VRF в «С-Терра Шлюз» 5.0

Введение

VRF (virtual routing and forwarding) в Linux – это надстройка над существующей системой правил направления трафика в различные таблицы маршрутизации (PBR). Она позволяет создать несколько изолированных виртуальных маршрутизаторов на одном устройстве. Изоляция происходит на 2-м (L2) и 3-м уровне OSI.

В Linux VRF-домен представлен специальным сетевым интерфейсом – VRF-устройством (master device). Это виртуальный интерфейс, который не передает трафик сам по себе, а служит некоторым контейнером, в который помещаются другие интерфейсы и к которому привязывается выделенная таблица маршрутизации.

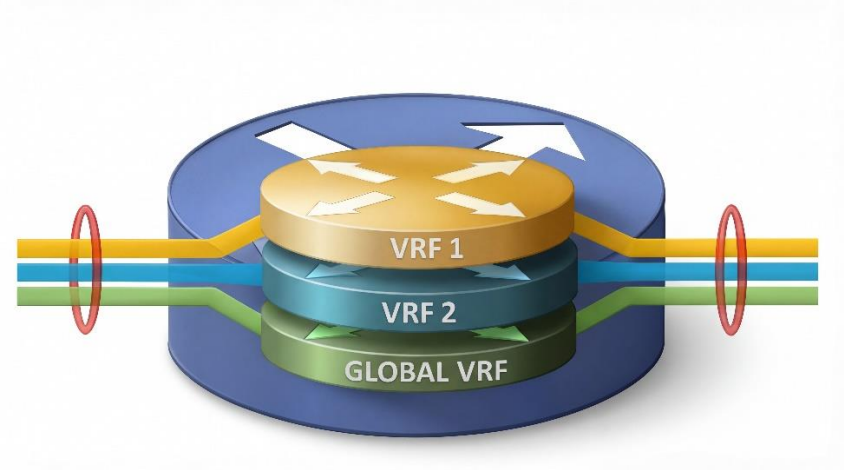


Рисунок 1. Несколько VRF в одном роутере

При помощи VRF и GRE/stVTI интерфейсов можно изолировать трафик с пересекающимся адресным пространством и защитить этот трафик посредством IPsec VPN. Данный подход реализует парадигму multitenancy (мультиарендность) – элемент архитектуры программного обеспечения, где единый экземпляр приложения обслуживает множество организаций-клиентов. Мультиарендность противопоставляется архитектуре из множественных экземпляров, где для каждой организации-клиента создаются отдельные программные экземпляры.

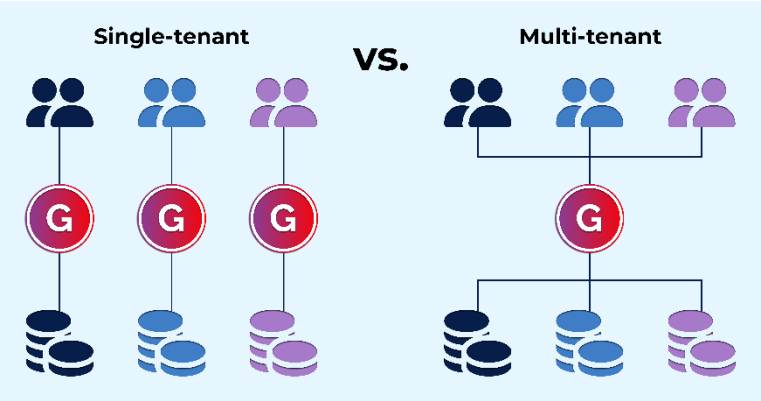


Рисунок 2. Мультиарендность

Ограничения

1. WAN интерфейсы нельзя помещать в VRF (сценарий Front door VRF), так как IKE может работать только в Global VRF.
2. VRF для управления (management VRF) не поддерживается, так как модули/ПО snmp, syslog, ssh, upagent (агент управления для системы управления) не могут работать в VRF.
3. К интерфейсу, входящему в VRF, можно прикреплять криптокарту только в том случае, если этот интерфейс является stVTI интерфейсом.
4. В сборке Шлюза 5.0.25337 (посмотреть при помощи команды `/opt/VPNagent/bin/ver_show`) консоль cisco-like некорректно работает с GRE-интерфейсами, на которых настроен VRF, из-за чего неверно считывается часть параметров (tunnel mode, tunnel source, tunnel destination и tunnel key). Поэтому настройка GRE интерфейсов должна осуществляться не в консоли, а через файлы. Настраивать GRE через консоль cisco-like можно только начиная со сборки Шлюза 5.0.26745 и новее.
5. Фильтрация трафика на основе контроля состояния сессий (stateful) не совместима с VRF в случае, когда пересекается адресное пространство. Это вызвано тем, что таблица соединений является глобальной и не учитывает VRF.
6. При изменении параметров сетевых интерфейсов, которые были добавлены в VRF, нужно перезапускать VRF (`ifdown <vrf_name>; ifup <vrf_name>`).
7. Если между **двумя** (либо для случая, когда адреса партнеров не известны и на центральном шлюзе используется dynamic crypto map) шлюзами нужно построить несколько IPsec туннелей с пересекающимся адресным пространством, то в криптокартах (crypto map) для разных VRF нужно использовать разные crypto isakmp profile с разным identity, например:

```
crypto pki certificate map FIND_LOCAL_CERT_BY_ISSUER_NAME 0
 issuer-name eq C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA
!
crypto isakmp profile ISAKMP_PROFILE_VTI0_VRF_RED
 match identity email VRF-RED@spoke1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-RED@hub1
 no set identity mapping
 set policy 1
!
crypto isakmp profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 match identity email VRF-BLUE@spoke1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-BLUE@hub1
 no set identity mapping
 set policy 1
```

Особенности порядка включения сетевых интерфейсов

По умолчанию, сетевые интерфейсы, которые описываются в конфигурационных файлах в директории `/etc/network/interfaces.d/`, включаются до интерфейсов, описанных в `/etc/network/interfaces`.

В связи с этим следует не допускать ситуации, когда VRF интерфейсы при поднятии включают в себя интерфейсы, которые на этот момент могут еще не существовать в системе (например, stVTI или VLAN), так как это приведет к ошибкам конфигурации.

Примечание. Интерфейсы, описанные в конфигурационных файлах в директории `/etc/network/interfaces.d/`, включаются в лексикографическом порядке (по именам файлов).

Ниже представлены два варианта настроек сетевых интерфейсов для обеспечения правильного порядка включения VRF интерфейсов.

