ACIにおける予期しないルート漏出のトラブルシューティング

内容

概要

ACIは、シンプルなポリシーを導入することで、従来の複雑なルーティングおよびスイッチングの設定を数多く処理します。これらの機能の中には、共有サービスを促進するためにvrf間でルートをリークする機能があります。従来、これはルートターゲットの定義、BGPアドレスファミリの作成、ルート識別子、および多数のデバイス間でこの設定を複製するなど、多くの手順を伴っていました。

ACI内のルート漏出は、コントラクトの組み合わせとサブネット上の特定の共有フラグの設定によって処理されます。ルート漏出を行うために必要な従来の設定はすべて、契約と共有サブネットの設定の結果としてバックエンドで処理されます。

ただし、この設定が抽象化されると、どの契約が実際にルートのリークを引き起こしているのかを特定することが難しくなります。これは、epg、vrf、および契約が多数の環境で特に当てはまります。vrf間でルートが予期せず漏出している場合、この問題の原因となっている設定（コントラクト）を管理者が特定するにはどうすればよいのですか。

このドキュメントの目的は、ACIのルートがVRF間で漏出する原因となっている契約関係を特定する方法を示すことです。ルートターゲットやBGP VPNv4などの従来のルート漏出の概念を理解しておくのに役立ちます。

使用するソフトウェア

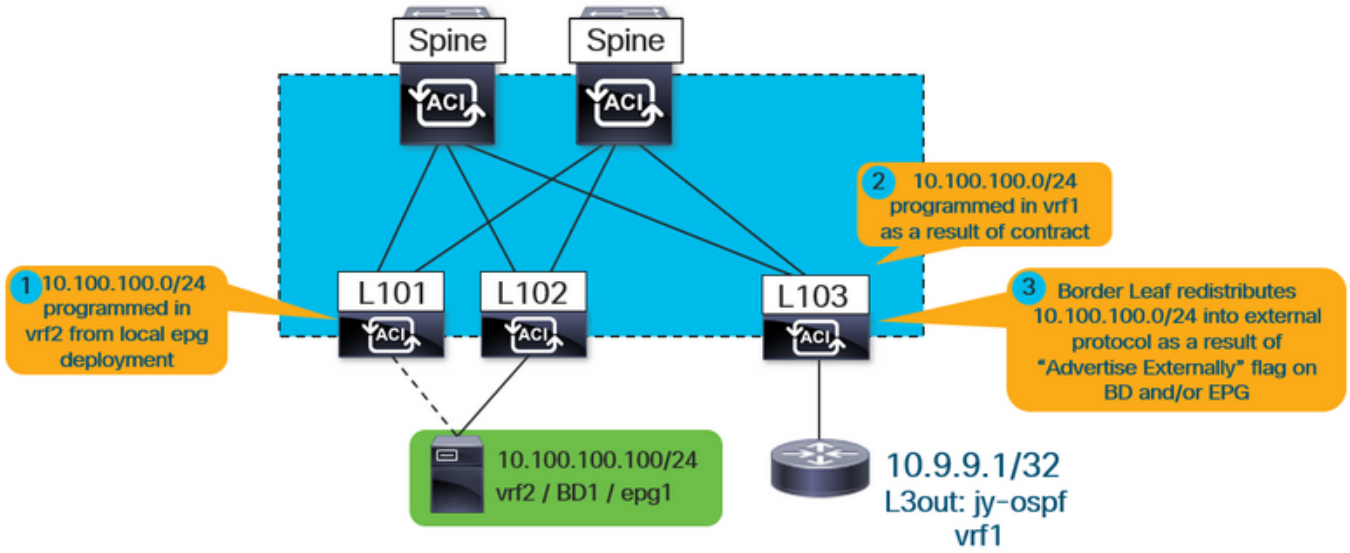
このドキュメントのすべての例は、aciソフトウェア4.2(3j)に基づいています。

VRF xからのブリッジドメイン/EPGサブネットがVRF yにインストールされるのはなぜですか。

このセクションでは、BDまたはEPGのサブネットが予期せず別のvrfにリークされるシナリオを中心に説明します。BD/EPGサブネットがリークされるためには、「Shared Between VRFs」フラグを設定する必要があります。より難しい部分は、どの契約がリークを引き起こしているのかを理解することであり、これがこのセクションで取り上げる内容です。

高レベルでは、BD/EPGサブネットがvrf間で漏出した場合に何が起こるかについてのワークフローです。

図 1.



