Отказоустойчивое решение на базе GRE-туннелей при наличии двух провайдеров

Настоящий документ содержит описание способа совместного использования Продуктов компании ООО «С-Терра СиЭсПи» и Продуктов третьих производителей.
ООО «С-Терра СиЭсПи» осуществляет сопровождение настоящего сценария в части настроек Продуктов Компании. Упоминание наименований, продуктов, торговых марок третьих организаций исключительно неформально и не является поддержкой, рекомендацией либо рекламой. ООО «С-Терра СиЭсПи» не несет какой-либо ответственности в отношении работоспособности и использования этих Продуктов. Документ имеет статус вспомогательного материала, который может быть использован технологическими партнерами, компаниями-интеграторами, при разработке собственных решений.
Решения, разработанные на базе данного сценария, могут применяться в действующих сетях/системах только после тестовой и/или опытной эксплуатации.

Введение

Данный сценарий содержит пример настойки «С-Терра Шлюз» для обеспечения отказоустойчивости защищенных каналов связи при помощи GRE-туннелей и протокола динамической маршрутизации BGP.

В данном сценарии между криптошлюзом центрального офиса и криптошлюзами филиалов строятся по два GRE-туннеля через разных провайдеров (Hub1-ISP1 и Hub1-ISP2). По GRE-туннелям происходит обмен маршрутами до защищаемых подсетей с помощью протокола BGP. Пользовательский трафик инкапсулируется в GRE и шифруется.

Партнеры по IPsec, в настоящем документе – криптошлюзы, для аутентификации будут использовать сертификаты в формате x509. В качестве криптопровайдера будет использована криптографическая библиотека, разработанная компанией «С-Терра СиЭсПи». Шлюзы безопасности (или криптошлюзы) – «С-Терра Шлюз» версии 5.0.

На криптошлюзах будет настроено межсетевое экранирование с применением статических списков доступа (ACL) и пакетной фильтрации с контролем состояния сессий (stateful, аналог CBAC в Cisco IOS) для протоколов TCP и UDP.

Предварительные требования

Требования к материально-техническому обеспечению

Для переноса запросов и сертификатов между криптошлюзами и центром выпуска сертификатов требуется USB Flash накопитель.

Требования к квалификации администратора

Администратор должен обладать обширными знаниями в области сетевой информационной безопасности, иметь опыт работы с аналогичным оборудованием/программным обеспечением, знать и понимать следующие технологии и протоколы: PKI, IPsec, NAT, Firewall, routing, GRE, BGP, switching.

Администратор должен быть ознакомлен со сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)». Далее по тексту он будет называться базовым сценарием.

Требования к инфраструктуре

1. Требования к устройствам.
   1.1. «С-Терра Шлюз».

1.1.1 Устройства «С-Терра Шлюз» должны быть инициализированы (подробнее на http://doc.s-terra.ru раздел С-Терра Шлюз -> С-Терра Шлюз <версия> -> «Подключение ПАК и инициализация С-Терра Шлюз на вычислительных системах архитектуры Intel x86-64»).

> В «С-Терра Шлюз» версии 5.0 появилась возможность пройти инициализацию через интерфейс управления (management) по SSH, данная процедура и ее ограничения описаны в документации и на портале https://doc.s-terra.ru.

1.2. Центр выпуска сертификатов.

1.2.1 Должен быть настроен центр выпуска сертификатов (удостоверяющий центр, далее УЦ) для IPsec. Устройство с именем Certification_authority на схеме.

1.2.2 Для выпуска цифровых сертификатов допускается использование встроенного в ОС Windows Server 2016 (или новее) удостоверяющего центра совместно с сертифицированным СКЗИ «КриптоПро» CSP 5.0 (или новее).

1.2.3 Для тестовых целей можно использовать тестовый УЦ от «КриптоПро» (веб-интерфейс: https://testgost2012.cryptopro.ru/certsrv/certrqxt.asp).

> Категорически запрещено использование тестового УЦ от «КриптоПро» в производственной (боевой) эксплуатации, так как в данном случае отсутствует возможность контролировать процесс выпуска сертификатов и, соответственно, процедуру аутентификации.

1.3. HTTP сервер для распространения списка отозванных сертификатов.

1.3.1 Функционирующий HTTP сервер для распространения списка отозванных сертификатов (COC). Устройство с именем CRL_distribution_point на схеме. Если по объективным причинам использование COC не представляется возможным или не требуется, то проверку COC можно отключить в CLI «С-Терра Шлюз» (при соблюдении правил пользования).

> Доставка нового списка отозванных сертификатов с удостоверяющего центра на HTTP сервер должна происходить заблаговременно, до истечения срока действия предыдущего списка.

2. Требования к сетевому взаимодействию.

2.1. Между устройствами стенда должна быть обеспечена IP связность.

Схема взаимодействия

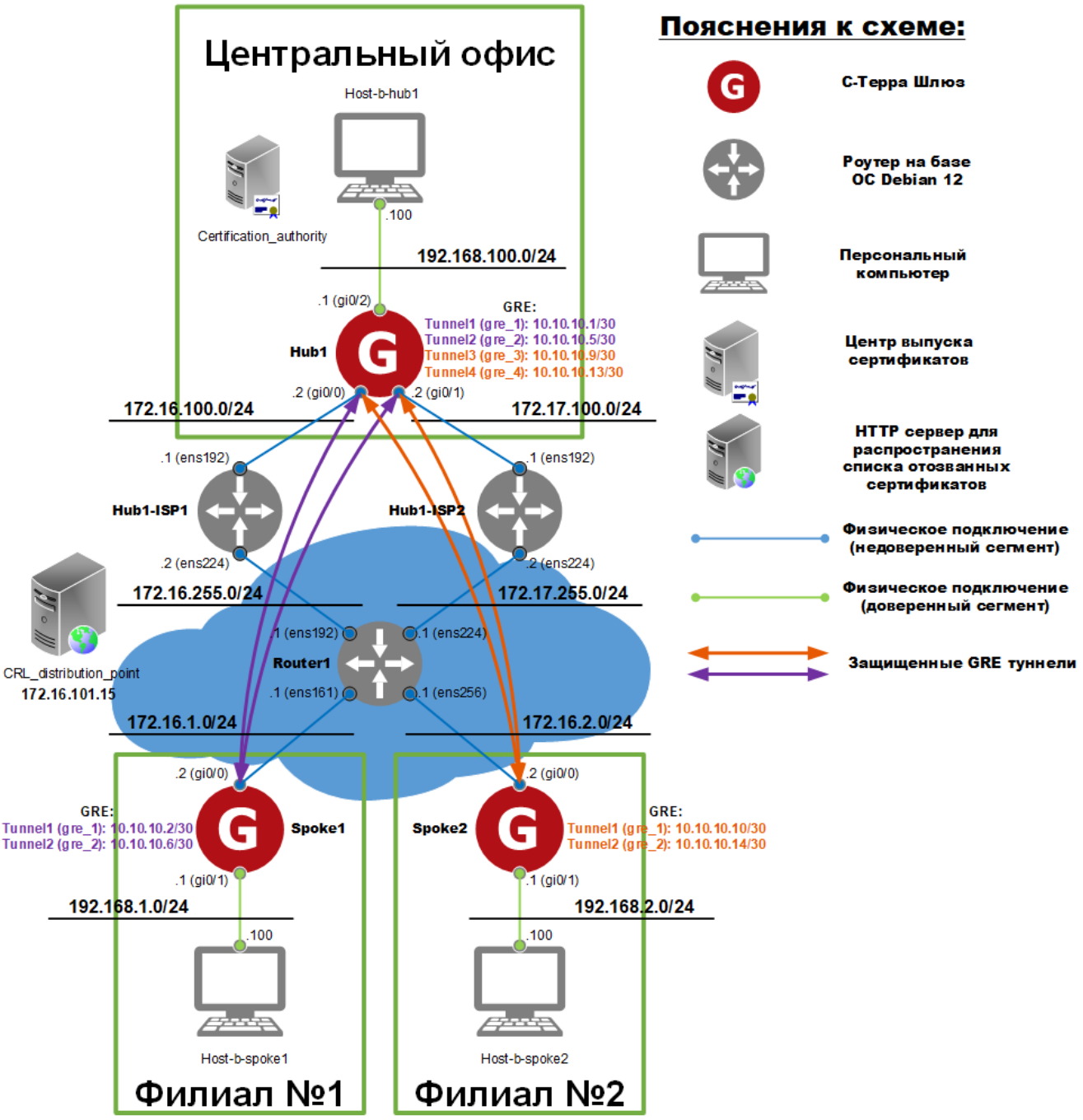


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), криптошлюз «С-Терра Шлюз» (Hub1) и персональный компьютер (Host-b-hub1).