Вариант 1. В файле `/etc/network/interfaces` строку `source /etc/network/interfaces.d/*` перенести в конец (ниже секции, обозначенной строками `###netifcfg-begin###` и `###netifcfg-end###`), пример:

```
root@Hub1:~# cat /etc/network/interfaces
# Отсюда убрали (достаточно закомментировать)
# source /etc/network/interfaces.d/*

...

# loopback configuration
auto lo
iface lo inet loopback

###netifcfg-begin###

...

###netifcfg-end###

# Перенесли сюда
source /etc/network/interfaces.d/*
```

Вариант 2. Допускается перенести описание интерфейса из директории `/etc/network/interfaces.d/` в файл `/etc/network/interfaces` (ниже секции, обозначенной строками `###netifcfg-begin###` и `###netifcfg-end###`), пример:

```
root@Hub1:~# cat /etc/network/interfaces
...
# Сохраняем строку в исходном варианте
source /etc/network/interfaces.d/*
...

# loopback configuration
auto lo
iface lo inet loopback

###netifcfg-begin###

...

###netifcfg-end###

# Настраиваем VRF ниже
auto vrf_blue
iface vrf_blue inet manual
...
```

Схема стенда

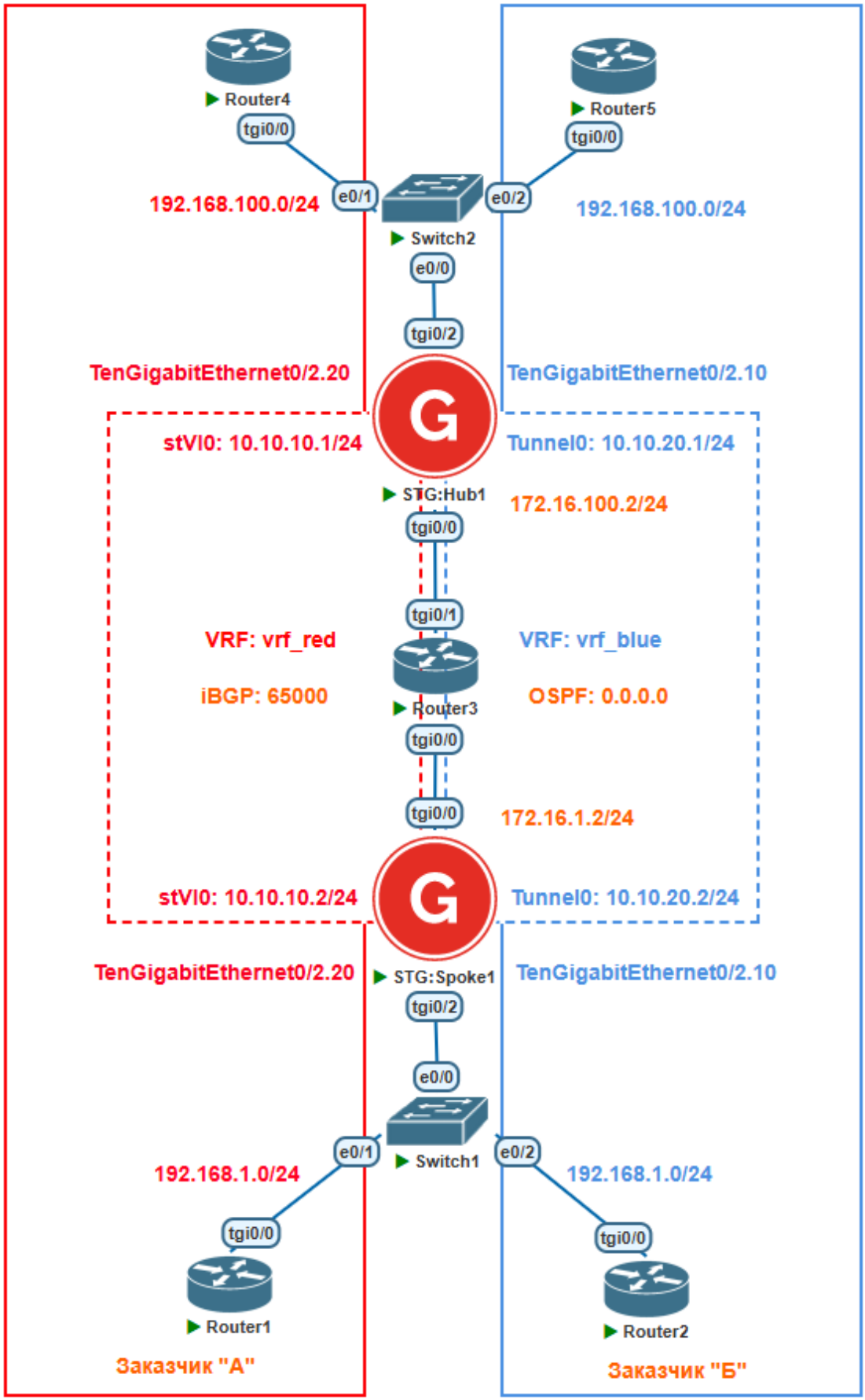


Рисунок 3. Схема стенда, реализующая концепцию мультиарендности

Логика работы

В данном документе будет рассмотрена настройка шлюзов STG:Hub1 и STG:Spoke1 (см. схему стенда) для обеспечения корректной работы при пересечении защищаемых подсетей 192.168.1.0/24 и 192.168.100.0/24 у разных заказчиков (например, под контролем заказчика «А» оборудование Router1 и Router4, а под контролем заказчика «Б» оборудование Router2 и Router5). Заказчики не могут поменять адресацию по тем или иным причинам.

На шлюзах STG:Hub1 и STG:Spoke1 будут созданы по два VRF интерфейса с именем vrf_blue и vrf_red.

VRF **vrf_blue**:

- таблица маршрутизации – 10;
- протокол маршрутизации – OSPF;
- интерфейсы – GRE (Tunnel0) и VLAN (TenGigabitEthernet0/2.10).

VRF **vrf_red**:

- таблица маршрутизации – 20;
- протокол маршрутизации – iBGP;
- интерфейсы – stVTI (stVTI0) и VLAN (TenGigabitEthernet0/2.20).

Шлюз STG:Hub1 анонсирует маршрут по умолчанию в сторону STG:Spoke1 по протоколам OSPF и iBGP.

Обеспечение безопасного взаимодействия между шлюзами достигается путем шифрования и туннелирования трафика с применением отечественных отраслевых стандартов ГОСТ и протокола IPsec. В рамках данного документа шифроваться будет весь трафик, который маршрутизируется через GRE и stVTI.

Настройка

Интерфейсы GRE, VLAN, stVTI были созданы предварительно (пример настроек представлен в Приложении).

На шлюзах STG:Hub1 и STG:Spoke1 выполните следующие действия с учетом IP адресов и имен интерфейсов.

Примеры ниже будут даны для устройства STG:Hub1 (hostname – Hub1).

1. Перенесите строку `source /etc/network/interfaces.d/*` в конец файла `/etc/network/interfaces` (чтобы VRF интерфейсы создавались строго позже интерфейсов, настроенных через cisco-like консоль).
2. Создайте файл `/etc/network/interfaces.d/interfaces` и добавьте в него следующее содержимое:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/interfaces
```

```
auto vrf_blue
iface vrf_blue inet manual
pre-up ip link del $IFACE type vrf 2>/dev/null || true
pre-up ip route flush table 10 2>/dev/null || true
pre-up ip link add $IFACE type vrf table 10
pre-up ip link set dev $IFACE up
pre-up ip route add table 10 unreachable default metric 4278198272
pre-up ip link set dev gre_0 master $IFACE
pre-up ip link set dev eth2_10 master $IFACE
post-down ip link set dev $IFACE down

auto vrf_red
iface vrf_red inet manual
pre-up ip link del $IFACE type vrf 2>/dev/null || true
pre-up ip route flush table 20 2>/dev/null || true
pre-up ip link add $IFACE type vrf table 20
```

