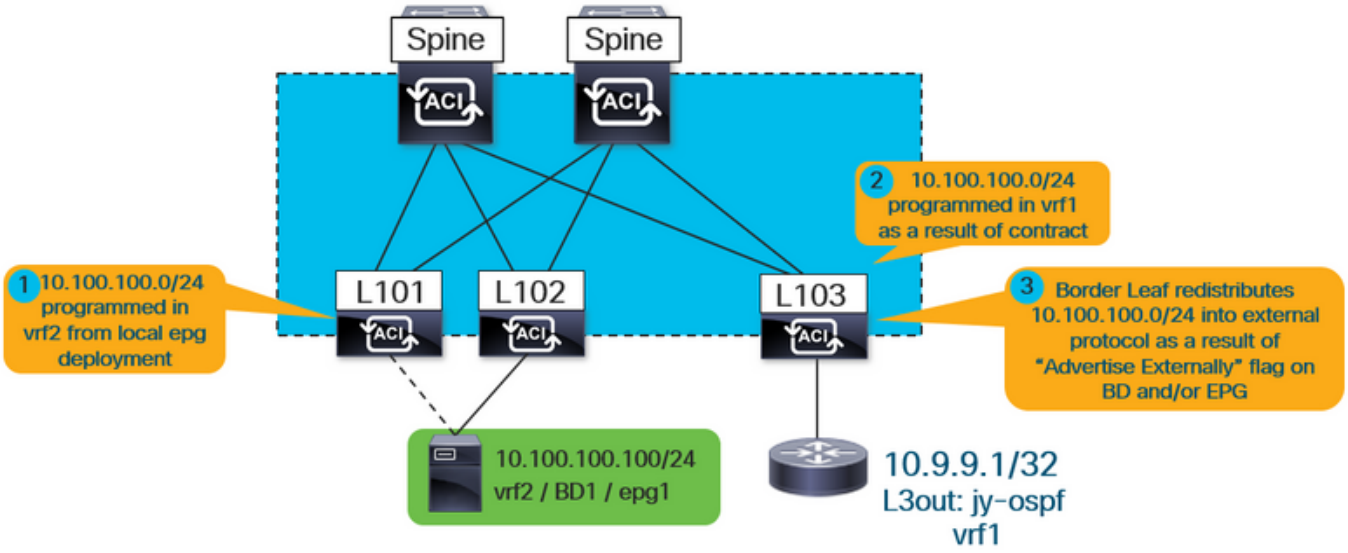
本文档中的所有示例均基于aci软件4.2(3j)。

为什么VRF x的网桥域/EPG子网安装在VRF y中？

本节将重点介绍BD或EPG子网意外泄漏到另一个vrf的场景。对于要泄漏的BD/EPG子网，必须配置“VRF之间共享”标志。更具挑战性的部分是了解导致此问题泄露的合同，以便本部分将解决此问题。

从高度讲，这是BD/EPG子网在vrfs之间泄漏时的工作流程。

图 1.



*请注意，#3仅在通告共享l3out时适用。#1和#2始终适用，无论是使用共享l3out还是共享服务完全为内部服务。

首先，用户如何知道安装的路由是否因BD或EPG子网而泄露？

当运行“**show ip route vrf <name>**”时，“**沉浸式**”标志表示路由是BD或EPG子网。

例如，在上述拓扑中，外部vrf(vrf1)的边界枝叶上会看到以下情况：

```
leaf103# show ip route 10.100.100.100 vrf jy:vrf1
IP Route Table for VRF "jy:vrf1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%
```

```
      pervasive *via 10.3.144.68%overlay-1, [1/0], 21:29:54, static, tag 4294967292 recursive
next hop: 10.3.144.68/32%overlay-1
```

此外，通过运行以下命令可以查看子网泄漏的目标vrf:

```
leaf103# vsh -c "show ip route 10.100.100.100 detail vrf jy:vrf1"
IP Route Table for VRF "jy:vrf1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%
```

```
      pervasive *via 10.3.144.68%overlay-1, [1/0], 21:34:16, static, tag 4294967292 recursive
next hop: 10.3.144.68/32%overlay-1


    vrf crossing information: VNID:0x258003 ClassId:0x18 Flush#:0x2
```

* (请注意，无论目的vrf与查找vrf是否不同，vrf交叉信息都会设置。)

在上述输出中，vrf交叉vnid设置为0x258003或十进制2457603。如何识别vnid 2457603所属的vrf?