   1.2. В филиале №1 размещаются: криптошлюз «С-Терра Шлюз» (Spoke1) и персональный компьютер (Host-b-spoke1).

   1.3. В филиале №2 размещаются: криптошлюз «С-Терра Шлюз» (Spoke2) и персональный компьютер (Host-b-spoke2).

1.4. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point), маршрутизаторы (Router1, Hub1-ISP1 и Hub1-ISP2).

2. Подключение к сети Интернет.
   В данном сценарии для эмуляции сети Интернет используются маршрутизаторы Router1, Hub1-ISP1 и Hub1-ISP2.

   Подключение к сети Интернет на устройствах «С-Терра Шлюз» будет считаться успешным, если по протоколу ICMP, например, при помощи утилиты «ping», будет доступен HTTP сервер для распространения списка отозванных сертификатов (устройство CRL_distribution_point на схеме).

   2.1. Криптошлюз Hub1 получает доступ к сети Интернет с помощью маршрутизаторов Hub1-ISP1 или Hub1-ISP2. Выбор маршрутизатора для отправки пакетов осуществляется с помощью сервиса multiwan, который будет добавлять маршруты по умолчанию через маршрутизаторы Hub1-ISP1 или Hub1-ISP2. Для трафика от внешних интерфейсов Hub1 (172.16.100.2 и 172.17.100.2) до Spoke1 (172.16.1.2) и Spoke2 (172.16.2.2) используется механизм PBR (Policy-based routing). С помощью PBR для внешних интерфейсов создаются правила и своя таблица маршрутизации, которая содержит маршруты до Spoke1 и Spoke2.

   2.2. Криптошлюз Spoke1 подключается к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router1).

   2.3. Криптошлюз Spoke2 подключается к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router1).

3. GRE-туннелирование.

   С целью балансировки трафика и обеспечения отказоустойчивости с каждого внешнего интерфейса Hub1 (172.16.100.2/24 и 172.17.100.2/24) строится по два GRE туннеля – до криптошлюза Spoke1 (172.16.1.2/24) и до криптошлюза Spoke2 (172.16.2.2/24). Внутри GRE-туннелей пропускается защищаемый трафик, а также пакеты динамической маршрутизации BGP.

4. Динамическая маршрутизация.

   Обмен маршрутами до защищаемых подсетей между криптошлюзами центрального офиса и филиалов происходит с помощью протокола динамической маршрутизации BGP. Для того, чтобы обеспечить одновременную работу через двух провайдеров, необходимо создать две таблицы маршрутизации. В одной таблице будут заданы маршруты до адресов WAN интерфейсов криптошлюзов Spoke1 и Spoke2 через провайдера Hub1-ISP1. Во второй таблице будут заданы маршруты до адресов WAN интерфейсов криптошлюзов Spoke1 и Spoke2 через провайдера Hub1-ISP2. Реализуется данная настройка при помощи технологии PBR, которая позволяет создать две таблицы маршрутизации и внести туда маршруты до Spoke1 и Spoke2. Благодаря этому, протокол BGP дополнит основную (main) таблицу маршрутизации маршрутами с одинаковыми метриками до подсетей за криптошлюзами Spoke1 и Spoke2, вследствие чего, автоматически начнёт свою работу механизм ECMP (Equal-Cost Multi-Path routing), осуществляющий балансировку трафика. По умолчанию, балансировка осуществляется на основании IP адреса получателя и отправителя, при этом возможно возникновение асимметрии, когда трафик в направлениях туда и обратно будет идти по разным GRE-туннелям. В случае сбоя одного из каналов связи обмен маршрутами по GRE-туннелю прекратится и в таблице маршрутизации останется только один маршрут. Как следствие, весь трафик будет направлен через рабочий канал связи. В сценарии, для обеспечения обменов маршрутными правилами между устройствами по протоколу BGP, использовано свободно распространяемое программное средство FRRouting. В силу особенностей GRE туннелирования, криптошлюзы не могут размещаться за динамическим NAT.

5. Параметры межсетевого экранирования.

   Для пакетов, которые не защищаются IPsec применяются следующие правила фильтрации:

   5.1. На LAN интерфейсах IPv4 трафик не контролируется.

   5.2. На LAN интерфейсах IPv6 трафик заблокирован.

5.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

5.3.1 Протокол UDP порты 500 и 4500 – для IKE и NAT-Traversal (ESP in UDP).

5.3.2 Протокол ICMP – для проверки доступности криптошлюза и работы механизма PathMTUd.

5.3.3 Протокол ESP (номер 50) – для работы IPsec.

5.3.4 Весь TCP и UDP трафик, для которого есть исходящее соединение (это динамические правила, которые создаются при помощи stateful фильтрации).

5.4. На WAN интерфейсах заблокирован исходящий незашифрованный GRE трафик и исходящий незашифрованный трафик между защищаемыми подсетями.

5.5. На WAN интерфейсах IPv6 трафик заблокирован.

5.6. На WAN интерфейсах инспектируются все исходящие локальные и транзитные IPv4 соединения для протоколов TCP и UDP. На основе результатов инспектирования осуществляется фильтрация входящего трафика с учетом состояния сессии (stateful).

6. Параметры шифрования IPsec.

Весь GRE трафик между криптошлюзами центрального офиса (Hub1) и филиалов (Spoke1 и Spoke2) защищается с использованием алгоритмов ГОСТ и протокола IPsec в туннельном режиме.

Инициировать защищенное соединение может как GRE трафик от шлюза центрального офиса (172.16.100.2/24 или 172.17.100.2/24) к шлюзам филиалов (172.16.1.2/24 или 172.16.2.2/24), так и наоборот. Пользовательский трафик до защищаемой подсети должен инкапсулироваться в GRE и шифроваться.

6.1. Параметры протокола IKE:

- Аутентификация при помощи цифровых сертификатов, алгоритм подписи – ГОСТ Р 34.10-2012 (ключ 256 бит);
- Алгоритм шифрования – ГОСТ Р 34.12-2015 Магма (ключ 256 бит);
- Алгоритм вычисления хеш-функции – ГОСТ Р 34.11-2012 ТК26 Стрибог (ключ 256 бит);
- Алгоритм выработки общего ключа (аналог алгоритма Диффи-Хеллмана) – VKO_GOSTR3410_2012_256 (ключ 256 бит).

6.2. Параметры протокола ESP:

- Комбинированный алгоритм шифрования и имитозащиты (контроль целостности) – ГОСТ Р 34.12-2015 Магма в режиме MGM (ключ 256 бит).

Настройка стенда

Настройка устройства Hub1-ISP1

1. Настройте IP адрес – 172.16.100.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.16.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Задайте маршрут по умолчанию через 172.16.255.1.
4. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Hub1-ISP2

1. Настройте IP адрес – 172.17.100.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.17.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Задайте маршрут по умолчанию через 172.17.255.1.
4. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Router1

1. Настройте IP адрес – 172.16.255.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.17.255.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Настройте IP адрес – 172.16.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens256.
4. Настройте IP адрес – 172.16.2.1 и маску – 255.255.255.0 на сетевом интерфейсе ens161.
5. Маршрут до сети 172.16.100.0 0.0.0.255 через 172.16.255.2.
6. Маршрут до сети 172.17.100.0 0.0.0.255 через 172.17.255.2.
7. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Host-b-hub1

1. Настройте IP адрес – 192.168.100.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.100.1.
3. Разрешите прием и отправку ICMP пакетов.

Настройка устройства Host-b-spoke1

1. Настройте IP адрес – 192.168.1.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.1.1.
3. Разрешите прием и отправку ICMP пакетов.

Настройка устройства Host-b-spoke2

1. Настройте IP адрес – 192.168.2.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.2.1.
3. Разрешите прием и отправку ICMP пакетов.

Настройка криптошлюза Hub1

Настройка будет происходить локально при помощи консольного подключения.

Настройка может осуществляться и удаленно (по SSH), но исключительно по доверенному каналу связи. Доверенным каналом связи может считаться канал в пределах контролируемой зоны в случае отсутствия в нем нарушителя (в нашем примере это подсети 192.168.100.0/24, 192.168.1.0/24, 192.168.2.0/24. Доверенным каналом связи также считается канал, защищенный при помощи протокола IPsec.

Начальные настройки

Дата и время на всех криптошлюзах и УЦ должны быть одинаковы, так как для аутентификации используются цифровые сертификаты, в которых зафиксированы дата и время начала их действия и окончания. Также одинаковые дата и время на всей инфраструктуре облегчают поиск неисправностей по лог-файлам.