```
pre-up ip link set dev $IFACE up
pre-up ip route add table 20 unreachable default metric 4278198272
pre-up ip link set dev st_vti0 master $IFACE
pre-up ip link set dev eth2_20 master $IFACE
post-down ip link set dev $IFACE down
```

3. Для того, чтобы VRF интерфейсы vrf_blue и vrf_red появились в системе, выполните команды ifup vrf_blue и ifup vrf_red после чего убедитесь, что данные интерфейсы доступны в системе:

```
root@Hub1:~# ifup vrf_blue
root@Hub1:~# ifup vrf_red
root@Hub1:~# ip address show vrf_blue
```

```
15: vrf_blue: <NOARP,MASTER,UP,LOWER_UP> mtu 65575 qdisc noqueue state UP group default
qlen 10000
    link/ether ae:9d:81:5b:84:15 brd ff:ff:ff:ff:ff:ff
```

```
root@Hub1:~# ip address show vrf_red
```

```
16: vrf_red: <NOARP,MASTER,UP,LOWER_UP> mtu 65575 qdisc noqueue state UP group default
qlen 10000
    link/ether 3a:14:80:c0:7d:a3 brd ff:ff:ff:ff:ff:ff
```

4. На примере VRF vrf_blue убедитесь, что интерфейсы, добавленные в данный VRF, действительно входят в него (master vrf_blue):

```
root@Hub1:~# ip address show gre_0
```

```
11: gre_0@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue master vrf_blue
state UNKNOWN group default qlen 10000
    link/gre 0.0.0.0 peer 172.16.1.2
    inet 10.10.20.1/24 brd 10.10.20.255 scope global gre_0
       valid_lft forever preferred_lft forever
    inet6 fe80::c0a8:6401/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::a0a:a01/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::a0a:1401/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::ac10:6402/64 scope link
       valid_lft forever preferred_lft forever
```

```
root@Hub1:~# ip address show eth2_10
```

```
13: eth2_10@eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master
vrf_blue state UP group default qlen 10000
    link/ether 50:63:5f:01:b8:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.1/24 brd 192.168.100.255 scope global eth2_10
       valid_lft forever preferred_lft forever
    inet6 fe80::3ff0:5835:9c79:2971/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

5. Запустите FRR и добавьте его в автозагрузку (перезапустите, если уже был запущен ранее):

```
root@Hub1:~# systemctl start frr
root@Hub1:~# systemctl enable frr
```

6. Выполните настройку BGP для **vrf_red**:

```
root@Hub1:~# vtysh
```

```
Hello, this is FRRouting (version 10.3.3).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
Hub1# conf t
Hub1(config)# router bgp 65000 vrf vrf_red
Hub1(config-router)# bgp router-id 10.10.10.1
Hub1(config-router)# neighbor 10.10.10.2 remote-as 65000
Hub1(config-router)# neighbor 10.10.10.2 default-originate
Hub1(config-router)# address-family ipv4 unicast
```

```
Hub1(config-router-af)# network 192.168.100.0/24
Hub1(config-router-af)# exit-address-family
Hub1(config-router)# exit
Hub1(config)#
Hub1(config)# exit
Hub1# write
```

```
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
```

7. Выполните настройку OSPF для **vrf_blue**:

```
Hub1# conf t
Hub1(config)# router ospf vrf vrf_blue
Hub1(config-router)# ospf router-id 10.10.20.1
Hub1(config-router)# passive-interface default
Hub1(config-router)# network 10.10.20.0/24 area 0
Hub1(config-router)# network 192.168.100.0/24 area 0
Hub1(config-router)# default-information originate
Hub1(config-router)# exit
Hub1(config)#
Hub1(config)# interface gre_0
Hub1(config-if)# no ip ospf passive
Hub1(config-if)# ip ospf network point-to-point
Hub1(config-if)# end
Hub1# write
```

```
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
```

8. Выполните настройку IKE.

   По умолчанию в рамках протокола IKE при аутентификации на сертификатах в качестве identity отправляется DN из локального сертификата шлюза. Но так как мы, по сути, заводим два инстанса (по количеству VRF) с пересекающимися параметрами (в данном случае это защищаемые подсети), то на уровне IKE (и далее на уровне IPsec) нужно также разграничить эти инстансы для корректной работы и для упрощения диагностики в случае возникновения проблем.

   Идея состоит в следующем. Локальный сертификат на шлюзах остается один (можно завести второй, но это более сложный случай с точки зрения последующей эксплуатации, так как потребуется обновлять уже два сертификата). Заводятся два профиля IKE (ISAKMP). В данных профилях отключается маппирование (по сути - использование) identity из сертификата (DN). В качестве identity будет использован email (по сути, просто пользовательская строка в формате `<строка>@<строка>`). В рамах данного примера, исключительно для удобства, вводится следующий формат строки `<VRF-NAME>@hostname`. Параметр `match identity email <VRF-NAME>@hostname` задает identity партнера, а параметр `self-identity email <VRF-NAME>@hostname` задает локальный identity самого шлюза (данный identity будет отослан партнеру по IKE). Данные параметры чувствительны к регистру букв и должны быть согласованны между партнерами по IKE.

   Параметр `self-certificate <cert_map_name>` используется для выбора локального сертификата шлюза (данный сертификат буде отослан партнеру по IKE), если не задать этот параметр, то в логах будет ошибка «Searching local certificate failed».

```
crypto pki certificate map FIND_LOCAL_CERT_BY_ISSUER_NAME 0
 issuer-name eq C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
```

```
crypto isakmp profile ISAKMP_PROFILE_VTI0_VRF_RED
 match identity email VRF-RED@spoke1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-RED@hub1
 no set identity mapping
 set policy 1
!
crypto isakmp profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 match identity email VRF-BLUE@spoke1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-BLUE@hub1
 no set identity mapping
 set policy 1
```

9. Выполните настройку IPsec.

   В данном документе описывается пример настройки шифрования в соответствии с концепцией, когда трафик для шифрования в политике IPsec определяется единожды, а то, что действительно будет направлено в IPsec туннель определяется маршрутами (Route Based IPsec).

   Примечание 1. Когда криптокарта прикрепляется к stVTI интерфейсу, то трафик перехватывается для зашифрования в том VRF, в которым находится stVTI интерфейс. Зашифрованный трафик маршрутизируется уже в соответствии с глобальной таблицей маршрутизации (Global VRF). Для данного случая можно использовать фильтр для защищаемого трафика: `permit ip any any`.

   Примечание 2. Для варианта, когда нужно настроить GRE over IPsec (шифрование GRE трафика) криптокарта прикрепляется к WAN интерфейсу, через который требуется маршрутизировать этот GRE трафик. В этом случае трафик перехватывается для зашифрования уже в Global VRF и зашифрованный трафик маршрутизируется в соответствии с глобальной таблицей маршрутизации. Для данного случая **НЕЛЬЗЯ** использовать фильтр для защищаемого трафика: «`permit ip any any`», иначе некоторые пакеты будут удалены с диагностикой «packet must be protected with IPsec» в `klogview -f drop`.

   Общее правило: фильтры для защищаемого трафика (traffic selectors) в рамках одной криптокарты **НЕ ДОЛЖНЫ** пересекаться.