从APIC只需查询fvCtx对象并基于segid进行过滤。

```
apic1# moquery -c fvCtx -f 'fv.Ctx.seg=="2457603"'
Total Objects shown: 1

# fv.Ctx
name                 : vrf2
dn                   : uni/tn-jy/ctx-vrf2
pcEnfDir             : ingress
pcEnfPref            : enforced
pcTag                : 49153
scope                : 2457603
seg                  : 2457603
```

如预期，路由是从vrf2 vrf获知的。

此时，它仍不清楚使用的合同以及提供的EPG和使用的EPG是什么，因此无法安装此路由。在提供商和消费者关系方面，需要牢记以下几点：
1.对于vrf间合同关系，合同（以及产生的分区规则）仅安装在消费者epg的vrf中。因此，提供程序vrf中的“show zoning-rule”将不显示关系。
2.即使合同仅安装在消费者vrf中，提供商vrf必须获得消费者vrf BD子网的路由，这意味着枝叶必须对合同有一些配置参考。

当路由意外泄露给消费者VRF时确定合同

枝叶上的ipCons对象安装在引用……

a.)泄露给消费者vrf的路由

b.)与关系方订立的合同

c.)提供商和消费者epg在关系中。

在下面的输出中，“jy:vrf1”是路由被泄漏到的消费者vrf，“10.100.100.0/24”是被泄漏的路由。

```
leaf103# moquery -c ipCons -f 'ip.Cons.dn*"jy:vrf1/rt-\[10.100.100.0/24\]"'
Total Objects shown: 1

# ip.Cons
consDn       : cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-
jy/brc-shared/dirass/cons-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-
```

```
shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]
subConsDn    :
childAction  :
dn           : sys/ipv4/inst/dom-jy:vrf1/rt-[10.100.100.0/24]/rsrouteToRouteDef-[bd-[uni/tn-
jy/BD-bd1]-isSvc-no/epgDn-[uni/tn-jy/ap-ap1/epg-epg1]/rt-[10.100.100.1/24]]/cons-[cdef-[uni/tn-
jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-jy/brc-shared/dirass/cons-
[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/ap-
ap1/epg-epg1]-any-no]]-sub-[]
lcOwn        : local
modTs        : 2019-12-23T12:50:51.440-05:00
name         :
nameAlias    :
rn           : cons-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-
[uni/tn-jy/brc-shared/dirass/cons-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-
shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]]-sub-[]
status       :
```

从上述输出中，合同名称为“共享”，使用者epg为l3out epg "uni/tn-jy/out-jy-ospf/instP-all"，提供商epg为“uni/tn-jy/ap-ap1/epg-epg1”。

当路由意外泄露给提供商VRF时确定合同

consNode对象安装在提供程序vrf的枝叶上。它引用了所泄露的消费者vrf中的BD子网、合同以及关系中的epg。在查询此对象之前，请查找配置了路由的BD子网。这可以通过查询apic上的fvSubnet对象来完成：

```
apic1:~> moquery -c fvSubnet -f 'fv.Subnet.dn*"10.100.100"'

# fv.Subnet
ip           : 10.100.100.1/24
dn           : uni/tn-jy/BD-bd1/subnet-[10.100.100.1/24]
preferred    : no
rn           : subnet-[10.100.100.1/24]
scope        : public,shared
```

