

# ACI のトラブルシューティング予想外ルート リーク

## 目次

### [概要](#)

#### [使用されるソフトウェア](#)

[ブリッジドメイン/EPG サブネットはなぜ VRF X からの VRF y にインストールされていますか。](#)

[ルートがコンシューマー VRF に予想に反して漏出されているとき契約を確認して下さい](#)

[ルートがプロバイダ VRF に予想に反して漏出されているとき契約を確認して下さい](#)

[ルートが消費された vzAny 契約によって予想に反して漏出されているとき契約を確認して下さい](#)

[vzAny Example 1:: 予想に反してコンシューマー VRF に漏出するルート](#)

[vzAny Example 2:: 予想に反してプロバイダ VRF に漏出するルート](#)

[外部ルートはなぜ VRF y からの VRF X にインストールされていますか。](#)

### [要約](#)

[BD/EPG サブネットから漏出するルート](#)

[L3out から漏出するルート](#)

## 概要

ACI は簡単なポリシーの配備によって多くの従来複雑なルーティング/スイッチング コンフィギュレーションを処理します。これらの機能性の中で共有サービスを促進するために vrf's 間のルーティングを漏出する機能があります。従来、これは BGP アドレス ファミリーを、ルート区分作成し、多くのデバイスを渡るこの設定を複製するルート ターゲットの定義のような多くのステップを必要としました。

ACI の中ではルート リークはサブネットの契約および設定仕様共用フラグの組み合わせによって処理されます。ルート リーク作業を作るために必要となる従来の設定すべてはおよび共用サブネット 設定の結果として契約の後部で処理されます。

ただし、抽出されてどの契約がルートを漏出するために実際に引き起こしているか確認するためにこの設定がそれはより挑戦的になることができます。これは多数の epg、vrf's および契約が付いている環境で言うことができます。どの設定 ( 契約 ) を起こすためにこれを引き起こしているか識別管理者がどのようにできるかルートが vrf's の間で予想に反して漏出されればか。

この資料の目的はどの契約関係が ACI でルートを VRF の間で漏出しますか識別する方法を示すことです。既にルート ターゲットおよび BGP VPNv4 のような従来のルート リーク概念について詳しく知っているその有用。

## 使用されるソフトウェア

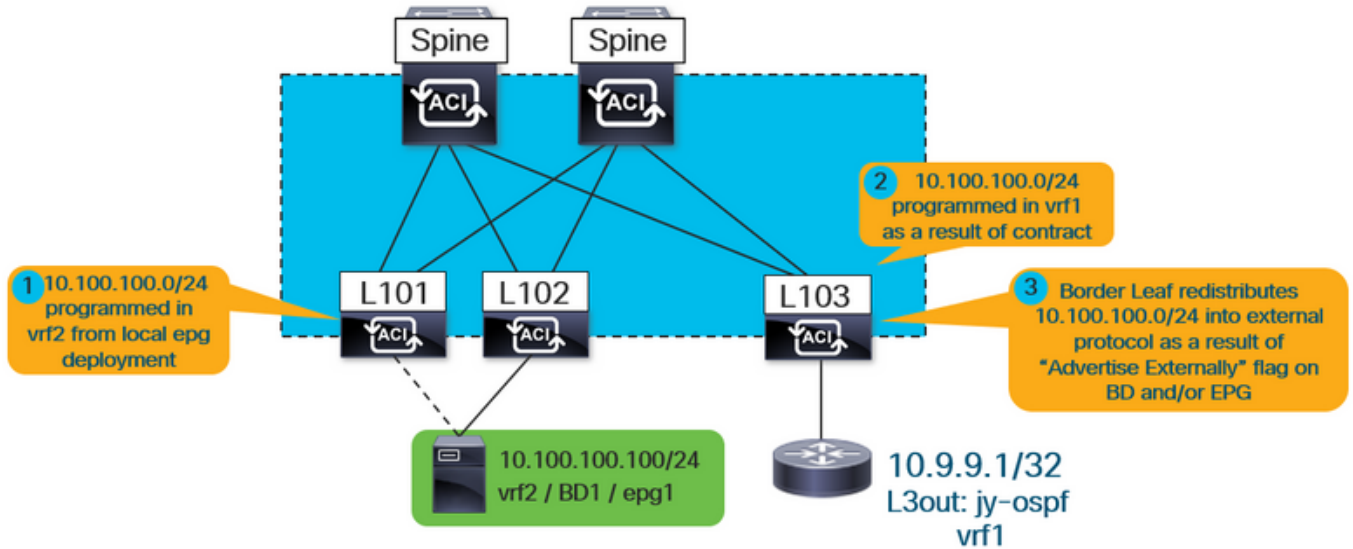
この資料のすべての例は aci ソフトウェア 4.2(3j) に基づいています。