   Нужно настроить две криптокарты, каждая для своего инстанса (VRF). К криптокрте прикрепляется свой isakmp profile, настроенный выше.

   В данном документе наименование объектов в консоли cisco-like условно, но верное наименование существенно поможет в анализе информации о туннелях (`sa_mgr show -detail, klogview -f all`) и о событиях в логах (`/var/log/cspvpngate.log`).

```
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
ip access-list extended VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1
 permit ip any any
!
ip access-list extended GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1
 permit gre host 172.16.100.2 host 172.16.1.2
!
crypto map VTI0_VRF_RED_VPN 1 ipsec-isakmp
 description VTI over IPsec from VRF-red to Spoke1 (global VRF)
 match address VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile ISAKMP_PROFILE_VTI0_VRF_RED
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map GLOBAL_VRF_VPN 1 ipsec-isakmp
```

```
 description GRE over IPsec from VRF-blue to Spoke1 (global VRF)
 match address GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 description WAN
 ip address 172.16.100.2 255.255.255.0
 crypto map GLOBAL_VRF_VPN
!
interface stVTI0
 description VRF-RED
 crypto map VTI0_VRF_RED_VPN
```

Проверка работоспособности стенда

Проверка VRF

На шлюзе STG:Spoke1 (на другом шлюзе действуйте по аналогии с учетом специфики) выполните следующие команды для проверки работоспособности VRF:

```
root@Spoke1:~# ip vrf show
```

```
Name              Table
-----------------------
vrf_blue            10
vrf_red             20
```

```
root@Spoke1:~# ip address show
```

```
...
8:  eth2_10@eth2:  <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu  1500  qdisc  noqueue  master
vrf_blue state UP group default  qlen 10000
    link/ether 50:f6:d2:01:a8:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth2_10
       valid_lft forever preferred_lft forever
    inet6 fe80::1da2:ee40:7920:83f7/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
9:  eth2_20@eth2:  <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu  1500  qdisc  noqueue  master
vrf_red state UP group default  qlen 10000
    link/ether 50:f6:d2:01:a8:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth2_20
       valid_lft forever preferred_lft forever
    inet6 fe80::3278:5373:7f3:635f/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
...
13: gre_0@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue master vrf_blue
state UNKNOWN group default  qlen 10000
    link/gre 0.0.0.0 peer 172.16.100.2
    inet 10.10.20.2/24 brd 10.10.20.255 scope global gre_0
       valid_lft forever preferred_lft forever
    inet6 fe80::a0a:a02/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::a0a:1402/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::c0a8:101/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::ac10:102/64 scope link
       valid_lft forever preferred_lft forever
```

```
14: st_vti0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue master vrf_red state
UNKNOWN group default qlen 100
    link/ether c6:ed:55:28:9d:34 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.2/24 brd 10.10.10.255 scope global st_vti0
       valid_lft forever preferred_lft forever
    inet6 fe80::e246:e197:8f9d:afed/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
15: vrf_blue: <NOARP,MASTER,UP,LOWER_UP> mtu 65575 qdisc noqueue state UP group default
qlen 10000
    link/ether ae:9d:81:5b:84:15 brd ff:ff:ff:ff:ff:ff
16: vrf_red: <NOARP,MASTER,UP,LOWER_UP> mtu 65575 qdisc noqueue state UP group default
qlen 10000
    link/ether 3a:14:80:c0:7d:a3 brd ff:ff:ff:ff:ff:ff
```

Если VRF интерфейсы не созданы или необходимые интерфейсы не добавлены в VRF проверьте настройки VRF и перезапустите их.

```
root@Spoke1:~# if_mgr show
```

```
Logical network interface "TenGigabitEthernet0/0":
        Physical name exact template: "eth0"

        Physical name: eth0
        State:         UP
        Index:         2
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           1500
        MAC addr:      50:F6:D2:01:A8:00
        Flags:         0x7(UP,MAC,BROADCAST)
        IP addr:       172.16.1.2 mask 255.255.255.0 brd 172.16.1.255
        IPv6 addr:     fe80::cb45:cede:5d0f:5cbe/64 SCOPE_LINK

        ...

Logical network interface "TenGigabitEthernet0/2.10":
        Physical name exact template: "eth2_10"

        Physical name: eth2_10
        State:         UP
        Index:         8
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           1500
        MAC addr:      50:F6:D2:01:A8:02
        Flags:         0x647(UP,MAC,BROADCAST,VLAN,VRF,LINK_SLAVE)
        Master_if:     ifidx=15, ifname=vrf_blue
        Routing table: 10
        IP addr:       192.168.1.1 mask 255.255.255.0 brd 192.168.1.255
        IPv6 addr:     fe80::1da2:ee40:7920:83f7/64 SCOPE_LINK

Logical network interface "TenGigabitEthernet0/2.20":
        Physical name exact template: "eth2_20"

        Physical name: eth2_20
        State:         UP
        Index:         9
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           1500
        MAC addr:      50:F6:D2:01:A8:02
        Flags:         0x647(UP,MAC,BROADCAST,VLAN,VRF,LINK_SLAVE)
        Master_if:     ifidx=16, ifname=vrf_red
```

```
        Routing table: 20
        IP addr:       192.168.1.1 mask 255.255.255.0 brd 192.168.1.255
        IPv6 addr:     fe80::3278:5373:7f3:635f/64 SCOPE_LINK

Logical network interface "Tunnel0":
        Physical name exact template: "gre_0"

        Physical name: gre_0
        State:         UP
        Index:         13
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           1400
        MAC addr:      00:00:00:00
        Flags:         0x703(UP,MAC,GRE,VRF,LINK_SLAVE)
        Master_if:     ifidx=15, ifname=vrf_blue
        Routing table: 10
        IP addr:       10.10.20.2 mask 255.255.255.0 brd 10.10.20.255
        IPv6 addr:     fe80::ac10:102/64 SCOPE_LINK
        IPv6 addr:     fe80::c0a8:101/64 SCOPE_LINK
        IPv6 addr:     fe80::a0a:1402/64 SCOPE_LINK
        IPv6 addr:     fe80::a0a:a02/64 SCOPE_LINK

Logical network interface "stVTI0":
        Physical name exact template: "st_vti0"

        Physical name: st_vti0
        State:         UP
        Index:         14
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           1500
        MAC addr:      C6:ED:55:28:9D:34
        Flags:         0xe07(UP,MAC,BROADCAST,VRF,LINK_SLAVE,LINK_FOREIGN)
        Master_if:     ifidx=16, ifname=vrf_red
        Routing table: 20
        IP addr:       10.10.10.2 mask 255.255.255.0 brd 10.10.10.255
        IPv6 addr:     fe80::e246:e197:8f9d:afed/64 SCOPE_LINK

Uncontrolled physical interfaces:

        ...

        Physical name: vrf_blue
        State:         UP
        Index:         15
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           65575
        MAC addr:      AE:9D:81:5B:84:15
        Flags:         0x223(UP,MAC,MASTER,VRF)
        Routing table: 10

        Physical name: vrf_red
        State:         UP
        Index:         16
        IPv4:          ENABLED
        IPv6:          ENABLED
        MTU:           65575
        MAC addr:      3A:14:80:C0:7D:A3
        Flags:         0x223(UP,MAC,MASTER,VRF)
        Routing table: 20
```

Если в выводе команды `if_mgr show` на каком-то из интерфейсов отсутствует информация о VRF или является некорректной, но в предыдущих выводах других команд было все ожидаемо, то обратитесь в службу технической поддержки С-Терра.

