

# Troubleshooting de Vazamento de Rota Inesperado na ACI

## Contents

[Overview](#)

[Software usado](#)

[Por que uma sub-rede Bridge Domain / EPG de VRF x é instalada no VRF y?](#)

[Identificar o contrato quando a rota está vazando inesperadamente para o VRF do consumidor](#)

[Identificar o contrato quando a rota está vazando inesperadamente para o VRF do provedor](#)

[Identificar o contrato quando a rota está vazando inesperadamente por um contrato vzAny consumido](#)

[vzAny Exemplo 1: Rota vazada inesperadamente para VRF do consumidor](#)

[vzAny Exemplo 2: Rota vazada inesperadamente para VRF do provedor](#)

[Por que uma rota externa do VRF é instalada no VRF x?](#)

[Summary](#)

[Rota vazada da sub-rede BD / EPG](#)

[Rota vazada do L3out](#)

## Overview

A ACI lida com muitas configurações de roteamento e switching tradicionalmente complexas por meio da implantação de políticas simples. Entre essas funcionalidades está a capacidade de vazarem rotas entre vrfs para facilitar serviços compartilhados. Tradicionalmente, isso envolvia muitas etapas, como definir destinos de rota, criar famílias de endereços BGP, distinguidores de rotas e replicar essa configuração em vários dispositivos.

Dentro da ACI, o vazamento de rota é tratado através da combinação de contratos e da definição de flags compartilhados específicos nas sub-redes. Toda a configuração tradicional necessária para fazer com que o vazamento de rota funcione é tratada no backend como resultado do contrato e da configuração de sub-rede compartilhada.

No entanto, com essa configuração abstraída, pode ser mais difícil identificar qual contrato está realmente causando vazamento de uma rota. Isso é especialmente verdadeiro em ambientes com grande número de epgs, vrfs e contratos. Se uma rota está vazando inesperadamente entre vrfs, como um administrador pode identificar qual configuração (contrato) está fazendo com que isso aconteça?

A finalidade deste documento é demonstrar como identificar qual relação de contrato está fazendo com que uma rota na ACI vaze entre VRFs. É útil já estar familiarizado com os conceitos tradicionais de vazamento de rota, como roteadores-alvos e BGP VPNv4.

## Software usado

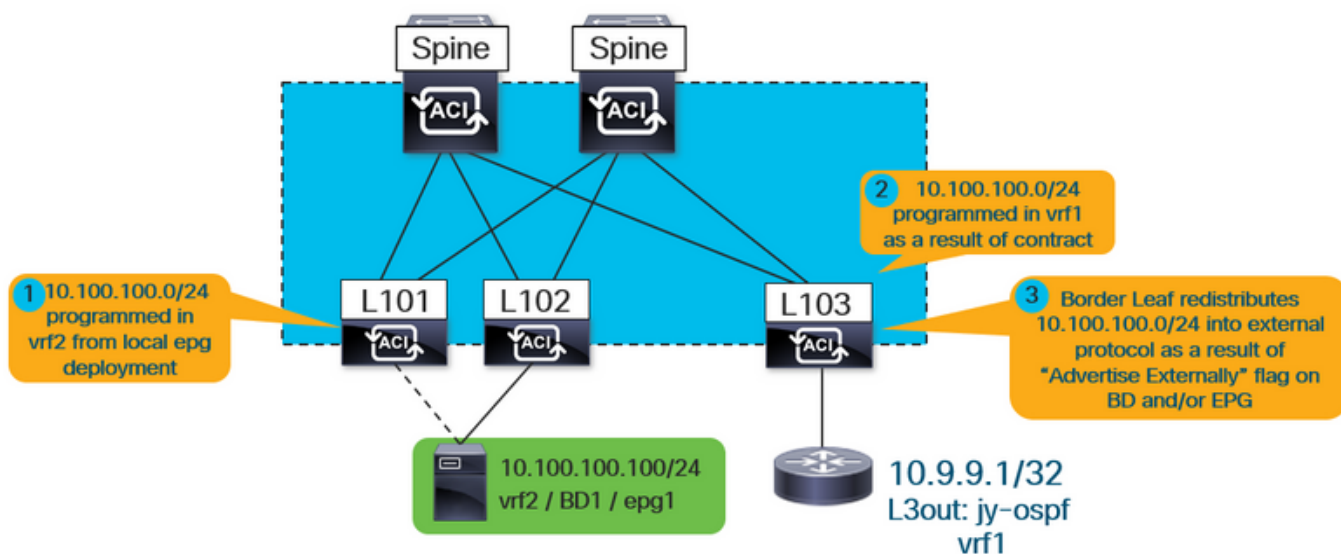
Todos os exemplos neste documento são baseados no software aci 4.2(3j).