*#3は、共有l3outをアドバタイズする場合にのみ適用されることに注意してください。#1と#2は、共有l3outが使用されているか、共有サービスが完全に内部であるかに関係なく、常に適用されます。

まず、BDまたはEPGのサブネットの結果として、インストールされたルートがリークしているかどうかをユーザが知るにはどうすればよいのですか。

「**show ip route vrf <name>**」を実行すると、「**pervent**」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、上記のトポロジでは、外部vrf(vrf1)の境界リーフで次のように表示されます。

```
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]
'%
```

```
      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
```

さらに、サブネットが漏出した宛先vrfは、次のコマンドを実行して表示できます。

```
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
```

* (宛先vrfがルックアップvrfと異なる場合でも、vrf交差情報が設定されることに注意してください) 。

上記の出力では、vrf crossing vnidは0x258003 (10進数) または10進数の2457603に設定されています。vnid 2457603が属するvrfはどのように識別されるのですか。

APICからfvCtxオブジェクトに対してクエリを実行し、segidに基づいてフィルタを実行します。

```
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
```

予想どおり、ルートはvrf2 vrfから学習されています。

どの契約が使用されているか、どのepgが提供されているか、どのepgがこのルートをインストールするために消費しているかについては、この時点ではまだ不明です。プロバイダーとコンシューマの関係に関しては、次の2つの点を考慮する必要があります。
1. vrf間の契約関係では、コントラクト (および結果のゾーニングルール) はコンシューマepgのvrfにのみインストールされます。その結果、プロバイダーvrfの「show zoning-rule」には関係が表示されません。
2.コントラクトがコンシューマvrfにのみインストールされている場合でも、プロバイダーvrfはコンシューマvrf BDサブネットのルートを取得する必要があります。つまり、リーフにはコントラクトに対する設定参照が必要です。

ルートがコンシューマVRFに予期せずリークされている場合に、コントラクトを特

定します

リーフ上のipConsオブジェクトは、参照するリーフにインストールされます…

a.)コンシューマvrfにリークされるルート

口。当該関係を成立させる契約

c)プロバイダーとコンシューマepgの関係

次の出力では、「jy:vrf1」はルートがリークされているコンシューマVRFで、「10.100.100.0/24」はリークされているルートです。

```
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       :
```

上記の出力から、contract name is "shared", consumer epg is l3out epg "uni/tn-jy/out-jy-ospf/instP-all", provider epg is "uni/tn-jy/ap-ap1/epg-epg1".

ルートが予期せずプロバイダーVRFにリークされている場合に、コントラクトを特定します

consNodeオブジェクトは、プロバイダーvrfのリーフにインストールされます。これは、リークされるコンシューマVRF内のBDサブネット、コントラクト、および関係のepgを参照します。このオブジェクトを照会する前に、ルートが設定されているBDサブネットを見つけます。これは、apicのfvSubnetオブジェクトを照会することによって実行できます。

```
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。次のコマンドを実行するには、この

VIDとプロバイダーvrfのvnid（ルートが漏出されているVRF）を使用します。

```
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
```

上記の出力から、コントラクト名は「shared」で、コンシューマepgは「uni/tn-jy/ap-ap1/epg-epg1」で、プロバイダーepgは「l3out」です。

ルートが消費されたvzAny契約によって予期せず漏出されている場合の契約の特定

vzAnyの例は、検証の観点から、従来のプロバイダー/コンシューマ関係と同じです。次の例は、この動作の様子を示しています。vrf間契約は、コンシューマとしてvzAnyでのみサポートされることに注意してください。

vzAnyの例1:コンシューマVRFに突然漏洩したルート

コンシューマVRFで検証が行われた最初の例と同様に、ipConsオブジェクトが再度使用されます。