Войдите в CLI разграничения доступа (пользователь и пароль по умолчанию: `administrator`, `s-terra`) установите правильный тип терминала и требуемую ширину, смените пароль для пользователя `administrator`.

Перейдите в linux bash консоль, установите пароль для пользователя `root`, правильное время и временную зону.

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройки PKI (запросы и сертификаты)

Продукт не поддерживает разностные (delta) списки отзыва сертификатов, только базовые. Учитывайте это при выборе или развертывании УЦ.

Для аутентификации партнеров по IPsec можно использовать цифровые сертификаты, выпущенные при помощи сертифицированного СКЗИ, или предварительно распределенные ключи, созданные и введённые в соответствии с правилами пользования СКЗИ (появились в версии 5.0).

Сгенерируйте запрос на сертификат криптошлюза и сохраните этот запрос в файл на USB Flash накопителе. Доставьте файл с запросом на УЦ и выпустите по нему сертификат. Сертификат криптошлюза вместе с сертификатом УЦ доставьте на шлюз и импортируйте их в базу Продукта.

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка паролей доступа к консоли cisco-like

Войдите в cisco-like консоль и смените пароль для пользователя `cscons` (пароль по умолчанию: csp) и пароль на enable (пароль по умолчанию: csp).

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка сетевых параметров

Изменение состояния интерфейсов и назначение IP адресов применяются сразу после ввода соответствующих команд. Политика безопасности применяется после выхода из режима конфигурирования.

1. Войдите в cisco-like консоль и задайте имя устройства:

```
sterragate#configure terminal
Enter configuration commands, one per line. End with CNTL/Z
sterragate(config)#hostname Hub1
Hub1(config)#
```

2. Включите внешние `GigabitEthernet0/0` и `GigabitEthernet0/1` и внутренний `GigabitEthernet0/2` интерфейсы:

```
Hub1(config)#interface range GigabitEthernet 0/0 - 2
Hub1(config-if-range)#no shutdown
Hub1(config-if-range)#exit
```

3. Убедитесь в том, что интерфейсы административно включены и line protocol (способность интерфейса передавать пакеты в данный момент) находится в состоянии up:

```
Hub1(config)#do show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware address is 0050.569e.85be
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.865b
  MTU 1500 bytes
GigabitEthernet0/2 is up, line protocol is up
  Hardware address is 0050.569e.6568
  MTU 1500 bytes
...
```

Если line protocol находится в состоянии down, то проверьте подключение сетевого интерфейса криптошлюза к коммутационному или прочему оборудованию.

4. Задайте IP адреса в соответствии со схемой стенда на внешних интерфейсах `GigabitEthernet0/0` и `GigabitEthernet0/1`, и внутреннем `GigabitEthernet0/2` интерфейсах:

```
Hub1(config)#interface GigabitEthernet 0/0
Hub1(config-if)#ip address 172.16.100.2 255.255.255.0
Hub1(config-if)# description ISP1
Hub1(config-if)#exit
Hub1(config)#interface GigabitEthernet 0/1
Hub1(config-if)#ip address 172.17.100.2 255.255.255.0
Hub1(config-if)# description ISP2
Hub1(config-if)#exit
Hub1(config)#interface GigabitEthernet 0/2
Hub1(config-if)#ip address 192.168.100.1 255.255.255.0
Hub1(config-if)#exit
```

5. Проверьте доступность устройства Hub1-ISP1:

```
Hub1(config)#do ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 100(128) bytes of data.
108 bytes from 172.16.100.1: icmp_seq=1 ttl=64 time=0.484 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.393 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.282 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.496 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.496 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4053ms
rtt min/avg/max/mdev = 0.282/0.430/0.496/0.083 ms
```

6. Проверьте доступность устройства Hub1-ISP2:

```
Hub1(config)#do ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 100(128) bytes of data.
108 bytes from 172.17.100.1: icmp_seq=1 ttl=64 time=0.276 ms
108 bytes from 172.17.100.1: icmp_seq=2 ttl=64 time=0.307 ms
108 bytes from 172.17.100.1: icmp_seq=3 ttl=64 time=0.435 ms
108 bytes from 172.17.100.1: icmp_seq=4 ttl=64 time=0.317 ms
108 bytes from 172.17.100.1: icmp_seq=5 ttl=64 time=0.495 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4063ms
rtt min/avg/max/mdev = 0.276/0.366/0.495/0.084 ms
```

7. Примените настройки:

```
Hub1(config)#end
```

Настройка multiwan

В данном сценарии для доступа к сети Интернет со шлюза Hub1 используются маршрутизаторы Hub1-ISP1 и Hub1-ISP2. Для выбора между ними и установки в систему маршрута по умолчанию используется пакет `multiwan`.

Настройка multiwan выполняется из linux bash консоли криптошлюза.

1. Приведите содержимое конфигурационного файла `/etc/multiwan/config.yaml` к следующему виду:

```
config_format_version: 1
links_params:
  check_links_reachability_on_start: true
  links:
    -
      iface_name: eth0
      name: ISP1
```

```
      prio: 200
      routes:
        ipv4:
          networks:
            - 0.0.0.0/0
          nexthop: 172.16.100.1
      check:
        hosts_reachability: any
        ipv4:
          hosts:
            - 172.16.101.15
        method:
          icmp:
            interval: 1
            restore_interval: 1
            restore_count: 3
            failure_count: 3

    -
      iface_name: eth1
      name: ISP2
      prio: 100
      routes:
        ipv4:
          networks:
            - 0.0.0.0/0
          nexthop: 172.17.100.1
      check:
        hosts_reachability: any
        ipv4:
          hosts:
            - 172.16.101.15
        method:
          icmp:
            interval: 1
            restore_interval: 1
            restore_count: 3
            failure_count: 3
```

2. Запустите сервис multiwan и добавьте его в автозапуск:

```
root@Hub1:~# systemctl start multiwan.service
root@Hub1:~# systemctl enable multiwan.service
```

В Приложении представлено содержимое конфигурационного файла `/etc/multiwan/config.yaml`.

Подробное описание настройки multiwan приведено в сценарии «Резервирование каналов связи при помощи пакета «multiwan».

Настройка GRE интерфейсов

1. В cisco-like консоли перейдите в конфигурационный режим:

```
Hub1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1(config)#
```

2. Настройте интерфейс GRE туннеля между Hub1 и Spoke1 через провайдера Hub1-ISP1:

   2.1. Создайте интерфейс `Tunnel 1`:

```
Hub1(config)#interface Tunnel 1
```

   2.2. Добавьте описание для данного интерфейса (опционально):

```
Hub1(config-if)#description GRE between Hub1 and Spoke1 via ISP1
```

2.3. Задайте IP адрес и маску на интерфейсе:

```
Hub1(config-if)#ip address 10.10.10.1 255.255.255.252
```

2.4. Укажите MTU для интерфейса:

```
Hub1(config-if)#mtu 1400
```

2.5. Задайте source IP адрес GRE туннеля:

```
Hub1(config-if)# tunnel source 172.16.100.2
```

2.6. Задайте destination IP адрес GRE туннеля:

```
Hub1(config-if)# tunnel destination 172.16.1.2
```

2.7. Отключите механизм Path MTU Discovery:

```
Hub1(config-if)#no tunnel path-mtu-discovery
```

2.8. Отключите копирование DF бита:

```
Hub1(config-if)#tunnel ignore-df
```

2.9. Выйдите из режима конфигурирования GRE интерфейса:

```
Hub1(config-if)#exit
```

3. Настройте интерфейс GRE туннеля между Hub1 и Spoke1 через провайдера Hub1-ISP2:

```
Hub1(config)#interface Tunnel 2
Hub1(config-if)#description GRE between Hub1 and Spoke1 via ISP2
Hub1(config-if)#ip address 10.10.10.5 255.255.255.252
Hub1(config-if)#mtu 1400
Hub1(config-if)#tunnel source 172.17.100.2
Hub1(config-if)#tunnel destination 172.16.1.2
Hub1(config-if)#no tunnel path-mtu-discovery
Hub1(config-if)#tunnel ignore-df
Hub1(config-if)#exit
```

4. Настройте интерфейс GRE туннеля между Hub1 и Spoke2 через провайдера Hub1-ISP1:

```
Hub1(config)#interface Tunnel 3
Hub1(config-if)#description GRE between Hub1 and Spoke2 via ISP1
Hub1(config-if)#ip address 10.10.10.9 255.255.255.252
Hub1(config-if)#mtu 1400
Hub1(config-if)#tunnel source 172.16.100.2
Hub1(config-if)#tunnel destination 172.16.2.2
Hub1(config-if)#no tunnel path-mtu-discovery
Hub1(config-if)#tunnel ignore-df
Hub1(config-if)#exit
```

5. Настройте интерфейс GRE туннеля между Hub1 и Spoke2 через провайдера Hub1-ISP2:

```
Hub1(config)#interface Tunnel 4
Hub1(config-if)#description GRE between Hub1 and Spoke2 via ISP2
Hub1(config-if)#ip address 10.10.10.13 255.255.255.252
Hub1(config-if)#mtu 1400
Hub1(config-if)#tunnel source 172.17.100.2
Hub1(config-if)#tunnel destination 172.16.2.2
Hub1(config-if)#no tunnel path-mtu-discovery
Hub1(config-if)#tunnel ignore-df
Hub1(config-if)#exit
```

Настройка шифрования

1. Параметры IKE.