# Por que uma sub-rede Bridge Domain / EPG de VRF x é instalada no VRF y?

Esta seção se concentrará no cenário em que uma sub-rede BD ou EPG está vazando inesperadamente para outro vrf. Para que uma sub-rede BD/EPG seja vazada, o sinalizador "Shared Between VRFs" deve ser configurado. A parte mais desafiadora é entender qual contrato está fazendo com que isso vaze, para que essa seção seja abordada.

Em um nível alto, esse é o fluxo de trabalho do que acontece quando uma sub-rede BD/EPG é vazada entre vrfs.

Figura 1.



\*Observe que o nº 3 só se aplica ao anúncio de uma L3out compartilhada. Os números 1 e 2 sempre se aplicam independentemente de se usar um L3out compartilhado ou se os serviços compartilhados forem totalmente internos.

Primeiro, como o usuário pode saber se a rota instalada vazou como resultado de uma sub-rede BD ou EPG?

Ao executar o "show ip route vrf <name>", o sinalizador "pervasivo" indica que a rota é uma sub-rede BD ou EPG.

Por exemplo, na topologia acima, isso seria visto na folha de borda no vrf externo (vrf1):

```
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
```

Além disso, o vrf de destino do qual a sub-rede vazou pode ser visualizado executando o seguinte comando:

```
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
```

\*(observe que as informações de travamento do vrf são definidas independentemente de o vrf de destino ser diferente do vrf de pesquisa).

Na saída acima, o vrf cruzando vnid é definido como 0x258003 ou decimal 2457603. Como o vrf que a vnid 2457603 pertence pode ser identificado?

A partir do APIC, basta consultar o objeto e o filtro fvCtx com base no segmento.

```
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
```

Como esperado, a rota está sendo aprendida do vrf2.

Ainda não se sabe qual contrato está sendo usado, bem como qual epg está fornecendo e qual epg está consumindo para que essa rota seja instalada. Há algumas considerações que devem ser lembradas em relação à relação entre o fornecedor e o consumidor:

1. Para uma relação de contrato entre vrf, o contrato (e a regra de zoneamento resultante) é instalado somente no vrf do epg consumidor. Como resultado, "show zoning-rule" no vrf do

provedor não mostrará a relação.

2. Embora o contrato seja instalado apenas no vrf do consumidor, o vrf do provedor deve obter a rota para a sub-rede vrf BD do consumidor, o que significa que o leaf deve ter alguma referência de configuração ao contrato.

## Identificar o contrato quando a rota está vazando inesperadamente para o VRF do consumidor

O objeto ipCons na folha é instalado na folha que faz referência a...

- a.) a rota que é objeto de fugas para o vrf consumidor
- b.) o contrato que estabelece a relação
- c.) o fornecedor e o cliente epg da relação.

Na saída abaixo, "jy.vrf1" é o vrf do consumidor ao qual a rota está sendo vazada e "10.100.100.0/24" é a rota que está sendo vazada.

```
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      :
```

Na saída acima, o nome do contrato é "compartilhado", o epg do consumidor é l3out epg "uni/tn-jy/out-jy-ospf/instP-all" e o epg do provedor é "uni/tn-jy/ap-ap1/epg-epg1".

## Identificar o contrato quando a rota está vazando inesperadamente para o VRF do provedor

O objeto consNode é instalado na folha no vrf do provedor. Ele faz referência à sub-rede BD no vrf do consumidor que está sendo vazado, o contrato e o epg está dentro do relacionamento. Antes de consultar esse objeto, localize a sub-rede BD onde a rota está configurada. Isso pode ser feito consultando o objeto fvSubnet no apic:

```
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
```

A rota é configurada no domínio de bridge tn-jy/BD-bd1. Use este comando e a vnid do vrf do provedor (para o qual a rota está vazando) para executar o comando abaixo.

```
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
```

Na saída acima, o nome do contrato é "compartilhado", a página do consumidor é "uni/tn-jy/ap-ap1/epg-epg1" e a página do provedor é l3out "tn-jy/out-jy-ospf/instP-all".

## Identificar o contrato quando a rota está vazando inesperadamente por um contrato vzAny consumido

O exemplo da vzAny será idêntico do ponto de vista da verificação a uma relação tradicional de provedor/consumidor. Os exemplos a seguir mostrarão como isso seria. Observe que um contrato entre vrf é suportado somente com o vzAny como consumidor.

### vzAny Exemplo 1: Rota vazada inesperadamente para VRF do consumidor

Da mesma forma que o primeiro exemplo observou onde a verificação foi feita no vrf do consumidor, o objeto ipCons será usado novamente.

```
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       :

```

Na saída acima, o nome do contrato é "compartilhado", o epg do consumidor é vrf1 vzAny "tn-jy/ctx-vrf1/any" e o epg do provedor é "uni/tn-jy/ap-ap1/epg-epg1".