Проверка маршрутизации

На шлюзе STG:Spoke1 (на другом шлюзе действуйте по аналогии с учетом специфики) выполните следующие команды для проверки работоспособности маршрутизации:

```
root@Spoke1:~# ip route show vrf vrf_red
```

```
default nhid 50 via 10.10.10.1 dev st_vti0 proto bgp metric 20
unreachable default metric 4278198272
10.10.10.0/24 dev st_vti0 proto kernel scope link src 10.10.10.2
192.168.1.0/24 dev eth2_20 proto kernel scope link src 192.168.1.1
192.168.100.0/24 nhid 50 via 10.10.10.1 dev st_vti0 proto bgp metric 20
```

```
root@Spoke1:~# ip route show vrf vrf_blue
```

```
default nhid 52 via 10.10.20.1 dev gre_0 proto ospf metric 20
unreachable default metric 4278198272
10.10.20.0/24 dev gre_0 proto kernel scope link src 10.10.20.2
192.168.1.0/24 dev eth2_10 proto kernel scope link src 192.168.1.1
192.168.100.0/24 nhid 52 via 10.10.20.1 dev gre_0 proto ospf metric 20
```

Обязательно убедитесь в том, что выводах присутствует маршрут «unreachable default metric 4278198272», который предотвращает утечку трафика (leaking).

Также проконтролируйте что в VRF есть необходимые непосредственно присоединенные маршруты от сетевых интерфейсов, которые должны быть в конкретном VRF.

Из консоли FRR выполните проверку протоколов динамической маршрутизации:

```
Spoke1# show bgp vrf all
```

```
Instance vrf_red:
No BGP prefixes displayed, 0 exist
Spoke1# show bgp vrf all summary

IPv4 Unicast Summary:
BGP router identifier 10.10.10.2, local AS number 65000 VRF vrf_red vrf-id 16
BGP table version 3
RIB entries 4, using 512 bytes of memory
Peers 1, using 23 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd
PfxSnt Desc
10.10.10.1      4      65000      1511      1510        3    0    0 1d01h06m            2
1 FRRouting/10.3.3

Total number of neighbors 1
```

```
Spoke1# show ip ospf vrf all neighbor
```

```
VRF Name: vrf_blue

Neighbor ID     Pri State           Up Time         Dead Time Address         Interface
RXmtLL
10.10.20.1            1  Full/-              1d01h07m              33.753s  10.10.20.1
gre_0:10.10.20.2                     00

OSPF is not enabled
```

```
Spoke1# show ip route vrf all
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
```

```
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

IPv4 unicast VRF default:
K>* 0.0.0.0/0 [0/0] via 172.16.1.1, eth0, weight 1, 1d01h05m
L   10.10.10.2/32 is directly connected, st_vti0 inactive, weight 1, 1d01h05m
C>* 172.16.1.0/24 is directly connected, eth0, weight 1, 1d01h05m
L * 172.16.1.2/32 is directly connected, eth0, weight 1, 1d01h05m
L>* 172.16.1.2/32 is directly connected, eth0, weight 1, 1d01h05m
L   192.168.1.1/32 is directly connected, eth2_20 inactive, weight 1, 1d01h05m

IPv4 unicast VRF vrf_blue:
O>* 0.0.0.0/0 [110/10] via 10.10.20.1, gre_0, weight 1, 1d00h57m
K * 0.0.0.0/0 [255/8192] unreachable (ICMP unreachable), weight 1, 1d01h05m
O   10.10.20.0/24 [110/10] is directly connected, gre_0, weight 1, 1d01h05m
C>* 10.10.20.0/24 is directly cone cted, gre_0, weight 1, 1d01h05m
L>* 10.10.20.2/32 is directly connected, gre_0, weight 1, 1d01h05m
O   192.168.1.0/24 [110/100] is directly connected, eth2_10, weight 1, 1d01h05m
C>* 192.168.1.0/24 is directly connected, eth2_10, weight 1, 1d01h05m
L * 192.168.1.1/32 is directly connected, eth2_10, weight 1, 1d01h05m
L>* 192.168.1.1/32 is directly connected, eth2_10, weight 1, 1d01h05m
O>* 192.168.100.0/24 [110/110] via 10.10.20.1, gre_0, weight 1, 1d01h05m

IPv4 unicast VRF vrf_red:
B>* 0.0.0.0/0 [200/0] via 10.10.10.1, st_vti0, weight 1, 1d00h16m
K * 0.0.0.0/0 [255/8192] unreachable (ICMP unreachable), weight 1, 1d01h05m
C>* 10.10.10.0/24 is directly connected, st_vti0, weight 1, 1d01h05m
L>* 10.10.10.2/32 is directly connected, st_vti0, weight 1, 1d01h05m
C>* 192.168.1.0/24 is directly connected, eth2_20, weight 1, 1d01h05m
L * 192.168.1.1/32 is directly connected, eth2_20, weight 1, 1d01h05m
L>* 192.168.1.1/32 is directly connected, eth2_20, weight 1, 1d01h05m
B>* 192.168.100.0/24 [200/0] via 10.10.10.1, st_vti0, weight 1, 1d01h05m
```

Убедитесь в том, что необходимые соседства установлены и маршруты добавлены в нужные VRF. Если это не так проверьте настройки на обеих сторонах и откорректируйте при необходимости.

Проверка таблицы соседей на L2

На шлюзе STG:Spoke1 (на другом шлюзе действуйте по аналогии с учетом специфики) запустите ICMP трафик с устройств Router1 и Router4 до 192.168.1.1 и выполните следующие команды для проверки состояния таблиц соседей:

```
root@Spoke1:~# ip neigh show
```

```
172.16.1.1 dev eth0 lladdr 50:70:07:01:b1:00 REACHABLE
192.168.1.2 dev eth2_10 lladdr 50:a7:a6:01:f0:00 REACHABLE
192.168.1.2 dev eth2_20 lladdr 50:b7:07:01:a9:00 REACHABLE
```

```
root@Spoke1:~# ip neigh show vrf vrf_blue
```

```
192.168.1.2 dev eth2_10 lladdr 50:a7:a6:01:f0:00 REACHABLE
```

```
root@Spoke1:~# ip neigh show vrf vrf_red
```

```
192.168.1.2 dev eth2_20 lladdr 50:b7:07:01:a9:00 REACHABLE
```

Убедитесь в том, что в общей таблице соседства и таблицы для конкретного VRF добавлены правильные записи. Если это не так, то проверьте настройки VRF и входящих в него сетевых интерфейсов, откорректируйте при необходимости.