## ブリッジドメイン/EPG サブネットが VRF y に VRF X からのインストールされている理由

このセクションは BD または EPG サブネットが別の VRF に予想に反してリークさせているシナリオに焦点を合わせます。「VRF」の間で共有されてフラグ リークするべき BD/EPG サブネットに関しては設定されなければなりません。より挑戦的な一部はこのセクションによってが当たるものであるようにどの契約がこれをリークさせますか理解しています。

高レベルでこれは BD/EPG サブネットが vrf の間でリークするとき何が起こるかのための作業の流れです。

図 1.



#3 が適当場合のだけアドバタイジング共用 I3out であること \*Note。 共用 I3out が使用されるか、または共有サービスが完全に内部なら場合 #1 および #2 はに関係なく常に適用します。

まず、ユーザはどのようにインストール済みルートが BD または EPG サブネットの結果として漏出するかどうか確認する場合がありますか。

「show ip route vrf <name>」を実行するとき「普及する」フラグはルートが BD または EPG サブネットであることを示します。

たとえば、上記のトポロジーでこれは外部 VRF ( vrf1 ) のボーダー リーフで見られます:

```
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]
'%<string>' in via output denotes VRF <string>

10.100.100.0/24, ubest/mbest: 1/0, attached, direct, 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
```

サブネットがリークしたことさらに宛先 VRF は次のコマンドの実行によって表示することができますから:

```
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]  
'%<string>' in via output denotes VRF <string>

```
10.100.100.0/24, ubest/mbest: 1/0, attached, direct, 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
```

\*宛先 VRF がルックアップ VRF と異なっている場合 ) ( VRF 交差情報に設定 されずに関係なく注意して下さい。

上では VRF 交差 vnid を設定 されます 0x258003、か小数点 2457603 に出力して下さい。VRF はどのように vnid 2457603 が識別されるために属することができるか。

APIC クエリから fvCtx は segid に基づいて単に反対し、フィルタリングします。

```
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
```

予想通り、ルートは vrf2 VRF から学習されています。

契約が消費して、また epg が使用されている epg がインストールされますこのルートに提供しているこの時点でそのまだ未知数。プロバイダおよびコンシューマー関係に関して覚えるべき幾つかの考慮事項があります:

1. 相互 VRF 契約関係に関しては契約 ( および生じるゾーン分割ルール コンシューマー epg の VRF にだけ ) インストールされています。その結果プロバイダ VRF のゾーン分割ルール」が関係を示さないことを「示して下さい。
2. 契約がコンシューマー VRF にインストールされているのに、プロバイダ VRF はコンシューマー VRF BD サブネットのためのルートを得る必要がありますつまりリーフは契約への設定参照がなければならないことだけを意味します。

**ルートがコンシューマー VRF に予想に反して漏出されているとき契約を確認して下さい**

ipCons は参照...リーフでリーフでインストールされています反対します

- a.) コンシューマー VRF にリークするルート
- b.) 関係を確立する契約
- c.) 関係のプロバイダおよびコンシューマー epg。

下記の出力で "jy.vrf1" はルートがに漏出され、リークさせている "10.100.100.0/24" がルートであることコンシューマー VRF です。

```
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      :
```

上からコントラクト名を「共有されます」出力して下さい、コンシューマー epg は「uni/tn-jy/out-jy-ospf/instP」すべての I3out epg 「であり、プロバイダ epg は "uni/tn-jy/ap-ap1/epg-epg1" です。

## ルートがプロバイダ VRF に予想に反して漏出されているとき契約を確認して下さい

consNode オブジェクトはプロバイダ VRF にリーフでインストールされています。それはリークさせているコンシューマー VRF、関係内の契約および epg で BD サブネットを参照します。このオブジェクトを照会する前にルートが設定される BD サブネットを見つけて下さい。これは apic の fvSubnet オブジェクトの照会によって実行することができます:

```
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
```

ルートは tn-jy/BD-bd1 ブリッジドメインで設定されます。下記のコマンドを実行するのに ( ルートが漏出されている ) のこれおよび vnid をプロバイダ VRF 使用して下さい。

```
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-[vxlان-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

```

上からコントラクト名を「共有されます」出力して下さい、コンシューマー epg は "uni/tn-jy/ap-ap1/epg-epg1" であり、プロバイダ epg は I3out 「TNjy/」 jy ospf/instP すべてです。

## ルートが消費された vzAny 契約によって予想に反して漏出されているとき契約を確認して下さい

vzAny 例は確認観点から従来のプロバイダ/コンシューマー関係と同一であることを行っています。下記の例はちょうどこれが見えるものに示します。相互 VRF 契約がコンシューマーとして vzAny とだけサポートされることに注目して下さい。

### vzAny Example 1:: 予想に反してコンシューマー VRF に漏出するルート

検知された最初の例に類似した 確認がコンシューマー VRF でどこに行われたか、ipCons オブジェクトは再度使用されます。

```

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        :

```

上からコントラクト名を「共有されます」出力して下さい、コンシューマー epg は vrf1 vzAny "tn-jy/ctx-vrf1/any" であり、プロバイダ epg は "uni/tn-jy/ap-ap1/epg-epg1" です。

### vzAny Example 2:: 予想に反してプロバイダ VRF に漏出するルート

検知された第 2 例に類似した 確認がプロバイダ VRF でどこに行われたか、consNode オブジェクトは再度使用されます。リークさせたサブネットが設定されるおよびにリークする VRF の

vnid を忘れないようにして下さい BD の bd 名前得ることを。

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

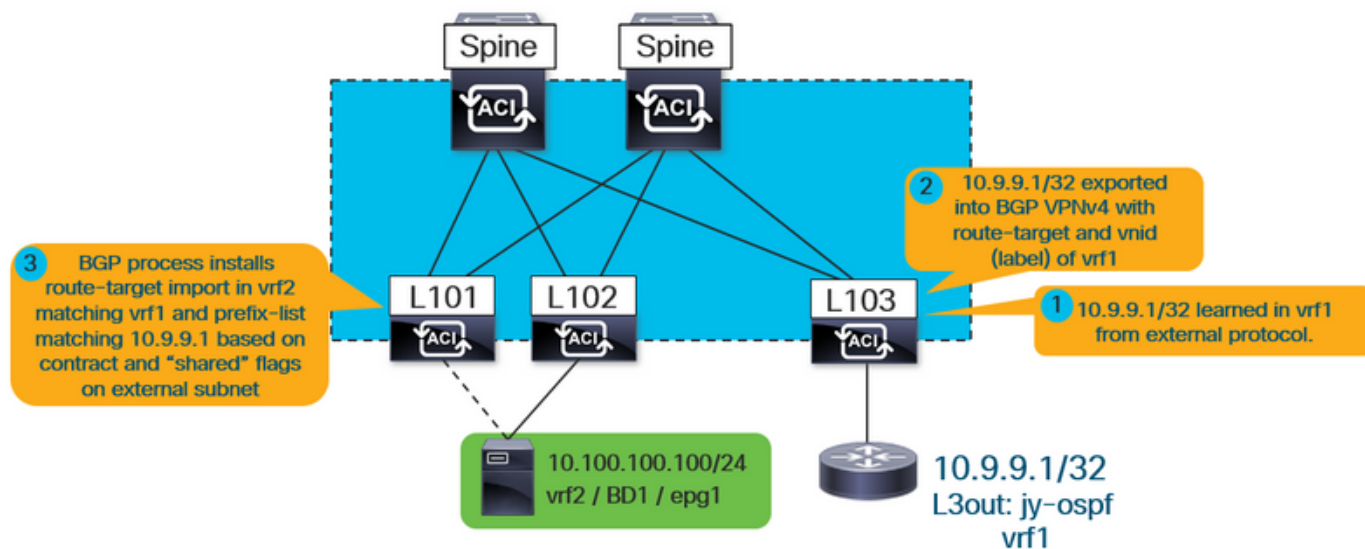
```
# cons.Node
consDn      : cdef-[uni/tn-jy/brc-shared]/epgCont-[uni/tn-jy/out-jy-ospf/instP-all]/fr-[uni/tn-jy/brc-shared/dirass/prov-[uni/tn-jy/out-jy-ospf/instP-all]-any-no]/to-[uni/tn-jy/brc-shared/any-[uni/tn-jy/ctx-vrf2/any]-type-cons_as_any/cons-[uni/tn-jy/ctx-vrf2/any]-any-yes]
annotation  :
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
```

上からコントラクト名を「共有されます」出力して下さい、コンシューマー epg は vrf2 vzAny "tn-jy/ctx-vrf2/any" であり、プロバイダ epg は l3out 「TNjy/」 jy ospf/instP すべてです。

## 外部ルートが VRF X に VRF y からのインストールされている理由

高レベルでこれは l3out-learned (外部) ルートが vrf の間で漏出するとき何が起こるかのための作業の流れです。

図 2 :



上で見られる場合があるように内部 VRF ( このケースの vrf2 ) は vrf1 と一致するルート・ターゲット インポートをインストールします。それはまた選択される「共用ルート制御サブネット」フラグがある I3out に定義されるすべてと一致するリスト エントリがあるはずである BGPプロセスでインポート マップをインストールします。

プロバイダまたはコンシューマーは契約がインストールされるためにルーティングを漏出する対応したプレフィクスリストおよびルート・ターゲット インポートを引き起こす役割がある常いのでどのに関係なく epg であるか、検証手順は同じです。

まずルートが実際 I3out によって学習されていること、検証して下さい:

```
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]
'%<string>' in via output denotes VRF <string>
```

```
10.9.9.1/32, ubest/mbest: 1/0
```

```
*via 10.3.248.4%overlay-1, [200/5], 00:00:13, bgp-65001, internal, tag 65001
```

上の例では、オーバーレイの別のリーフを指すファブリック BGPプロセスから学習されるというファクトはこれが I3out から来たことを示します。

どの VRF 得るために次の情報をから学習されたかについてのより多くのヒントを実行して下さい:

```
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]
'%<string>' in via output denotes VRF <string>
```

```
10.9.9.1/32, ubest/mbest: 1/0
```

```
*via 10.3.248.4%overlay-1, [200/5], 00:05:46, bgp-65001, internal, tag 65001 (mpls-vpn)
```

```
MPLS[0]: Label=0 E=0 TTL=0 S=0 (VPN)
```

```
client-specific data: 6b
```

```
recursive next hop: 10.3.248.4/32%overlay-1
```

```
extended route information: BGP origin AS 65001 BGP peer AS 65001 rw-vnid: 0x2d0002
```

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

この資料で示されていた ealier ように、書き直し vnid 0x2d0002/2949122 は宛先 VRF です。rwnid はこれが別の VRF から学習されたことを示します外部ルート例の非ゼロ値に設定されることを評価します。moquery を実行して -c fvCtx -これが vrf1 に属することを apic の f 'fv.Ctx.seg=="2949122" は示します。

次に、ルート・ターゲット 輸入高を見つけて下さい、またインポート ルート ルート・マップ BGPプロセスに結ばれる。

```
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
```

上記の内部 VRF はで自身のルート・ターゲットをエクスポートし、インポートします ( 65001:2457603 )。それはまた 65001:2949122 をインポートしています。2949122 RT はそれがインポートしている VRF vnid に対応します ( vrf1 )。bgpRtCtrlMapP はプレフィクスリストが含まれているインポート ルート ルート・マップのオブジェクト名です。bgpRttEntry ルート・ターゲット インポートのオブジェクト名はです。

次に、外部 VRF ルーティングを学習している内部 VRF の vnid を使用して、共有サービス ルート マップの中でインストールされているプレフィクスリストすべてを問い合わせして下さい。