1.1. Задайте общие параметры IKE:

```
Hub1(config)#crypto isakmp identity dn
Hub1(config)#crypto isakmp keepalive 3 2
Hub1(config)#crypto isakmp keepalive retry-count 5
```

```
Hub1(config)#crypto isakmp fragmentation
Hub1(config)#crypto isakmp security-association lifetime delta 50
Hub1(config)#crypto isakmp initiator-sessions-max 100
Hub1(config)#crypto isakmp responder-sessions-max 100
```

1.2. Создайте политику, описывающую параметры IKE туннеля:

```
Hub1(config)#crypto isakmp policy 1
Hub1(config-isakmp)#encryption magma
Hub1(config-isakmp)#hash stribog-256
Hub1(config-isakmp)#authentication gost-sig
Hub1(config-isakmp)#group vko2
Hub1(config-isakmp)#exit
```

1.3. Создайте профиль настроек ISAKMP (алгоритм IKE) под именем `NEW_ALGS`, чтобы далее его прикрепить к криптокарте:

```
Hub1(config)#crypto isakmp profile NEW_ALGS
Hub1(conf-isa-prof)#match identity dn
Hub1(conf-isa-prof)#set identity mapping
Hub1(conf-isa-prof)#crypto isakmp policy 1
```

2. Параметры IPsec.

2.1. Отключите обновление ключа IPsec SA по достижению определенного объема переданного трафика:

```
Hub1(config)#crypto ipsec security-association lifetime kilobytes unlimited
```

2.2. Задайте комбинированный алгоритм шифрования и имитозащиты (набор преобразований) для трафика:

```
Hub1(config)#crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
Hub1(cfg-crypto-trans)#exit
```

2.3. Создайте списки доступа (ACL):

2.3.1 Создайте два списка доступа для GRE трафика, который нужно будет защищать между центральным офисом и филиалом №1:

```
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE1
Hub1(config-ext-nacl)#permit gre host 172.16.100.2 host 172.16.1.2
Hub1(config-ext-nacl)#exit
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE2
Hub1(config-ext-nacl)#permit gre host 172.17.100.2 host 172.16.1.2
Hub1(config-ext-nacl)#exit
```

2.3.2 Создайте два списка доступа для GRE трафика, который нужно будет защищать между центральным офисом и филиалом №2:

```
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE2_GRE1
Hub1(config-ext-nacl)#permit gre host 172.16.100.2 host 172.16.2.2
Hub1(config-ext-nacl)#exit
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE2_GRE2
Hub1(config-ext-nacl)#permit gre host 172.17.100.2 host 172.16.2.2
Hub1(config-ext-nacl)#exit
```

2.4. Создайте криптокарту для построения туннелей через провайдера Hub1-ISP1:

2.4.1 Создайте первый раздел криптокарты (имя VPN_ISP1, раздел 1):

```
Hub1(config)#crypto map VPN_ISP1 1 ipsec-isakmp
```

2.4.2 Добавьте описание для данного раздела криптокарты (опционально):

```
Hub1(config-crypto-map)#description VPN between HUB1 and Spoke1 via ISP1
```

2.4.3 Укажите список доступа для защищаемого трафика:

```
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE1_GRE1
```

2.4.4 Укажите при помощи какого набора алгоритмов нужно защищать трафик:

```
Hub1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
```

2.4.5 Укажите локальный IP адрес, с которого будет отправляться шифрованные пакеты:

```
Hub1(config-crypto-map)#set local-address 172.16.100.2
```

2.4.6 Укажите IP адрес партнера по IPsec (в данном сценарии это внешний IP адрес устройства Spoke1):

```
Hub1(config-crypto-map)#set peer 172.16.1.2
```

2.4.7 Прикрепите ISAKMP профиль `NEW_ALGS` к криптокарте:

```
Hub1(config-crypto-map)#set isakmp-profile NEW_ALGS
```

2.4.8 Включите префрагментацию (сначала трафик фрагментируется, потом зашифровывается), чтобы в недоверенной сети не было фрагментов (некоторые интернет-провайдеры блокируют фрагменты) и чтобы снизить вероятность ложного удаления фрагментированных пакетов при интенсивном трафике:

```
Hub1(config-crypto-map)#set pre-fragmentation
```

2.4.9 Обязательно отключите историю удаленных туннелей (если не отключить, то могут быть проблемы с построением IPsec туннелей с устройствами, которые находятся за NAT):

```
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
```

2.4.10 Создайте второй раздел криптокарты (имя VPN_ISP1, раздел 2):

```
Hub1(config)#crypto map VPN_ISP1 2 ipsec-isakmp
Hub1(config-crypto-map)#description VPN between HUB1 and Spoke2 via ISP1
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE2_GRE1
Hub1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set local-address 172.16.100.2
Hub1(config-crypto-map)#set peer 172.16.2.2
Hub1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
```

2.5. Создайте криптокарту для построения туннелей через провайдера Hub1-ISP2:

2.5.1 Создайте первый раздел криптокарты (имя VPN_ISP2, раздел 1):

```
Hub1(config)#crypto map VPN_ISP2 1 ipsec-isakmp
Hub1(config-crypto-map)#description VPN between HUB1 and Spoke1 via ISP2
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE1_GRE2
Hub1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set local-address 172.17.100.2
Hub1(config-crypto-map)#set peer 172.16.1.2
Hub1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
```

2.5.2 Создайте второй раздел криптокарты (имя VPN_ISP2, раздел 2):

```
Hub1(config)#crypto map VPN_ISP2 2 ipsec-isakmp
Hub1(config-crypto-map)#description VPN between HUB1 and Spoke2 via ISP2
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE2_GRE2
Hub1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set local-address 172.17.100.2
Hub1(config-crypto-map)#set peer 172.16.2.2
Hub1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
```

3. Прикрепите созданные крипто-карты к внешним интерфейсам. VPN_ISP1 для `GigabitEthernet0/0`, VPN_ISP2 для `GigabitEthernet0/1`:

```
Hub1(config)#interface GigabitEthernet0/0
Hub1(config-if)#crypto map VPN_ISP1
Hub1(config-if)#exit
Hub1(config)#interface GigabitEthernet0/1
Hub1(config-if)#crypto map VPN_ISP2
Hub1(config-if)#exit
```

4. Примените настройки:

```
Hub1(config)#end
```

Настройка политики обработки COC (CRL)

В целях безопасности настоятельно рекомендуется включать проверку списков отзыва сертификатов. Разностные списки отозванных сертификатов (delta CRL) не поддерживаются.

Включите проверку COC и настройте автоматическую загрузку и импортирование в базу Продукта списка отозванных сертификатов с HTTP сервера CRL_distribution_point.

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка межсетевого экрана

1. Создайте ACL с именем `FW_OUTPUT_WAN` для исходящего на WAN интерфейсе трафика:

```
Hub1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1(config)#ip access-list extended FW_OUTPUT_WAN
Hub1(config-ext-nacl)#deny gre any any
Hub1(config-ext-nacl)#deny ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub1(config-ext-nacl)#deny ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
Hub1(config-ext-nacl)#permit ip any any
Hub1(config-ext-nacl)#exit
```

Данный ACL запрещает незашифрованный GRE трафик и незашифрованный трафик до защищаемых подсетей за криптошлюзами Spoke1 и Spoke2.