Проверка доступности удаленных подсетей со шлюза из VRF

На шлюзе STG:Spoke1 (на другом шлюзе действуйте по аналогии с учетом специфики) запустите ICMP трафик до 192.168.100.1 с адреса 192.168.1.1 из vrf_blue и vrf_red:

```
root@Spoke1:~# ip vrf exec vrf_blue ping 192.168.100.1 -I 192.168.1.1 -c 4
PING 192.168.100.1 (192.168.100.1) from 192.168.1.1 : 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=1.81 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=1.82 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=1.98 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=1.63 ms

--- 192.168.100.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.627/1.809/1.977/0.123 ms
```

```
root@Spoke1:~# ip vrf exec vrf_red ping 192.168.100.1 -I 192.168.1.1 -c 4
PING 192.168.100.1 (192.168.100.1) from 192.168.1.1 : 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=1.27 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=1.54 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=1.69 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=1.52 ms

--- 192.168.100.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.273/1.506/1.688/0.149 ms
```

Убедитесь в наличии доступности удаленной подсети, если доступность отсутствует, проверьте настройки VRF, маршрутизации, IPsec с двух сторон.

Проверка IPsec

На шлюзе STG:Spoke1 (на другом шлюзе действуйте по аналогии с учетом специфики) выполните следующие команды для проверки работоспособности IPsec:

```
root@Spoke1:~# sa_mgr show
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 3 (172.16.1.2,500)-(172.16.100.2,500) active 28572 28456
2 4 (172.16.1.2,500)-(172.16.100.2,500) active 28760 28708

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 6 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 3036 3036
2 7 (172.16.1.2,*)-(172.16.100.2,*) 47 ESP tunn 12256 12372
```

```
root@Spoke1:~# sa_mgr show -detail
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connection id: 3
    cookies: FA7AFE1F9C21C259.40684F9EEA43294F
    local  peer (addr/port): (172.16.1.2/500)
    remote  peer (addr/port): (172.16.100.2/500)

    local identity (USER_FQDN): VRF-BLUE@spoke1
    remote identity (USER_FQDN): VRF-BLUE@hub1
    IKERule name: IKERule:GLOBAL_VRF_VPN:1
    auth: gost signature
    mode: main

    sa:
     transform: magma-cfb gostr341112s
     Oakley group: vko2
     sa limits: key lifetime (qm/k/sec): -/-/52857
```

```
     sa timing: remaining key lifetime (qm/k/sec): -/-/48710
     status: active

ISAKMP connection id: 4
    cookies: 0663F30C962693C4.0756F7928FFEB0DE
    local  peer (addr/port): (172.16.1.2/500)
    remote  peer (addr/port): (172.16.100.2/500)

    local identity (USER_FQDN): VRF-RED@spoke1
    remote identity (USER_FQDN): VRF-RED@hub1
    IKERule name: IKERule:VTI0_VRF_RED_VPN:1
    auth: gost signature
    mode: main

    sa:
     transform: magma-cfb gostr341112s
     Oakley group: vko2
     sa limits: key lifetime (qm/k/sec): -/-/56923
     sa timing: remaining key lifetime (qm/k/sec): -/-/52831
     status: active


IPsec connection id: 6
    local  ident (addr/prot/port): (0.0.0.0-255.255.255.255/*/*)
    remote  ident (addr/prot/port): (0.0.0.0-255.255.255.255/*/*)

    #pkts sent/rcvd: 50/50
    #send/recv errors: 0/0

    local crypto endpt.: 172.16.1.2, remote crypto endpt.: 172.16.100.2
    connection status: {initiated locally, }

    remote identity (USER_FQDN): VRF-RED@hub1
    IPsecAction name: IPsecAction:VTI0_VRF_RED_VPN:1
    Filter                                                                  LogEventID:
IPsec:Protect:VTI0_VRF_RED_VPN:1:VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1

    inbound esp:
     spi: 0x3475E30D(880141069)
      transform: esp-magma-mgm
      in use settings ={Tunnel, }
      sa limits: key lifetime (k/sec): -/2662
      sa timing: remaining key lifetime (k/sec): -/1167

    outbound esp:
     spi: 0x25A81403(631772163)
      transform: esp-magma-mgm
      in use settings ={Tunnel, }
      sa limits: key lifetime (k/sec): -/2662
      sa timing: remaining key lifetime (k/sec): -/1167

    created by: ISAKMP 4

IPsec connection id: 7
    local  ident (addr/prot/port): (172.16.1.2/47/*)
    remote  ident (addr/prot/port): (172.16.100.2/47/*)

    #pkts sent/rcvd: 133/134
    #send/recv errors: 0/0

    local crypto endpt.: 172.16.1.2, remote crypto endpt.: 172.16.100.2
    connection status: {initiated locally, }
```

```
    remote identity (USER_FQDN): VRF-BLUE@hub1
    IPsecAction name: IPsecAction:GLOBAL_VRF_VPN:1
    Filter                                                                  LogEventID:
IPsec:Protect:GLOBAL_VRF_VPN:1:GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1

    inbound esp:
     spi: 0x456F8BE2(1164938210)
      transform: esp-magma-mgm
      in use settings ={Tunnel, }
      sa limits: key lifetime (k/sec): -/2895
      sa timing: remaining key lifetime (k/sec): -/1578

    outbound esp:
     spi: 0xD9E664BD(3655754941)
      transform: esp-magma-mgm
      in use settings ={Tunnel, }
      sa limits: key lifetime (k/sec): -/2895
      sa timing: remaining key lifetime (k/sec): -/1578

    created by: ISAKMP 3
```

Уведитесь в том, что на шлюзе по два IKE и IPsec туннеля с правильными local identity и remote identity, если это не так проверьте настройки на обеих сторонах и откорректируйте при необходимости.

Примечание: Могут наблюдаться дублирующиеся IPsec туннели, это не является проблемой и вызвано встречным построением.

Особенности настройки

PBR в FRR и VRF

Правила PBR, заданные в FRR, для трафика более приоритетны (обрабатываются раньше) чем общее правило (`1000: from all lookup [l3mdev-table]`) для всех VRF.

```
root@Spoke1:~# vtysh
```

```
Hello, this is FRRouting (version 10.3.3).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
Spoke1# sh run
```

```
!
vrf vrf_blue
exit-vrf
!
vrf vrf_red
exit-vrf
!
interface eth2
 pbr-policy PBR
exit
!
pbr-map PBR seq 1
 match src-ip 192.168.1.0/24
 set nexthop 172.16.1.1
exit
!
pbr-map PBR seq 700
 match src-ip 192.168.2.0/24
 set nexthop 172.16.1.1
exit
```

```
!
end
```

```
root@Spoke1:~# ip rule
```

```
0:      from all lookup local
300:    from 192.168.1.0/24 iif eth2 lookup 10000 proto zebra
999:    from 192.168.2.0/24 iif eth2 lookup 10001 proto zebra
1000:   from all lookup [l3mdev-table]
32766:  from all lookup main
32767:  from all lookup default
```

Правила с номерами 300 и 999 заданы в консоли FRR (seq 1 и seq 700), правило с номером 1000 – это общее правило для всех VRF.

NAT в VRF

При настройке NAT на шлюзе при наличии VRF, где может присутствовать трафик из пересекающихся адресных пространств, требуется выполнить дополнительную настройку для обеспечения корректной работы NAT вввиду того, что таблица соединений (conntrack) по умолчанию общая и ничего не знает про VRF. Без этой настройки подсистема conntrack может ошибочно принимать сетевые соединения в разных VRF с полностью совпадающими параметрами (IP адреса, номера портов и др.) за одно соединение.

Настройка заключается в назначении уникального идентификатора зоны conntrack для трафика каждого VRF инстанса. Эти действия выполняются в таблице `raw`, до момента обработки трафика подсистемой conntrack.