## vzAny Exemplo 2: Rota vazada inesperadamente para VRF do provedor

Da mesma forma que o segundo exemplo observou onde a verificação foi feita no vrf do provedor, o objeto consNode será usado novamente. Lembre-se de obter o nome de bd do BD em que a sub-rede vazada está configurada e o vnid do vrf em que está vazada.

```

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

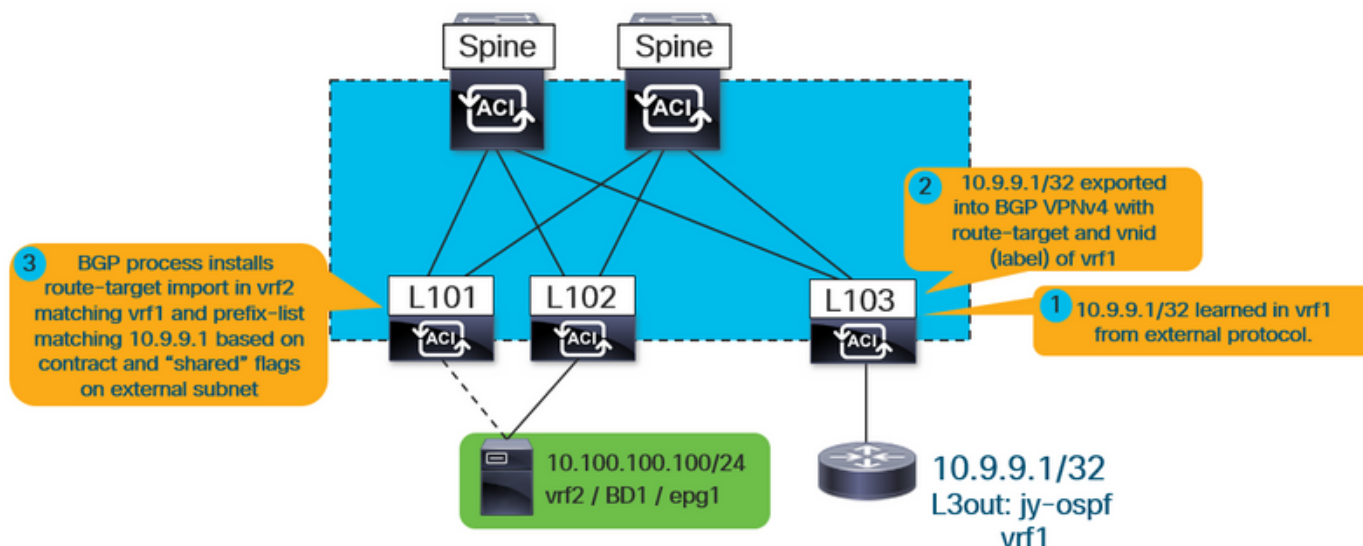
```

Na saída acima, o nome do contrato é "compartilhado", o epg do consumidor é o vrf2 vzAny "tn-jy/ctx-vrf2/any" e o epg do provedor é l3out "tn-jy/out-jy-ospf/instP-all".

## Por que uma rota externa do VRF é instalada no VRF x?

Em um nível alto, esse é o fluxo de trabalho do que acontece quando uma rota l3out-learning (externa) vaza entre vrfs.

Figura 2.



Como pode ser visto acima, o vrf interno (vrf2 neste caso) instala uma importação de destino de rota que corresponde a vrf1. Ele também instala um mapa de importação no processo bgp que deve ter entradas de lista de prefixos correspondentes a tudo definido na l3out que tem o sinalizador "sub-rede de controle de rota compartilhada" selecionado.

Independentemente de qual epg é o provedor ou consumidor, as etapas de verificação são as mesmas, pois o contrato sempre será responsável por causar a importação do destino da rota e as listas de prefixos correspondentes que vazarão as rotas para serem instaladas.

Em primeiro lugar, confirme se a rota está, de fato, sendo aprendida através de uma l3out:

```
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
```

No exemplo acima, o fato de que ele é aprendido do processo de bgp de estrutura apontando para outra folha na sobreposição indica que isso veio de uma l3out.

Execute as seguintes informações para obter mais informações sobre a qual vrf ela foi aprendida:

```
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
```

Como mostrado anteriormente neste documento, rewrite vnid 0x2d0002 / 2949122 é o vrf de destino. O valor rw-vnid sendo definido como um valor diferente de zero em um exemplo de rota externa indica que isso foi aprendido de um vrf diferente. A execução de `moquery -c fvCtx -f 'fv.Ctx.seg="2949122"'` no apic indicaria que isto pertence a vrf1.