```

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

```

各エントリは外部サブネットに対応する必要があります。「正確で/不正確な」アトリビュートは



「集約が外部サブネットで」フラグを設定された共有したかどうか示します。不正確なフラグの 0.0.0.0/0 プレフィクスはより多くの仕様 ( 効果的にすべて ) であるすべてのルーティングを一致することを示します。正確なフラグの 10.9.9.0/24 プレフィクスは示しますその /24 だけを一致することを。

予想に反してリークさせているエントリ) その一致しますルートとを ( 検出して下さい。この場合プレフィクスは上記の出力の ent-2 および ent-3 によって 10.9.9.1/32 一致しますです。

プレフィクスリスト名前を使用する、それと一致するルート マップ内のシーケンス番号を見つけて下さい。

```
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
```

上記の出力はこれがルート マップ エントリ 1001 であることを示したものです。ここの最終的な部分はこの契約が 2457603 共有 SVC リーク ルート マップ内のルート マップ エントリ 1001 を作成する役割があったか理解することです。これは fvAppEpGCons オブジェクトからのリーフで問い合わせることができます。

```
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 :
```

上記の出力はコントラクト名が「共有される」ことを、epg によってが "tn-jy/ap-ap1/epg-epg1" であるコンシューマー l3out epg は「TNjy/」jy ospf/instP すべてですプロバイダ示したものです、

## 要約

### BD/EPG サブネットから漏出するルート

漏出されたルートに「show ip route」で「普及する」フラグが設定があれば設定される BD/EPG サブネットからリークします。次の2つのコマンドがどの契約関係がこれをリークさせますかチェックするのに使用することができます。それらはルートが予想に反してインストールされているリーフで動作します。

ルートが予想に反して漏出する VRF がコンシューマーなら:

```
moquery は- c ipCons - f 'ip.Cons.dn*"jy:vrf1/rt-[10.100.100.0/24]" <--jy:vrf1 ルートがに漏出する VRF の名前、ルートです 10.100.100.0/24 です
```

ルートが予想に反して漏出する VRF がプロバイダなら:

```
moquery は- c consNode - f 'cons.Node.dn*"2949122" - f 'cons.Node.dn*"tn-jy/BD-bd1" <-- 2949122 ルートがに漏出する VRF の vnid、tn-jy/BD-bd1 ですサブネットが設定される BD の名前です ( VRF の内でルートはから漏出します )。
```

### L3out から漏出するルート

漏出されたルートが内部 ファブリック iBGP プロセスによって学習されればおよび実行 vsh - c 「ゼロ以外の rwnvid 値がそれからルート別の VRF の l3out から学習されていることを show ip route x.x.x.x/y 詳細 VRF <name>」は示します。検証は epg がコンシューマーのプロバイダがあり、同じです。

1. 内部 VRF BGPプロセスの共有サービス インポート ルート ルート・マップを識別して下さい: BGPプロセス VRF jy:vrf2 を表示して下さい | ルートがに漏出することグレップ「インポート ルート ルート・マップ」 <--jy:vrf2 は内部 VRF です

2. 漏出されたルートと一致する共有サービス ルート マップの内にあるプレフィクスリストを識別して下さい:

```
moquery - rtpfxEntry c - f 'rtpfx.Entry.dn*"pfxlist-IPv4'."*2457603-shared-svc-leak" | egrep 「基準 [dn|pfx|]」 <--2457603 をですこの例の内部 VRF の vnid toPfxLen
```

3. どのプレフィクスリスト参照ルートが、確認するか見つけた後どのルート マップシーケンス番号がリストを参照するか:

```
moquery は- c rtmapRsRtDstAtt - f 'rtmap.RsRtDstAtt.tDn*"pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak" <--pfxlist-IPv4-2949122-24-25-2457603-shared-svc-leak プレフィクスリスト名前です
```

4. rtmap および Entry Number を使用する調べるためにそれを押される関係をルート マップ エントリ引き締める次のコマンドを実行して下さい:

```
moquery は- c fvAppEpGCons - f 'fv.AppEpGCons.dn*"rtmap-2457603-shared-svc-leak/ent-1001" <--rtmap-2457603-shared-svc-leak/ent-1001 ステップ 3.からのマップ ネームおよび Entry Number です。
```