```
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        :
```

上記の出力から、contract name is "shared"、コンシューマepgはvrf1 vzAny "tn-jy/ctx-vrf1/any"、プロバイダーepgは"uni/tn-jy/ap-ap1/epg-epg1"です。

vzAnyの例2:ルートがプロバイダーVRFに予期せずリークする

プロバイダーvrfで検証が行われた場所を調べた2番目の例と同様に、consNodeオブジェクトが再度使用されます。漏出したサブネットが設定されているBDのbd名と、漏出したVRFのvnidを取得することを忘れないでください。

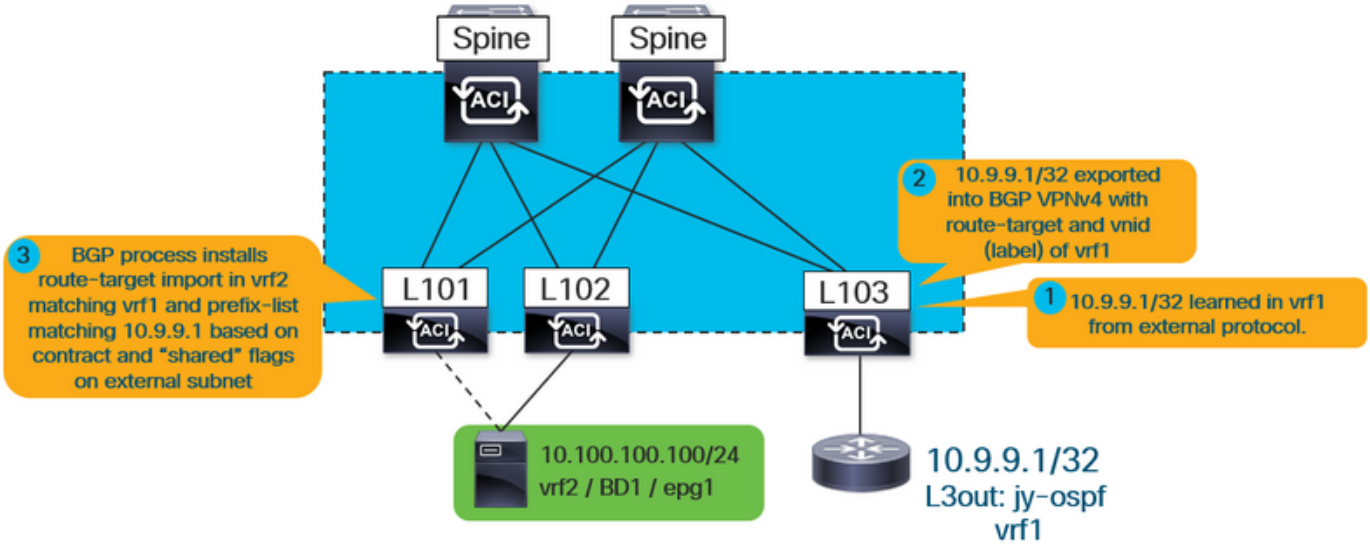
```
leaf103# moquery -c consNode -f 'cons.Node.dn*"vxlan-2949122"' -f 'cons.Node.dn*"tn-jy/BD-bd1"'
Total Objects shown: 1

# 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
```

上記の出力から、contract name is "shared"、コンシューマepgはvrf2 vzAny "tn-jy/ctx-vrf2/any"、プロバイダーepgはl3out "tn-jy/out-jy-ospf/instP-all"です。

VRF yからの外部ルートがVRF xにインストールされるのはなぜですか。

高レベルでは、l3outで学習された（外部）ルートがvrf間で漏出された場合に何が起こるかについてのワークフローです。

図 2：



上記のように、内部vrf（この場合はvrf2）は、vrf1に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。また、l3outで「共有ルート制御サブネット」フラグが選択された全てに一致するプレフィックスリストエントリを持つインポートマップもインストールします。

どのepgがプロバイダーまたはコンシューマであるかに関係なく、検証の手順は同じです。これは、契約が常にルートターゲットのインポートと、ルートをリークしてインストールされる対応するプレフィックスリストの原因になるためです。

まず、ルートが実際にl3outを介して学習されていることを確認します。

```
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
```

上記の例では、オーバーレイの別のリーフを指すfabric bgpプロセスから学習されたという事実は、これがl3outから発生したことを示しています。

次の情報を実行して、どのVRFから学習されたかについての詳細を取得します。

```
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
```

このドキュメントで示すように、rewrite vnid 0x2d0002 / 2949122が宛先vrfです。外部ルートの例でrw-vnid値が0以外の値に設定されている場合、これは別のvrfから学習されたことを示します。apicで**moquery -c fvCtx -f 'fv.Ctx.seg=="2949122"を実行する**と、これはvrf1に属していることがわかります。

次に、bgpプロセスに関連付けられたインポートのルートマップとルートターゲットのインポートを見つけます。