路由在tn-jy/BD-bd1网桥域中配置。使用此命令和提供商vrf的vnid（路由被泄漏到）运行以下命令。

```
leaf103# moquery -c consNode -f 'cons.Node.dn*"2949122"' -f 'cons.Node.dn*"tn-jy/BD-bd1"'
Total Objects shown: 1

# cons.Node
consDn       : cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-
jy/brc-shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-
shared/dirass/cons-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]
annotation   :
childAction  :
descr        :
dn           : consroot-[bd-[uni/tn-jy/BD-bd1]-isSvc-no]-[sys/ctx-[vxlan-2949122]]/consnode-
[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-jy/brc-
shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-
shared/dirass/cons-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]]
extMngdBy    :
lcOwn        : local
modTs        : 2019-12-23T12:25:36.153-05:00
name         :
nameAlias    :
rn           : consnode-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-
all]/fr-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-
jy/brc-shared/dirass/cons-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]]
status       :
```

```
uid          : 0
```
从上述输出中，合同名称为“shared”，使用者epg为“uni/tn-jy/ap-ap1/epg-epg1”，提供商epg为l3out "tn-jy/out-jy-ospf/instP-all"。

确定路由意外被使用的vzAny合同泄露时的合同

vzAny示例从验证角度与传统提供商/消费者关系相同。以下示例将展示这种情况。请注意，vrf间合同仅作为消费者受vzAny支持。

vzAny示例1:路由意外泄漏到消费者VRF

与查看在使用者vrf中进行验证的第一个示例类似，将再次使用ipCons对象。

```
leaf103# moquery -c ipCons -f 'ip.Cons.dn*"jy:vrf1/rt-\[10.100.100.0/24\]"'
Total Objects shown: 1

# ip.Cons
consDn       : cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/ctx-vrf1/any]/fr-[uni/tn-jy/brc-shared/any-[uni/tn-jy/ctx-vrf1/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf1/any]-any-yes]/to-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]
subConsDn    :
childAction  :
dn           : sys/ipv4/inst/dom-jy:vrf1/rt-[10.100.100.0/24]/rsrouteToRouteDef-[bd-[uni/tn-jy/BD-bd1]-isSvc-no/epgDn-[uni/tn-jy/ap-ap1/epg-epg1]/rt-[10.100.100.1/24]]/cons-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/ctx-vrf1/any]/fr-[uni/tn-jy/brc-shared/any-[uni/tn-jy/ctx-vrf1/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf1/any]-any-yes]/to-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]]-sub-[]
lcOwn        : local
modTs        : 2019-12-23T13:11:08.077-05:00
name         :
nameAlias    :
rn           : cons-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/ctx-vrf1/any]/fr-[uni/tn-jy/brc-shared/any-[uni/tn-jy/ctx-vrf1/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf1/any]-any-yes]/to-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]]-sub-[]
status       :
```
从上述输出中，合同名称为“shared”，使用者epg为vrf1 vzAny "tn-jy/ctx-vrf1/any"，提供商epg为“uni/tn-jy/ap-ap1/epg-epg1”。

vzAny示例2:路由意外泄漏到提供商VRF

与查看在提供程序vrf中进行验证的位置的第二个示例类似，将再次使用consNode对象。请记住，要获取配置了泄漏子网的BD的bd名称，并获取其泄漏的vrf的vnid。

```
leaf103# moquery -c consNode -f 'cons.Node.dn*"vxlan-2949122"' -f 'cons.Node.dn*"tn-jy/BD-bd1"'
Total Objects shown: 1

# cons.Node
consDn       : cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-shared/any-[uni/tn-jy/ctx-vrf2/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf2/any]-any-yes]
annotation   :
childAction  :
descr        :
dn           : consroot-[bd-[uni/tn-jy/BD-bd1]-isSvc-no]-[sys/ctx-[vxlan-2949122]]/consnode-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-jy/brc-
```