Допустим, на шлюзе настроено два VRF инстанса – `vrf_blue` и `vrf_red`. В `vrf_blue` входят интерфейсы `eth1` и `st_vti1`, в `vrf_red` – `eth2` и `st_vti2`. Трафик VRF инстанса `vrf_blue` будет ассоциирован с зоной под номером 10, `vrf_red` – под номером 20. В таком случае настройка будет выглядеть следующим образом:

1. Для `vrf_blue`:

   1.1. Для входящего и транзитного трафика:

   ```
   root@sterragate:~# iptables -t raw -A PREROUTING -i eth1 -j CT --zone 10
   root@sterragate:~# iptables -t raw -A PREROUTING -i st_vti1 -j CT --zone 10
   ```

   1.2. Для локально сгенерированного трафика:

   ```
   root@sterragate:~# iptables -t raw -A OUTPUT -o eth1 -j CT --zone 10
   root@sterragate:~# iptables -t raw -A OUTPUT -o st_vti1 -j CT --zone 10
   ```

2. Для `vrf_red`:

   2.1. Для входящего и транзитного трафика:

   ```
   root@sterragate:~# iptables -t raw -A PREROUTING -i eth2 -j CT --zone 20
   root@sterragate:~# iptables -t raw -A PREROUTING -i st_vti2 -j CT --zone 20
   ```

   2.2. Для локально сгенерированного трафика:

   ```
   root@sterragate:~# iptables -t raw -A OUTPUT -o eth2 -j CT --zone 20
   root@sterragate:~# iptables -t raw -A OUTPUT -o st_vti2 -j CT --zone 20
   ```

3. Сохраните настройки:

   ```
   root@sterragate:~# netfilter-persistent save
   ```

Приложение

Конфигурация STG:Hub1

Консоль cisco-like

```
!
version 12.4
```

```
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 password 0 csp
aaa new-model
!
!
hostname Hub1
enable password csp
!
crypto pki certificate map FIND_LOCAL_CERT_BY_ISSUER_NAME 0
 issuer-name eq C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA
!
!
!
!
logging trap debugging
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile ISAKMP_PROFILE_VTI0_VRF_RED
 match identity email VRF-RED@spoke1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-RED@hub1
 no set identity mapping
 set policy 1
!
crypto isakmp profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 match identity email VRF-BLUE@spoke1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-BLUE@hub1
 no set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
ip access-list extended VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1
 permit ip any any
!
ip access-list extended GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1
 permit gre host 172.16.100.2 host 172.16.1.2
!
ipv6 default filtering deny
!
!
crypto map VTI0_VRF_RED_VPN 1 ipsec-isakmp
 description VTI over IPsec from VRF-red to Spoke1 (global VRF)
 match address VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
```

```
 set isakmp-profile ISAKMP_PROFILE_VTI0_VRF_RED
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map GLOBAL_VRF_VPN 1 ipsec-isakmp
 description GRE over IPsec from VRF-blue to Spoke1 (global VRF)
 match address GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 description WAN
 ip address 172.16.100.2 255.255.255.0
 crypto map GLOBAL_VRF_VPN
!
interface TenGigabitEthernet0/1
 no ip address
 shutdown
!
interface TenGigabitEthernet0/2
 no ip address
!
interface TenGigabitEthernet0/2.20
 description VRF-RED
 ip address 192.168.100.1 255.255.255.0
!
interface TenGigabitEthernet0/2.10
 description VRF-BLUE
 ip address 192.168.100.1 255.255.255.0
!
interface TenGigabitEthernet0/3
 no ip address
 shutdown
!
interface TenGigabitEthernet0/4
 no ip address
 shutdown
!
interface TenGigabitEthernet0/5
 no ip address
 shutdown
!
interface stVTI0
 description VRF-RED
 ip address 10.10.10.1 255.255.255.0
 crypto map VTI0_VRF_RED_VPN
!
interface Tunnel0
 description VRF-BLUE
 ip address 10.10.20.1 255.255.255.0
 mtu 1400
 tunnel key 1
 tunnel source 172.16.100.2
 tunnel destination 172.16.1.2
```

```
 offloading tx-checksumming
 tunnel ttl 64
 tunnel tos inherit
!
!
ip route 0.0.0.0 0.0.0.0 172.16.100.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check none
crypto pki certificate chain s-terra_technological_trustpoint
certificate DCC5E00574B641CDA1EA92E12C950190
308201E33082018EA003020102021100DCC5E00574B641CDA1EA92E12C950190
300C06082A850307010103020500305D310B300906035504061302525531133O
110603550407130A5A656C656E6F67726164311430120603550 40A130B532D54
65727261204353503 10C300A060355040B1303526E443115301306035504031 3
0C5465737420526F6F74204341301E170D323430323231313932363130 5A170D
333930323137313932363130 5A305D310B300906035504061302525531133011
0603550407130A5A656C656E6F67726164311430120603550 40A130B532D5465
72726120435350310C300A060355040B1303526E443115301306035504031 30C
5465737420526F6F742043413066301F06082A850307010101013013060 72A85
030202230106082A850307010102020343000440618 9FF08D1F9B6BFF4D18B4B
1454F16500AC2F012F644B5D054AAFFCB665FD30D4059066BB02F280D03B2EAD
64404930928612DBAE998A892C60D889F6521409A320301E300B0603551D0F04
0403020186300F0603551D130101FF040530030101FF300C06082A8503070101
030205000341000 35EA36F2A71887FB6ABA782AFA73B782F0D4506B09F49490C
D2CA89A8C2A447EBC9602DB9229520F8FC69B0D8BE8A7F3BEFF17B78388B623A
47F06874D0E717

quit
!
end
```

Сетевые интерфейсы

1. Файл /etc/network/interfaces

```
cat /etc/network/interfaces | tail -n 3
```

```
source /etc/network/interfaces.d/*
```

2. Файл /etc/network/interfaces.d/interfaces

```
auto st_vti0
iface st_vti0 inet static
address 10.10.10.1/24
pre-up ip link del $IFACE type dummy 2>/dev/null || true
pre-up ip link add $IFACE type dummy
post-down ip link set dev $IFACE down

auto vrf_blue
iface vrf_blue inet manual
pre-up ip link del $IFACE type vrf 2>/dev/null || true
pre-up ip route flush table 10 2>/dev/null || true
pre-up ip link add $IFACE type vrf table 10
pre-up ip link set dev $IFACE up
pre-up ip route add table 10 unreachable default metric 4278198272
pre-up ip link set dev gre_0 master $IFACE
pre-up ip link set dev eth2_10 master $IFACE
post-down ip link set dev $IFACE down

auto vrf_red
iface vrf_red inet manual
pre-up ip link del $IFACE type vrf 2>/dev/null || true
```

```
pre-up ip route flush table 20 2>/dev/null || true
pre-up ip link add $IFACE type vrf table 20
pre-up ip link set dev $IFACE up
pre-up ip route add table 20 unreachable default metric 4278198272
pre-up ip link set dev st_vti0 master $IFACE
pre-up ip link set dev eth2_20 master $IFACE
post-down ip link set dev $IFACE down
```