```
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、独自のルートターゲット(65001:2457603)をエクスポートおよびインポートしています。 また、65001:2949122をインポートしています。2949122 RTは、インポートするvrf vnid(vrf1)に対応しています。bgpRtCtrlMapPは、プレフィックスリストを含むインポートルートマップのオブジェクト名です。bgpRttEntryは、インポートルートターゲットのオブジェクト名です。

次に、外部vrfルート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内にインストールされているすべてのプレフィックスリストを照会します。

```
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している必要があります。「exact / inexact」属性は、「aggregate shared」フラグが外部サブネットに設定されているかどうかを示します。0.0.0.0/0プレフィクスにinexactフラグを付けると、より具体的な（事実上すべてのもの）すべてのルートに一致することが示されます。 10.9.9.0/24プレフィクスとexactフラグは、/24とだけ一致することを示します。

予期せず漏出しているルートに一致するエントリ（またはエントリ）を検索します。この場合、プレフィックス10.9.9.1/32は、上記の出力でent-2とent-3によって照合されます。

プレフィックスリスト名を使用して、一致するルートマップ内のシーケンス番号を検索します。

```
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であることを示しています。最後に、2457603-shared-svc-leak route-map内でルートマップエントリ1001を作成するコントラクトを理解します。これは、fvAppEpGConsオブジェクトからリーフ上で照会できます。

```
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「shared」、プロバイダーepgが「tn-jy/ap-ap1/epg-epg1」、コンシューマl3out epgが「tn-jy/out-jy-ospf/instP-all」であることを示しています

要約

BD/EPGサブネットから漏出したルート

漏出ルートの「show ip route」で「pervacive」フラグが設定されている場合、設定されたBD/EPGサブネットから漏出されます。次の2つのコマンドを使用して、この情報がリークする原因となっている契約関係を確認できます。ルートが予期せずインストールされたリーフで実行されます。

ルートが予期せず漏出するVRFがコンシューマの場合：
moquery -c ipCons -f 'ip.Cons.dn*"jy:vrf1/rt-\[10.100.100.0/24\]" *<—jy:vrf1は、ルートが漏出されるvrfの名前で、ルートは10.100.100.0/24です*

ルートが予期せず漏出されるVRFがプロバイダーの場合：
moquery -c consNode -f 'cons.Node.dn*"2949122"' -f 'cons.Node.dn*"tn-jy/BD-bd1" *<—2949122は、ルートが漏出されるvrfのvnidです。tn-jy/BD-bd1は、サブネットが設定されているBDの名前です（ルートが漏出されるvrf内）。*

L3outから漏出したルート

内部ファブリックiBGPプロセスを通じて漏出ルートが学習され、**vsh -c "show ip route x.x.x/y detail vrf <name>"を実行してゼロ以外のrw-vnid値が表示された場合**は、ルートは別のvrfのl3outから学習されます。どのepgがコンシューマで、どのプロバイダーであるかに関係なく、検証は同じです。

1. *内部vrf bgpプロセスでshared services import route-mapを特定します。*
show bgp process vrf jy:vrf2 | grep "ルートマップのインポート" *<—jy:vrf2は、ルートが漏出される内部vrfです*

2.リークしたルートに一致する共有サービスルートマップ内のプレフィックスリストを特定します。
moquery -c rtpfxEntry -f 'rtpfx.Entry.dn*"pfxlist-IPv4'.*'2457603-shared-svc-leak"' | egrep "criteria|dn|pfx|toPfxLen" *<—2457603は、この例の内部vrfのvnidです*

3.ルートを参照するプレフィックスリストを見つけたら、どのルートマップシーケンス番号がリストを参照しているかを特定します。
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はプレフィックスリスト名です*

4. rtmapとエントリ番号を使用して、次のコマンドを実行し、ルートマップエントリをプッシュしたコントラクト関係を確認します。
moquery -c fvAppEpGCons -f 'fv.AppEpGCons.dn*"rtmap-2457603-shared-svc-leak/ent-1001"' *<—rtmap-2457603-shared-svc-leak/ent-1001は、ステップ3のルートマップ名とエントリ番号です。*