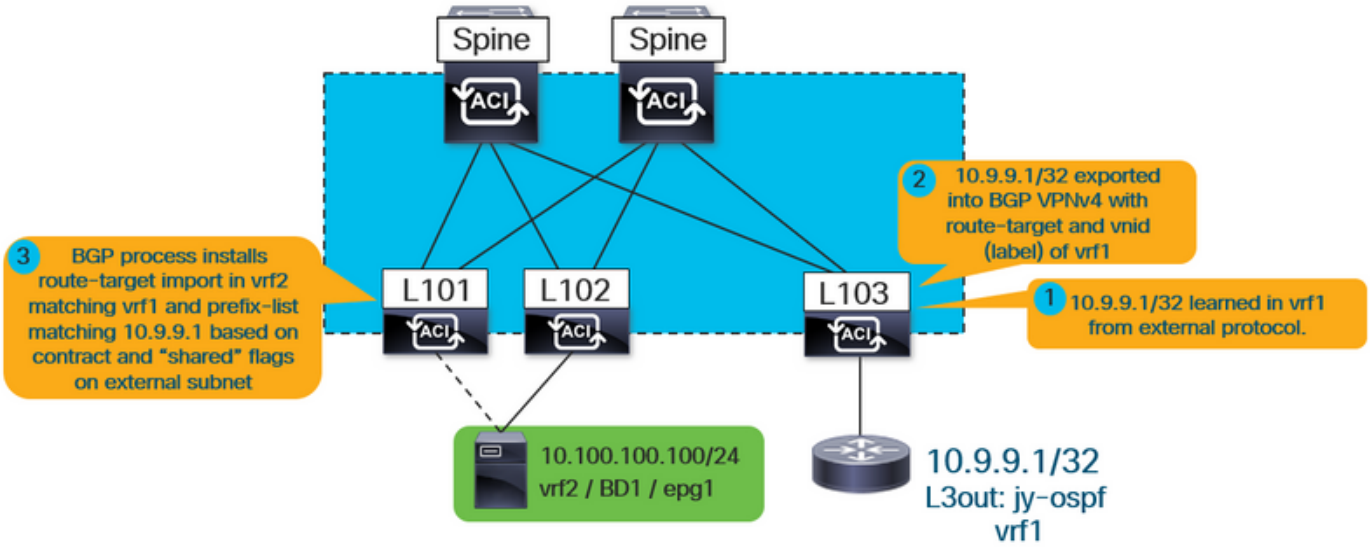
```
shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-shared/any-
[uni/tn-jy/ctx-vrf2/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf2/any]-any-yes]]
extMngdBy     :
lcOwn         : local
modTs         : 2019-12-23T13:06:09.016-05:00
name          :
nameAlias     :
rn            : consnode-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-
all]/fr-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-
jy/brc-shared/any-[uni/tn-jy/ctx-vrf2/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf2/any]-any-
yes]]
status        :
uid           : 0
```

从上述输出中，合同名称为“shared”，使用者epg为vrf2 vzAny "tn-jy/ctx-vrf2/any"，提供商epg为l3out "tn-jy/out-jy-ospf/instP-all"。

为什么VRF x中会安装来自VRF y的外部路由？

在高层，这是当l3out-learned（外部）路由在vrfs之间泄漏时发生的情况的工作流程。

图 2.



如上所示，内部vrf（本例中为vrf2）安装与vrf1匹配的路由目标导入。它还在bgp进程上安装导入映射，该映射应具有与l3out中定义的所有条目匹配的前缀列表条目，该l3out中选择了“共享路由控制子网”标志。

无论哪个epg是提供商或消费者，验证步骤都是相同的，因为合同始终负责导致路由目标导入和相应的前缀列表，这些前缀列表将泄漏路由以安装。

首先，验证路由实际上是通过l3out获知的：

```
leaf101# show ip route 10.9.9.1 vrf jy:vrf2
IP Route Table for VRF "jy:vrf2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%
```

```
       via 10.3.248.4%

    overlay-1, [200/5], 00:00:13,

    bgp-65001, internal, tag 65001
```

在上例中，从指向重叠中另一个枝叶的交换矩阵bgp进程获知的事实表明这来自l3out。

运行以下信息，以获取有关从哪个vrf获知的更多信息：

```
leaf101# vsh -c "show ip route 10.9.9.1 detail vrf jy:vrf2"
IP Route Table for VRF "jy:vrf2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%
```

```
       rw-vnid: 0x2d0002 table-id: 0x17 rw-mac: 0
```

如本文档前面所示，rewrite vnid 0x2d0002 / 2949122是目的vrf。在外部路由示例中，rw-vnid值设置为非零值表示这是从不同vrf获知的。在apic上运行moquery -c fvCtx -f 'fv.Ctx.seg="2949122"表示该属于vrf1。