Em seguida, localize as importações de destino de rota, bem como o mapa de rota de importação que está vinculado ao processo de bgp.

```
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

```

O vrf interno acima está exportando e importando seu próprio destino de rota (65001:2457603). Também está importando 65001:2949122. O 2949122 RT corresponde ao vrf vnid que está importando (vrf1). bgpRtCtrlMapP é o nome do objeto para o mapa de rota de importação que contém as listas de prefixo. bgpRttEntry é o nome do objeto para o destino da rota de importação.

Em seguida, usando a vnid do vrf interno que está aprendendo as rotas vrf externas, consulte todas as listas de prefixos instaladas no mapa de rota de serviços compartilhados.

```

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

```

Cada entrada deve corresponder a uma sub-rede externa. O atributo "exato / inexato" indica se o sinalizador "agregado compartilhado" foi definido na sub-rede externa. O prefixo 0.0.0.0/0 com o sinalizador inexato indica que ele corresponderia a todas as rotas mais específicas (efetivamente, tudo). O prefixo 10.9.9.0/24 com o sinalizador exato indica que ele só corresponderia a /24.

Encontre a entrada (ou entradas) que corresponde à rota que está vazando inesperadamente. Nesse caso, o prefixo é 10.9.9.1/32 seria correspondido por ent-2 e ent-3 nas saídas acima.

Usando o nome da lista de prefixos, localize o número de sequência no mapa de rotas que corresponde a ele.

```

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
```

A saída acima mostra que esta é a entrada 1001 do mapa de rota. A parte final aqui é entender qual contrato foi responsável pela criação da entrada 1001 do mapa de rotas no mapa de rotas 2457603-shared-svc-leak route-map. Isso pode ser consultado na folha do objeto fvAppEpGCons.

```
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]/epgp01-[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      :
```

A saída acima mostra que o nome do contrato é "compartilhado", o epg do provedor é "tn-jy/ap-ap1/epg-epg1" e o epg l3out do consumidor é "tn-jy/out-jy-ospf/instP-all"

## Summary

### Rota vazada da sub-rede BD / EPG

Se uma rota vazada tiver o sinalizador "pervasivo" definido em "show ip route", ela será vazada de uma sub-rede BD/EPG configurada. Os dois comandos a seguir podem ser usados para verificar qual relação de contrato está causando vazamento. Eles seriam executados no leaf onde a rota é inesperadamente instalada.

*Se o vrf onde a rota vazou inesperadamente for o consumidor:*

```
moquery -c ipCons -f 'ip.Cons.dn*"jy:vrf1/rt-[10.100.100.0/24]'" ←jy:vrf1 é o nome do vrf ao qual
a rota está vazando, a rota é 10.100.100.0/24
```

*Se o vrf onde a rota vazou inesperadamente é o provedor:*

```
moquery -c consNode -f 'cons.Node.dn*"2949122"' -f 'cons.Node.dn*"tn-jy/BD-bd1"' ←2949122 é
a vrid do vrf ao qual a rota está vazando, tn-jy/BD-bd1 é o nome do BD onde a sub-rede está
configurada (dentro do vrf do qual a rota está vazando).
```

## Rota vazada do L3out

Se a rota vazada for aprendida através do processo iBGP de estrutura interna e executar **vsh -c "show ip route x.x.x.x/y detail vrf <name>"** mostrar um valor **rw-vnid** diferente de zero, então a rota está sendo aprendida de um L3out em outro vrf. A validação é a mesma independentemente de qual epg é o consumidor e qual é o fornecedor.

1. *Identifique o mapa de rota de importação de serviços compartilhados no processo bgp interno do vrf:*

**show bgp process vrf jy:vrf2 | grep "Import route-map"** *←jy:vrf2 é o vrf interno ao qual a rota está vazando*

2. *Identifique a lista de prefixos que está dentro do mapa de rota de serviços compartilhados que corresponde à rota vazada:*

**moquery -c rtpfxEntry -f 'rtpfx.Entry.dn\*"pfxlist-IPv4'.\*2457603-shared-svc-leak"' | egrep "Criteria|dn|pfx|toPfxLen"** *←2457603 é a vnid do vrf interno neste exemplo*

3. *Depois de descobrir quais listas de prefixos fazem referência à rota, identifique quais números de sequência de mapa de rotas fazem referência à(s) lista(s):*

**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 é o nome da lista de prefixos*

4. *Usando o mapa de rota e o número de entrada, execute o seguinte comando para descobrir qual relação de contrato empurrou a entrada do mapa de rota:*

**moquery -c fvAppEpGCons -f 'fv.AppEpGCons.dn\*"rtmap-2457603-shared-svc-leak/ent-1001"'** *←rtmap-2457603-shared-svc-leak/ent-1001 é o nome do mapa de rota e o número de entrada da etapa 3.*