Динамическая маршрутизация (FRR)

```
frr version 10.3.3
frr defaults traditional
hostname Hub1
log syslog informational
service integrated-vtysh-config
!
vrf vrf_blue
exit-vrf
!
vrf vrf_red
exit-vrf
!
interface gre_0
 ip ospf network point-to-point
 no ip ospf passive
exit
!
router bgp 65000 vrf vrf_red
 bgp router-id 10.10.10.1
 neighbor 10.10.10.2 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.100.0/24
  neighbor 10.10.10.2 default-originate
 exit-address-family
exit
!
router ospf vrf vrf_blue
 ospf router-id 10.10.20.1
 passive-interface default
 network 10.10.20.0/24 area 0
 network 192.168.100.0/24 area 0
 default-information originate
exit
!
end
```

Конфигурация STG:Spoke1

Консоль cisco-like

```
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
```

```
crypto isakmp keepalive retry-count 5
username cscons privilege 15 password 0 csp
aaa new-model
!
!
hostname Spoke1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
crypto pki certificate map FIND_LOCAL_CERT_BY_ISSUER_NAME 0
 issuer-name eq C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA
!
!
!
!
logging trap debugging
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile ISAKMP_PROFILE_VTI0_VRF_RED
 match identity email VRF-RED@hub1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-RED@spoke1
 no set identity mapping
 set policy 1
!
crypto isakmp profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 match identity email VRF-BLUE@hub1
 self-certificate FIND_LOCAL_CERT_BY_ISSUER_NAME
 self-identity email VRF-BLUE@spoke1
 no set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
ip access-list extended VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1
 permit ip any any
!
ip access-list extended GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1
 permit gre host 172.16.1.2 host 172.16.100.2
!
ipv6 default filtering deny
!
!
crypto map VTI0_VRF_RED_VPN 1 ipsec-isakmp
 description VTI over IPsec from VRF-red to Spoke1 (global VRF)
 match address VTI0_VRF_RED_IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile ISAKMP_PROFILE_VTI0_VRF_RED
 set security-association mss-fix out
 set local-address 172.16.1.2
 set peer 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map GLOBAL_VRF_VPN 1 ipsec-isakmp
 description GRE over IPsec from VRF-blue to Spoke1 (global VRF)
 match address GRE0_VRF_BLUE_IPSEC_ACL_HUB1_AND_SPOKE1
```

```
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile ISAKMP_PROFILE_GRE0_VRF_BLUE
 set security-association mss-fix out
 set local-address 172.16.1.2
 set peer 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 description WAN
 ip address 172.16.1.2 255.255.255.0
 crypto map GLOBAL_VRF_VPN
!
interface TenGigabitEthernet0/1
 no ip address
 shutdown
!
interface TenGigabitEthernet0/2
 no ip address
!
interface TenGigabitEthernet0/2.10
 description VRF-BLUE
 ip address 192.168.1.1 255.255.255.0
!
interface TenGigabitEthernet0/2.20
 description VRF-RED
 ip address 192.168.1.1 255.255.255.0
!
interface TenGigabitEthernet0/3
 no ip address
 shutdown
!
interface TenGigabitEthernet0/4
 no ip address
!
interface TenGigabitEthernet0/5
 no ip address
 shutdown
!
interface stVTI0
 description VRF-RED
 ip address 10.10.10.2 255.255.255.0
 crypto map VTI0_VRF_RED_VPN
!
interface Tunnel0
 description VRF-BLUE
 ip address 10.10.20.2 255.255.255.0
 mtu 1400
 tunnel key 1
 tunnel source 172.16.1.2
 tunnel destination 172.16.100.2
 offloading tx-checksumming
 tunnel ttl 64
 tunnel tos inherit
!
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check none
crypto pki certificate chain s-terra_technological_trustpoint
```

```
certificate DCC5E00574B641CDA1EA92E12C950190
308201E33082018EA003020102021100DCC5E00574B641CDA1EA92E12C950190
300C06082A850307010103020500305D310B30090603550406130253553113300
110603550407130A5A656C656E6F6772616431143012060355040A130B532D54
6572726120435350310C300A060355040B1303526E443115301306035504031 3
0C5465737420526F6F74204341301E170D323430323231313932363130 5A170D
333930323137313932363130 5A305D310B30090603550406130252553113301 1
0603550407130A5A656C656E6F6772616431143012060355040A130B532D5465
7272612043535031 0C300A060355040B1303526E4431153013060355040313 C
5465737420526F6F742043413066301F06082A850307010101013013060 72A85
030202230106082A850307010102020343000440618 9FF08D1F9B6BFF4D18B4B
1454F16500AC2F012F644B5D054AAFFCB665FD30D4059066BB02F280D03B2EAD
64404930928612DBAE998A892C60D889F6521409A320301E300B0603551D0F04
0403020186300F0603551D130101FF040530030101FF300C06082A8503070101
030205000341000 35EA36F2A71887FB6ABA782AFA73B782F0D4506B09F49490C
D2CA89A8C2A447EBC9602DB9229520F8FC69B0D8BE8A7F3BEFF17B78388B623A
47F06874D0E717

quit
!
end
```

Сетевые интерфейсы

1. Файл /etc/network/interfaces

```
cat /etc/network/interfaces | tail -n 3
```

```
source /etc/network/interfaces.d/*
```

2. Файл /etc/network/interfaces.d/interfaces

```
auto st_vti0
iface st_vti0 inet static
address 10.10.10.2/24
pre-up ip link del $IFACE type dummy 2>/dev/null || true
pre-up ip link add $IFACE type dummy
post-down ip link set dev $IFACE down

auto vrf_blue
iface vrf_blue inet manual
pre-up ip link del $IFACE type vrf 2>/dev/null || true
pre-up ip route flush table 10 2>/dev/null || true
pre-up ip link add $IFACE type vrf table 10
pre-up ip link set dev $IFACE up
pre-up ip route add table 10 unreachable default metric 4278198272
pre-up ip link set dev gre_0 master $IFACE
pre-up ip link set dev eth2_10 master $IFACE
post-down ip link set dev $IFACE down

auto vrf_red
iface vrf_red inet manual
pre-up ip link del $IFACE type vrf 2>/dev/null || true
pre-up ip route flush table 20 2>/dev/null || true
pre-up ip link add $IFACE type vrf table 20
pre-up ip link set dev $IFACE up
pre-up ip route add table 20 unreachable default metric 4278198272
pre-up ip link set dev st_vti0 master $IFACE
pre-up ip link set dev eth2_20 master $IFACE
post-down ip link set dev $IFACE down
```

Динамическая маршрутизация (FRR)

```
frr version 10.3.3
```

```
frr defaults traditional
hostname Spoke1
log stdout informational
log syslog informational
service integrated-vtysh-config
!
vrf vrf_blue
exit-vrf
!
vrf vrf_red
exit-vrf
!
interface gre_0
 ip ospf network point-to-point
 no ip ospf passive
exit
!
router bgp 65000 vrf vrf_red
 bgp router-id 10.10.10.2
 neighbor 10.10.10.1 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.1.0/24
 exit-address-family
exit
!
router ospf vrf vrf_blue
 ospf router-id 10.10.20.2
 passive-interface default
 network 10.10.20.0/24 area 0
 network 192.168.1.0/24 area 0
exit
!
end
```