接下来，查找路由目标导入以及与bgp进程关联的导入路由映射。

```
leaf101# show bgp process vrf jy:vrf2

Information regarding configured VRFs:

BGP Information for VRF jy:vrf2
VRF Type                        : System
VRF Id                          : 23
VRF state                       : UP
VRF configured                  : yes
VRF refcount                    : 0
VRF VNID                        : 2457603
Router-ID                       : 10.100.100.1
Configured Router-ID            : 0.0.0.0
Confed-ID                       : 0
Cluster-ID                      : 0.0.0.0
MSITE Cluster-ID                : 0.0.0.0
```

```
No. of configured peers        : 0
No. of pending config peers    : 0
No. of established peers       : 0
VRF RD                         : 101:2457603
VRF EVPN RD                    : 101:2457603

    Information for address family IPv4 Unicast in VRF jy:vrf2
    Table Id                   : 17
    Table state                : UP
    Table refcount             : 5
    Peers      Active-peers    Routes     Paths      Networks   Aggregates
    0          0               2          2          0          0

    Redistribution
        None

    Wait for IGP convergence is not configured
    Import route-map 2457603-shared-svc-leak <-- bgpRtCtrlMapP
    Export RT list:
        65001:2457603
    Import RT list:
        65001:2457603
        65001:2949122 <-- bgpRttEntry
    Label mode: per-prefix
```

上述内部vrf正在导出和导入其自己的路由目标(65001:2457603)。 它也在导入65001:2949122。 2949122 RT对应于它导入的vrf vnid(vrf1)。bgpRtCtrlMapP是包含前缀列表的导入路由映射的对象名称。bgpRttEntry是导入路由目标的对象名称。

接下来，使用学习外部vrf路由的内部vrf的vnid查询共享服务路由映射中安装的所有前缀列表。

```
leaf101# moquery -c rtpfxEntry -f 'rtpfx.Entry.dn*"pfxlist-IPv4'.*'2457603-shared-svc-leak"' |
egrep "criteria|dn|pfx|toPfxLen"
# rtpfx.Entry
criteria         : inexact
dn               : sys/rpm/pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak/ent-2
pfx              : 0.0.0.0/0
toPfxLen         : 32
# rtpfx.Entry
criteria         : exact
dn               : sys/rpm/pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak/ent-3
pfx              : 10.9.9.1/32
toPfxLen         : 0
# rtpfx.Entry
criteria         : exact
dn               : sys/rpm/pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak/ent-1
pfx              : 10.9.9.0/24
toPfxLen         : 0
```

每个条目应对应一个外部子网。“exact / inexact”属性指示是否在外部子网上设置了“聚合共享”标志。带有不精确标志的0.0.0.0/0前缀表示它将匹配所有更具体（有效的所有路由）的路由。 带有确切标志的10.9.9.0/24前缀表示它只与/24匹配。

查找与意外泄漏的路由匹配的条目（或条目）。在这种情况下，前缀为10.9.9.1/32，在上述输出中，ent-2和ent-3将匹配。

使用前缀列表名称，在路由映射中找到与其匹配的序列号。

```
leaf101# moquery -c rtmapRsRtDstAtt -f 'rtmap.RsRtDstAtt.tDn*"pfxlist-IPv4-2949122-24-25-
2457603-shared-svc-leak"'
```

```
Total Objects shown: 1

# rtmap.RsRtDstAtt
tDn          : sys/rpm/pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak
childAction  :
dn           : sys/rpm/rtmap-2457603-shared-svc-leak/ent-1001/mrtdst/rsrtDstAtt-
[sys/rpm/pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak]
forceResolve : yes
lcOwn        : local
modTs        : 2019-12-24T11:17:08.668-05:00
rType        : mo
rn           : rsrtDstAtt-[sys/rpm/pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak]
state        : formed
stateQual    : none
status       :
tCl          : rtpfxRule
tSKey        : IPv4-2949122-24-25-2457603-shared-svc-leak
tType        : mo
```