2. Создайте сетевой `object-group` с именем `SELF` для хранения в нем WAN адресов криптошлюза с целью дальнейшего его использования в ACL:

```
Hub1(config)#object-group network SELF
Hub1(config-network-group)#remark WAN IP
Hub1(config-network-group)#host 172.16.100.2
Hub1(config-network-group)#host 172.17.100.2
Hub1(config-network-group)#exit
```

3. Создайте ACL с именем `FW_INPUT_WAN` для входящего на WAN интерфейс трафика:

```
Hub1(config)#ip access-list extended FW_INPUT_WAN
Hub1(config-ext-nacl)#permit udp any object-group SELF eq 500
Hub1(config-ext-nacl)#permit udp any object-group SELF eq 4500
Hub1(config-ext-nacl)#permit icmp any object-group SELF
Hub1(config-ext-nacl)#permit esp any object-group SELF
Hub1(config-ext-nacl)#exit
```

4. Создайте правила инспектирования всего TCP и UDP трафика и задайте необходимое максимальное количество сессий:

```
Hub1(config)#ip inspect name INSPECT tcp
Hub1(config)#ip inspect name INSPECT udp
Hub1(config)#ip inspect tcp sessions-max 256000
Hub1(config)#ip inspect udp sessions-max 65536
```

5. Прикрепите правила межсетевого экранирования на WAN интерфейсы `GigabitEthernet0/0` и `GigabitEthernet0/1`:

```
Hub1(config)#interface GigabitEthernet0/0
Hub1(config-if)#ip access-group FW_INPUT_WAN in
Hub1(config-if)#ip access-group FW_OUTPUT_WAN out
Hub1(config-if)#ip inspect INSPECT out
Hub1(config-if)#exit
Hub1(config)#interface GigabitEthernet0/1
Hub1(config-if)#ip access-group FW_INPUT_WAN in
Hub1(config-if)#ip access-group FW_OUTPUT_WAN out
Hub1(config-if)#ip inspect INSPECT out
Hub1(config-if)#exit
```

6. Примените настройки:

```
Hub1(config)#end
```

Настройка PBR

Настройка PBR выполняется из linux консоли шлюза.

1. Создайте файл `/etc/network/pbr_common`, который будет содержать параметры, используемые в PBR скриптах:

```
SPOKES_IP_ADDRESSES="172.16.1.2 172.16.2.2"

WAN_0_INTERFACE="eth0"
WAN_0_IP_ADDRESS="172.16.100.2"
WAN_0_NEXTHOP_ADDRESS="172.16.100.1"

WAN_1_INTERFACE="eth1"
WAN_1_IP_ADDRESS="172.17.100.2"
WAN_1_NEXTHOP_ADDRESS="172.17.100.1"

GRE1_TABLE_ID="100"
GRE2_TABLE_ID="101"
PRIO="100"
```

2. Добавьте PBR политику, для этого создайте файл `/etc/network/if-pre-up.d/pbr` со следующим содержимым:

```
#!/bin/bash
# if-pre-up script

source /etc/network/pbr_common

if [[ $IFACE == $WAN_0_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_0="from $WAN_0_IP_ADDRESS to $SPOKE_IP lookup $GRE1_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_0 2>/dev/null || true
        ip rule add $RULE_WLAN_0
    done
fi

if [[ $IFACE == $WAN_1_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_1="from $WAN_1_IP_ADDRESS to $SPOKE_IP lookup $GRE2_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_1 2>/dev/null || true
        ip rule add $RULE_WLAN_1
    done
fi

exit 0
```

Файл `/etc/network/if-pre-up.d/pbr` используется для того, чтобы политики маршрутизации для интерфейсов eth0 и eth1 присутствовали в системе до перехода интерфейсов в состояние UP.

Данный файл будет выполнять следующие команды для добавления политики PBR:

```
ip rule add from 172.16.100.2 to 172.16.1.2 lookup 100 prio 100
ip rule add from 172.16.100.2 to 172.16.2.2 lookup 100 prio 100
ip rule add from 172.17.100.2 to 172.16.1.2 lookup 101 prio 100
ip rule add from 172.17.100.2 to 172.16.2.2 lookup 101 prio 100
```

Эти команды добавляют правила маршрутизации указанного в них трафика в соответствии с одной из двух таблиц маршрутизации. Первая таблица имеет индекс 100, вторая – 101. Для маршрутизации пакетов с IP адресом источника 172.16.100.2 и IP адресом назначения 172.16.1.2 будет использована таблица 100, а для пакетов с IP источника 172.17.100.2 и IP адресом назначения 172.16.1.2 – таблица 101.

3. Настройте маршруты для политики PBR, для этого создайте файл `/etc/network/if-up.d/pbr` со следующим содержимым:

```
#!/bin/bash
# if-up script

source /etc/network/pbr_common

if [[ $IFACE == $WAN_0_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        ip route add "$SPOKE_IP/32" via "$WAN_0_NEXTHOP_ADDRESS" dev "$WAN_0_INTERFACE"
table $GRE1_TABLE_ID
    done
fi

if [[ $IFACE == $WAN_1_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        ip route add "$SPOKE_IP/32" via "$WAN_1_NEXTHOP_ADDRESS" dev "$WAN_1_INTERFACE"
table $GRE2_TABLE_ID
    done
fi

exit 0
```

Файл `/etc/network/if-up.d/pbr` используется для того, чтобы маршрут по умолчанию добавлялся в политику PBR после перехода интерфейса eth1 в состояние UP.

Данный файл будет выполнять следующие команды для добавления маршрутов в политику PBR:

```
ip route add 172.16.1.2/32 via 172.16.100.1 dev eth0 dev eth0 table 100
ip route add 172.16.2.2/32 via 172.16.100.1 dev eth0 dev eth0 table 100
ip route add 172.16.1.2/32 via 172.17.100.1 dev eth1 dev eth1 table 101
ip route add 172.16.2.2/32 via 172.17.100.1 dev eth1 dev eth1 table 101
```

4. После перехода интерфейсов eth0 и eth1 в состояние DOWN требуется удалить политики, созданные модулем PBR. Для этого создайте файл `/etc/network/if-post-down.d/pbr` со следующим содержимым:

```
#!/bin/bash
# if-post-down script

source /etc/network/pbr_common

if [[ $IFACE == $WAN_0_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
```

```
        RULE_WLAN_0="from $WAN_0_IP_ADDRESS to $SPOKE_IP lookup $GRE1_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_0 2>/dev/null || true
    done
fi

if [[ $IFACE == $WAN_1_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_1="from $WAN_1_IP_ADDRESS to $SPOKE_IP lookup $GRE2_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_1 2>/dev/null || true
    done
fi

exit 0
```

Данный файл будет выполнять следующие команды для удаления политики PBR:

```
ip rule del from 172.16.100.2 to 172.16.1.2 lookup 100 prio 100
ip rule del from 172.16.100.2 to 172.16.2.2 lookup 100 prio 100
ip rule del from 172.17.100.2 to 172.16.1.2 lookup 101 prio 100
ip rule del from 172.17.100.2 to 172.16.2.2 lookup 101 prio 100
```

5. Присвойте созданным файлам права на запуск:

```
root@Hub1:~# chmod +x /etc/network/if-pre-up.d/pbr
root@Hub1:~# chmod +x /etc/network/if-up.d/pbr
root@Hub1:~# chmod +x /etc/network/if-post-down.d/pbr
```

6. Перезапустите сервис networking:

Если после начала настройки сетевых интерфейсов шлюз не перезагружался, то перезагрузите его. В таком случае перезапускать сервис networking не требуется.

```
root@Hub1:~# systemctl restart networking.service
```

Настройка динамической маршрутизации BGP

BGP настраивается средствами консоли сервиса FRR.

1. Добавьте сервис FRR в автозагрузку и запустите его из linux bash:

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

```
Synchronizing state of frr.service with SysV service script with /lib/system/system-
sysv-install.
Executing: /lib/system/system-sysv-install enable frr
```

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

2. Войдите в консоль сервиса FRR, для этого нужно из linux bash набрать следующую команду:

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

```
Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

Hub1#
```

3. Настройте BGP (см. описание http://docs.frrouting.org):

```
Hub1# configure terminal
Hub1(config)# router bgp 65000
Hub1(config-router)# bgp router-id 172.16.100.2
Hub1(config-router)# neighbor 10.10.10.2 remote-as 65000
Hub1(config-router)# neighbor 10.10.10.6 remote-as 65000
Hub1(config-router)# neighbor 10.10.10.10 remote-as 65000
Hub1(config-router)# neighbor 10.10.10.14 remote-as 65000
Hub1(config-router)# network 192.168.100.0/24
```

4. Сохраните настройки сервиса FRR:

```
Hub1(config-router)# end
Hub1# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
Hub1# exit
root@Hub1:~#
```

В Приложении представлены тексты конфигураций для криптошлюза Hub1:

- текст консоли cisco-like;
- текст конфигурации LSP;
- содержимое `/etc/multiwan/config.yaml`;
- конфигурация FRR;
- содержимое `/etc/network/spokes_ip_list`;
- содержимое `/etc/network/if-pre-up.d/pbr`;
- содержимое `/etc/network/if-up.d/pbr`;
- содержимое `/etc/network/if-post-down.d/pbr`.