以上输出显示这是路由映射条目1001。此处的最后一部分是了解哪个合同负责在2457603-shared-svc-leak路由映射中创建路由映射条目1001。可以在fvAppEpGCons对象的枝叶上查询此项。

```
leaf101# moquery -c fvAppEpGCons -f 'fv.AppEpGCons.dn*"rtmap-2457603-shared-svc-leak/ent-1001"'
Total Objects shown: 1

# fv.AppEpGCons
consDn       : cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/ap-ap1/epg-epg1]/fr-[uni/tn-
jy/brc-shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]/to-[uni/tn-jy/brc-
shared/dirass/cons-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]
childAction  :
descr        :
dn           : uni/ctxrefcont/ctxref-[sys/ctx-[vxlan-2457603]]/epgref-[uni/tn-jy/ap-ap1/epg-
epg1]/epgpol-[sys/rpm/rtmap-2457603-shared-svc-leak/ent-1001]/epgcons-[cdef-[uni/tn-jy/brc-
shared]/epgCont-[uni/tn-jy/ap-ap1/epg-epg1]/fr-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/ap-
ap1/epg-epg1]-any-no]/to-[uni/tn-jy/brc-shared/dirass/cons-[uni/tn-jy/out-jy-ospf/instP-all]-
any-no]]
lcOwn        : local
modTs        : 2019-12-23T14:36:48.753-05:00
name         :
nameAlias    :
ownerKey     :
ownerTag     :
rn           : epgcons-[cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/ap-ap1/epg-epg1]/fr-
[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/ap-ap1/epg-epg1]-any-no]/to-[uni/tn-jy/brc-
shared/dirass/cons-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]]
status       :
```

以上输出显示合同名称为“共享”，提供商epg为“tn-jy/ap-ap1/epg-epg1”，消费者l3out epg为“tn-jy/out-jy-ospf/instP-all”

摘要

从BD/EPG子网泄漏的路由

如果泄漏的路由在“show ip route”中设置了“沉浸式”标志，则会从配置的BD/EPG子网中泄漏该路由。以下两个命令可用于检查导致此情况泄露的合同关系。它们将在意外安装路由的枝叶上运行。

如果路由意外泄露的vrf是消费者：
moquery -c ipCons -f 'ip.Cons.dn*"jy:vrf1/rt-\[10.100.100.0/24\]"' <—* ***jy:vrf1是路由泄漏到的vrf的名***

称，路由为10.100.100.0/24

如果路由意外泄露的vrf是提供商：
moquery -c consNode -f 'cons.Node.dn*"2949122"' -f 'cons.Node.dn*"tn-jy/BD-bd1"'
<—2949122是路由泄漏到的vrf的vnid，tn-jy/BD-bd1是配置子网的BD的名称（在路由泄漏的vrf内）。

从L3out泄露的路由

如果泄漏的路由是通过内部交换矩阵iBGP进程获知的，并且运行**vsh -c "show ip route x.x.x.x/y detail vrf <name>"显示非零rw-vnid值，则该路由是从另一个vrf中的l3out获知的。**无论哪个epg是消费者，哪个是提供商，验证都是相同的。

1.确*定内部vrf bgp进程上的共享服务导入路由映射：*
show bgp process vrf jy:vrf2 | grep "导入路由映射" ***<—jy:vrf2是路由泄漏到的内部vrf***

2.确定共享服务路由映射中与泄漏路由匹配的前缀列表：
moquery -c rtpfxEntry -f 'rtpfx.Entry.dn*"pfxlist-IPv4'.*'2457603-shared-svc-leak"' |egrep "criteria|dn|pfx|toPfxLen" ***<—2457603是本示例中内部vrf的vnid***

3.在找到引用路由的前缀列表后，确定引用该列表的路由映射序列号：
moquery -c rtmapRsRtDstAtt -f 'rtmap.RsRtDstAtt.tDn*"pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak"' ***<—pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak是前缀列表名称***

4.使用rtmap和条目编号运行以下命令以找出推送该路由映射条目的合同关系：
moquery -c fvAppEpGCons -f 'fv.AppEpGCons.dn*"rtmap-2457603-shared-svc-leak/ent-1001"'
<—rtmap-2457603-shared-svc-leak/ent-1001是步骤3中的路由映射名称和条目编号。