Настройка криптошлюза Hub1 для подключения дополнительных филиальных криптошлюзов

В данном сценарии описывается настройка для случая, когда к центральному криптошлюзу Hub1 подключаются два филиальных криптошлюза. Если требуется подключить дополнительные филиалы к центру, то для построения безопасного соединения с каждым новым филиальным криптошлюзом на Hub1 потребуется выполнить следующие настройки:

- В настройках PBR в файле `/etc/network/pbr_common` добавить адрес нового криптошлюза в список `SPOKES_IP_ADDRESSES`. Для того, чтобы новые настройки PBR вступили в силу, следует перезапустить `networking.service` или в ручную добавить в linux bash консоли криптошлюза соответствующие правила и маршруты (пример команд при подключении нового филиального криптошлюза с адресом 172.16.3.2):

```
root@Hub1:~# ip rule add from 172.16.100.2 to 172.16.3.2 lookup 100 pref 100
root@Hub1:~# ip rule add from 172.17.100.2 to 172.16.3.2 lookup 101 pref 100
root@Hub1:~# ip route add 172.16.3.2/32 via 172.16.100.1 dev eth0 table 100
root@Hub1:~# ip route add 172.16.3.2/32 via 172.17.100.1 dev eth1 table 101
```

- Добавить два GRE интерфейса для построения GRE туннелей до нового криптошлюза;
- Добавить в криптокарты на WAN интерфейсах еще один раздел для шифрования GRE трафика между Hub1 и новым криптошлюзом;
- В настройках FRR добавить IP адреса GRE интерфейсов нового криптошлюза в качестве BGP neighbor.

Настройка криптошлюза Spoke1

Начальные настройки

Войдите в CLI разграничения доступа (пользователь и пароль по умолчанию: `administrator`, `s-terra`) установите правильный тип терминала и требуемую ширину, смените пароль для пользователя `administrator`.

Перейдите в linux bash консоль, установите пароль для пользователя `root`, правильное время и временную зону.

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройки PKI (запросы и сертификаты)

Продукт не поддерживает разностные (delta) списки отзыва сертификатов, только базовые. Учитывайте это при выборе или развертывании УЦ.

Для аутентификации партнеров по IPsec можно использовать цифровые сертификаты, выпущенные при помощи сертифицированного СКЗИ, или предварительно распределенные ключи, созданные и введённые в соответствии с правилами пользования СКЗИ (появились в версии 5.0).

Сгенерируйте запрос на сертификат криптошлюза, доставьте его на УЦ и выпустите по нему сертификат. Сертификат криптошлюза вместе с сертификатом УЦ доставьте на шлюз и импортируйте их в базу Продукта.

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка сетевых параметров

Изменение состояния интерфейсов и назначение IP адресов применяются сразу после ввода соответствующих команд. Политика безопасности применяется после выхода из режима конфигурирования.

1. Войдите в cisco-like консоль и задайте имя устройства:

```
sterragate#configure terminal
sterragate(config)#hostname Spoke1
```

2. Включите внешний `GigabitEthernet0/0` и внутренний `GigabitEthernet0/1` интерфейсы:

```
Spoke1(config)#interface range GigabitEthernet 0/0 - 1
Spoke1(config-if-range)#no shutdown
Spoke1(config-if-range)#exit
```

3. Убедитесь, что интерфейсы административно включены и line protocol (способность интерфейса передавать пакеты в данный момент) находится в состоянии up:

```
Spoke1(config)#do show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware address is 0050.569e.b06d
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.d3c3
  MTU 1500 bytes
...
```

Если line protocol находится в состоянии down, то проверьте подключение сетевого интерфейса криптошлюза к коммутационному или прочему оборудованию.

4. Задайте IP адреса в соответствии со схемой стенда на внешнем `GigabitEthernet0/0` и внутреннем `GigabitEthernet0/1` интерфейсах:

```
Spoke1(config)#interface GigabitEthernet 0/0
Spoke1(config-if)#ip address 172.16.1.2 255.255.255.0
Spoke1(config-if)#exit
Spoke1(config)#interface GigabitEthernet 0/1
Spoke1(config-if)#ip address 192.168.1.1 255.255.255.0
Spoke1(config-if)#exit
```

5. Задайте маршрут по умолчанию через устройство Router1:

```
Spoke1(config)#ip route 0.0.0.0 0.0.0.0 172.16.1.1
```

6. Проверьте доступность устройства Router1:

```
Spoke1(config)#do ping 172.16.1.1
```

```
PING 172.16.1.1 (172.16.1.1) 100(128) bytes of data.
108 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.197 ms
108 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.423 ms
108 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.386 ms
108 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.429 ms
108 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.446 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4080ms
rtt min/avg/max/mdev = 0.197/0.376/0.446/0.091 ms
```

Настройка GRE интерфейсов

1. Настройте интерфейс GRE туннеля между Spoke1 и Hub1 через провайдера Hub1-ISP1:

```
Spoke1(config)#interface Tunnel 1
Spoke1(config-if)#description GRE between Spoke1 and Hub1 via ISP1
Spoke1(config-if)#ip address 10.10.10.2 255.255.255.252
Spoke1(config-if)#mtu 1400
Spoke1(config-if)#tunnel source 172.16.1.2
Spoke1(config-if)#tunnel destination 172.16.100.2
Spoke1(config-if)#no tunnel path-mtu-discovery
Spoke1(config-if)#tunnel ignore-df
Spoke1(config-if)#exit
```

2. Настройте интерфейс GRE туннеля между Spoke1 и Hub1 через провайдера Hub1-ISP2:

```
Spoke1(config)#interface Tunnel 2
Spoke1(config-if)#description GRE between Spoke1 and Hub1 via ISP2
Spoke1(config-if)#ip address 10.10.10.6 255.255.255.252
Spoke1(config-if)#mtu 1400
Spoke1(config-if)#tunnel source 172.16.1.2
Spoke1(config-if)#tunnel destination 172.17.100.2
Spoke1(config-if)#no tunnel path-mtu-discovery
Spoke1(config-if)#tunnel ignore-df
Spoke1(config-if)#exit
```

Настройка шифрования

1. Параметры IKE.

   1.1. Задайте общие параметры IKE:

```
Spoke1(config)#crypto isakmp identity dn
Spoke1(config)#crypto isakmp keepalive 3 2
Spoke1(config)#crypto isakmp keepalive retry-count 5
Spoke1(config)#crypto isakmp fragmentation
Spoke1(config)#crypto isakmp security-association lifetime delta 50
Spoke1(config)#crypto isakmp initiator-sessions-max 100
Spoke1(config)#crypto isakmp responder-sessions-max 100
```

   1.2. Создайте политику, описывающую параметры IKE туннеля:

```
Spoke1(config)#crypto isakmp policy 1
```

```
Spoke1(config-isakmp)#encryption magma
Spoke1(config-isakmp)#hash stribog-256
Spoke1(config-isakmp)#authentication gost-sig
Spoke1(config-isakmp)#group vko2
Spoke1(config-isakmp)#exit
```

1.3. Создайте профиль настроек ISAKMP (алгоритм IKE) под именем `NEW_ALGS`, чтобы далее его прикрепить к криптокарте:

```
Spoke1(config)#crypto isakmp profile NEW_ALGS
Spoke1(conf-isa-prof)#match identity dn
Spoke1(conf-isa-prof)#set identity mapping
Spoke1(conf-isa-prof)#crypto isakmp policy 1
```

2. Параметры IPsec.

2.1. Отключите обновление ключа IPsec SA по достижению определенного объема переданного трафика:

```
Spoke1(config)#crypto ipsec security-association lifetime kilobytes unlimited
```

2.2. Задайте комбинированный алгоритм шифрования и имитозащиты (набор преобразований) для трафика:

```
Spoke1(config)#crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
Spoke1(cfg-crypto-trans)#exit
```

2.3. Создайте списки доступа (ACL) для трафика, который нужно защищать между центральным офисом и филиалом №1:

```
Spoke1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE1
Spoke1(config-ext-nacl)#permit gre host 172.16.1.2 host 172.16.100.2
Spoke1(config-ext-nacl)#exit
Spoke1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE2
Spoke1(config-ext-nacl)#permit gre host 172.16.1.2 host 172.17.100.2
Spoke1(config-ext-nacl)#exit
Spoke1(config)#
```

2.4. Создайте крипто-карту для построения туннелей через Hub1-ISP1 (имя VPN, раздел 1):

```
Spoke1(config)#crypto map VPN 1 ipsec-isakmp
Spoke1(config-crypto-map)#description VPN between Spoke1 and HUB1 via ISP1
Spoke1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE1_GRE1
Spoke1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Spoke1(config-crypto-map)#set peer 172.16.100.2
Spoke1(config-crypto-map)#set isakmp-profile NEW_ALGS
Spoke1(config-crypto-map)#set pre-fragmentation
Spoke1(config-crypto-map)#set dead-connection history off
Spoke1(config-crypto-map)#exit
```

2.5. Добавьте к крипто-карте раздел для туннелей через Hub1-ISP2 (имя VPN, раздел 2):

```
Spoke1(config)#crypto map VPN 2 ipsec-isakmp
Spoke1(config-crypto-map)#description VPN between Spoke1 and HUB1 via ISP2
Spoke1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE1_GRE2
Spoke1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Spoke1(config-crypto-map)#set peer 172.17.100.2
Spoke1(config-crypto-map)#set isakmp-profile NEW_ALGS
Spoke1(config-crypto-map)#set pre-fragmentation
Spoke1(config-crypto-map)#set dead-connection history off
Spoke1(config-crypto-map)#exit
```

2.6. Примените настройки:

```
Spoke1(config-if)#end
```

Настройка политики обработки COC (CRL)

В целях безопасности настоятельно рекомендуется включать проверку списков отзыва сертификатов. Разностные списки отозванных сертификатов (delta CRL) не поддерживаются.

Включите проверку COC и настройте автоматическую загрузку и импортирование в базу Продукта списка отозванных сертификатов с HTTP сервера CRL_distribution_point.

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка межсетевого экрана

1. Создайте ACL с именем `FW_OUTPUT_WAN` для исходящего на WAN интерфейсе трафика:

```
Spoke1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Spoke1(config)#ip access-list extended FW_OUTPUT_WAN
Spoke1(config-ext-nacl)#deny gre any any
Spoke1(config-ext-nacl)#deny ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
Spoke1(config-ext-nacl)#permit ip any any
Spoke1(config-ext-nacl)#exit
```

Данный ACL запрещает незашифрованный GRE трафик и незашифрованный трафик до защищаемой подсети за криптошлюзом Hub1.

2. Создайте сетевой `object-group` с именем `SELF` для хранения в нем WAN адресов криптошлюза с целью дальнейшего его использования в ACL:

```
Spoke1(config)#object-group network SELF
Spoke1(config-network-group)#remark WAN IP
Spoke1(config-network-group)#host 172.16.1.2
Spoke1(config-network-group)#exit
```

3. Создайте ACL с именем `FW_INPUT_WAN` для входящего на WAN интерфейс трафика:

```
Spoke1(config)#ip access-list extended FW_INPUT_WAN
Spoke1(config-ext-nacl)#permit udp any object-group SELF eq 500
Spoke1(config-ext-nacl)#permit udp any object-group SELF eq 4500
Spoke1(config-ext-nacl)#permit icmp any object-group SELF
Spoke1(config-ext-nacl)#permit esp any object-group SELF
Spoke1(config-ext-nacl)#exit
```

4. Создайте правила инспектирования всего TCP и UDP трафика и задайте необходимое максимальное количество сессий:

```
Spoke1(config)#ip inspect name INSPECT tcp
Spoke1(config)#ip inspect name INSPECT udp
Spoke1(config)#ip inspect tcp sessions-max 256000
Spoke1(config)#ip inspect udp sessions-max 65536
```

5. Прикрепите правила межсетевого экранирования на WAN интерфейс `GigabitEthernet0/0`:

```
Spoke1(config)#interface GigabitEthernet0/0
Spoke1(config-if)#ip access-group FW_INPUT_WAN in
Spoke1(config-if)#ip access-group FW_OUTPUT_WAN out
Spoke1(config-if)#ip inspect INSPECT out
Spoke1(config-if)#exit
```

6. Примените настройки:

```
Spoke1(config)#end
```

Настройка динамической маршрутизации BGP

BGP настраивается средствами консоли сервиса FRR.

1. Добавьте сервис FRR в автозагрузку и запустите его из linux bash:

```
root@Spoke1:~# systemctl enable frr
```

```
Synchronizing state of frr.service with SysV service script with /lib/system/system-sysv-install.
Executing: /lib/system/system-sysv-install enable frr
```

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

2. Войдите в консоль сервиса FRR, для этого нужно из linux bash набрать следующую команду:

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

```
Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

Hub1#
```

3. Настройте BGP (см. описание http://docs.frrouting.org):

```
Spoke1# configure terminal
Spoke1(config)# router bgp 65000
Spoke1(config-router)# bgp router-id 172.16.1.2
Spoke1(config-router)# neighbor 10.10.10.1 remote-as 65000
Spoke1(config-router)# neighbor 10.10.10.5 remote-as 65000
Spoke1(config-router)# network 192.168.1.0/24
```

4. Сохраните настройки сервиса FRR:

```
Spoke1(config-router)# do write
Spoke1(config-router)# end
Spoke1# exit
```

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст консоли cisco-like;
- текст конфигурации LSP;
- конфигурация FRR.

Настройка криптошлюза Spoke2

Настройка криптошлюза Spoke2 выполняется аналогично настройке Spoke1.

В Приложении представлены тексты конфигураций для криптошлюза Spoke2:

- текст консоли cisco-like;
- текст конфигурации LSP;
- конфигурация FRR.

Проверка работоспособности стенда

Проверку работоспособности нужно выполнять после завершения настройки всех устройств стенда.

Проверка IP связности

1. Проверьте, что с криптошлюза Hub1 доступны маршрутизаторы провайдеров (Hub1-ISP1 и Hub1-ISP2), и с криптошлюзов Spoke1 и Spoke2 по ICMP доступен шлюз по умолчанию (Router1). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub1#ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 100(128) bytes of data.
108 bytes from 172.16.100.1: icmp_seq=1 ttl=64 time=0.297 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.275 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.412 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.320 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.309 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4060ms
rtt min/avg/max/mdev = 0.275/0.322/0.412/0.051 ms
```

```
Hub1#ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 100(128) bytes of data.
108 bytes from 172.17.100.1: icmp_seq=1 ttl=64 time=0.433 ms
108 bytes from 172.17.100.1: icmp_seq=2 ttl=64 time=0.451 ms
108 bytes from 172.17.100.1: icmp_seq=3 ttl=64 time=0.412 ms
108 bytes from 172.17.100.1: icmp_seq=4 ttl=64 time=0.331 ms
108 bytes from 172.17.100.1: icmp_seq=5 ttl=64 time=0.653 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.331/0.456/0.653/0.106 ms
```

```
Spoke1#ping 172.16.1.1
```

```
PING 172.16.1.1 (172.16.1.1) 100(128) bytes of data.
108 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.192 ms
108 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.369 ms
108 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.266 ms
108 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.291 ms
108 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.417 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4101ms
rtt min/avg/max/mdev = 0.192/0.307/0.417/0.078 ms
```

```
Spoke2#ping 172.16.2.1
```

```
PING 172.16.2.1 (172.16.2.1) 100(128) bytes of data.
108 bytes from 172.16.2.1: icmp_seq=1 ttl=64 time=0.347 ms
108 bytes from 172.16.2.1: icmp_seq=2 ttl=64 time=0.428 ms
108 bytes from 172.16.2.1: icmp_seq=3 ttl=64 time=0.343 ms
108 bytes from 172.16.2.1: icmp_seq=4 ttl=64 time=0.264 ms
108 bytes from 172.16.2.1: icmp_seq=5 ttl=64 time=0.431 ms
```

```
--- 172.16.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4088ms
rtt min/avg/max/mdev = 0.264/0.362/0.431/0.062 ms
```

2. Убедитесь в том, что с защищаемых устройств host-b-hub1, host-b-spoke1 и host-b-spoke2 доступен по протоколу ICMP криптошлюз, являющийся шлюзом по умолчанию для данного устройства. Для этого выполните команду `ping` из linux консоли защищаемых устройств.

```
root@Host-b-hub1:~# ping 192.168.100.1 -c 5
```

```
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.211 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.235 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.223 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=0.254 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=64 time=0.233 ms

--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4088ms
rtt min/avg/max/mdev = 0.211/0.231/0.254/0.017 ms
```

```
root@Host-b-spoke1:~# ping 192.168.1.1 -c 5
```

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.214 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.189 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.210 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.212 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.189/0.209/0.223/0.019 ms
```

```
root@Host-b-spoke2:~# ping 192.168.2.1 -c 5
```

```
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.295 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.296 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.187 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.308 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=0.283 ms

--- 192.168.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.187/0.273/0.308/0.044 ms
```

3. Убедитесь в том, что с криптошлюзов Spoke1 и Spoke2 доступны интерфейсы провайдеров Hub1-ISP1 и Hub1-ISP2 направленные в сторону Hub1 (172.16.100.1/24 и 172.17.100.1/24). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Spoke1#ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 100(128) bytes of data.
108 bytes from 172.16.100.1: icmp_seq=1 ttl=63 time=0.384 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=63 time=0.477 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=63 time=0.526 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=63 time=0.524 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=63 time=0.668 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4075ms
rtt min/avg/max/mdev = 0.384/0.515/0.668/0.096 ms
```

```
Spoke1#ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 100(128) bytes of data.
```

```
108 bytes from 172.17.100.1: icmp_seq=1 ttl=63 time=0.459 ms
108 bytes from 172.17.100.1: icmp_seq=2 ttl=63 time=0.583 ms
108 bytes from 172.17.100.1: icmp_seq=3 ttl=63 time=0.498 ms
108 bytes from 172.17.100.1: icmp_seq=4 ttl=63 time=0.632 ms
108 bytes from 172.17.100.1: icmp_seq=5 ttl=63 time=0.413 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4084ms
rtt min/avg/max/mdev = 0.413/0.517/0.632/0.080 ms
```

```
Spoke2#ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 100(128) bytes of data.
108 bytes from 172.16.100.1: icmp_seq=1 ttl=63 time=0.285 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=63 time=0.596 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=63 time=0.607 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=63 time=0.732 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=63 time=0.674 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4024ms
rtt min/avg/max/mdev = 0.285/0.578/0.732/0.154 ms
```

```
Spoke2#ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 100(128) bytes of data.
108 bytes from 172.17.100.1: icmp_seq=1 ttl=63 time=0.492 ms
108 bytes from 172.17.100.1: icmp_seq=2 ttl=63 time=0.405 ms
108 bytes from 172.17.100.1: icmp_seq=3 ttl=63 time=0.715 ms
108 bytes from 172.17.100.1: icmp_seq=4 ttl=63 time=0.591 ms
108 bytes from 172.17.100.1: icmp_seq=5 ttl=63 time=0.798 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4050ms
rtt min/avg/max/mdev = 0.405/0.600/0.798/0.142 ms
```

Проверка работы PBR

1. Убедитесь в том, что база маршрутных политик Hub1 содержит правила для пакетов, отправленных с интерфейсов 172.16.100.2 и 172.17.100.2 к криптошлюзам Spoke1 (172.16.1.2) и Spoke2 (172.16.2.2), и применяется таблица маршрутизации 100 и 101 соответственно. Для этого выполните команду `run ip rule show` из cisco-like консоли криптошлюза Hub1.

   ```
   Hub1#run ip rule show
   ```

   ```
   0:    from all lookup local
   100:  from 172.16.100.2 to 172.16.1.2 lookup 100
   100:  from 172.16.100.2 to 172.16.2.2 lookup 100
   100:  from 172.17.100.2 to 172.16.1.2 lookup 101
   100:  from 172.17.100.2 to 172.16.2.2 lookup 101
   100:  from all to 172.16.101.15 fwmark 0x3e9/0x7ff oif eth1 ipproto icmp lookup 1001
   proto multiwan
   100:  from all to 172.16.101.15 fwmark 0x3e8/0x7ff oif eth0 ipproto icmp lookup 1000
   proto multiwan
   32766: from all lookup main
   32767: from all lookup default
   ```

2. Убедитесь в том, что таблицы содержат маршруты до криптошлюзов Spoke1 и Spoke2, проходящие через маршрутизаторы соответствующих провайдеров. Для этого выполните команду `run ip route show table` с номерами таблиц из cisco-like консоли криптошлюза Hub1.

   ```
   Hub1#run ip route show table 100
   ```

   ```
   172.16.1.2 via 172.16.100.1 dev eth0
   172.16.2.2 via 172.16.100.1 dev eth0
   ```

   ```
   Hub1#run ip route show table 101
   ```

   ```
   172.16.1.2 via 172.17.100.1 dev eth1
   ```

```
172.16.2.2 via 172.17.100.1 dev eth1
```

3. Убедитесь в том, что с криптошлюзов Spoke1 и Spoke2 доступен по ICMP криптошлюз Hub1 (по внешним интерфейсам). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Spoke1#ping 172.16.100.2
```

```
PING 172.16.100.2 (172.16.100.2) 100(128) bytes of data.
108 bytes from 172.16.100.2: icmp_seq=1 ttl=62 time=0.715 ms
108 bytes from 172.16.100.2: icmp_seq=2 ttl=62 time=1.03 ms
108 bytes from 172.16.100.2: icmp_seq=3 ttl=62 time=0.987 ms
108 bytes from 172.16.100.2: icmp_seq=4 ttl=62 time=1.06 ms
108 bytes from 172.16.100.2: icmp_seq=5 ttl=62 time=0.989 ms

--- 172.16.100.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4036ms
rtt min/avg/max/mdev = 0.715/0.957/1.064/0.124 ms
```

```
Spoke1#ping 172.17.100.2
```

```
PING 172.17.100.2 (172.17.100.2) 100(128) bytes of data.
108 bytes from 172.17.100.2: icmp_seq=1 ttl=62 time=0.897 ms
108 bytes from 172.17.100.2: icmp_seq=2 ttl=62 time=0.781 ms
108 bytes from 172.17.100.2: icmp_seq=3 ttl=62 time=0.903 ms
108 bytes from 172.17.100.2: icmp_seq=4 ttl=62 time=0.919 ms
108 bytes from 172.17.100.2: icmp_seq=5 ttl=62 time=1.06 ms

--- 172.17.100.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 0.781/0.912/1.062/0.093 ms
```

```
Spoke2#ping 172.16.100.2
```

```
PING 172.16.100.2 (172.16.100.2) 100(128) bytes of data.
108 bytes from 172.16.100.2: icmp_seq=1 ttl=62 time=0.879 ms
108 bytes from 172.16.100.2: icmp_seq=2 ttl=62 time=0.852 ms
108 bytes from 172.16.100.2: icmp_seq=3 ttl=62 time=0.762 ms
108 bytes from 172.16.100.2: icmp_seq=4 ttl=62 time=0.891 ms
108 bytes from 172.16.100.2: icmp_seq=5 ttl=62 time=1.25 ms

--- 172.16.100.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.762/0.925/1.245/0.165 ms
```

```
Spoke2#ping 172.17.100.2
```

```
PING 172.17.100.2 (172.17.100.2) 100(128) bytes of data.
108 bytes from 172.17.100.2: icmp_seq=1 ttl=62 time=0.864 ms
108 bytes from 172.17.100.2: icmp_seq=2 ttl=62 time=0.902 ms
108 bytes from 172.17.100.2: icmp_seq=3 ttl=62 time=1.23 ms
108 bytes from 172.17.100.2: icmp_seq=4 ttl=62 time=0.810 ms
108 bytes from 172.17.100.2: icmp_seq=5 ttl=62 time=0.906 ms

--- 172.17.100.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 0.810/0.942/1.228/0.147 ms
```

Проверка построения защищенного соединения между подсетями Hub1 и Spoke1

1. Защищенные соединения на криптошлюзах должны построиться автоматически, за счет наличия служебного трафика динамической маршрутизации. Убедитесь в том, что на криптошлюзах Hub1, Spoke1 и Spoke2 успешно созданы IPsec SA. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 16 (172.17.100.2,500)-(172.16.2.2,500) active 2232 2328
2 17 (172.16.100.2,500)-(172.16.2.2,500) active 2232 2328
3 18 (172.16.100.2,500)-(172.16.1.2,500) active 2328 2360
4 19 (172.17.100.2,500)-(172.16.1.2,500) active 2264 2360

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 17 (172.16.100.2,*)-(172.16.2.2,*) 47 ESP tunn 288 184
2 18 (172.17.100.2,*)-(172.16.2.2,*) 47 ESP tunn 288 184
3 19 (172.16.100.2,*)-(172.16.2.2,*) 47 ESP tunn 8384 8464
4 20 (172.17.100.2,*)-(172.16.2.2,*) 47 ESP tunn 7824 8464
5 21 (172.16.100.2,*)-(172.16.1.2,*) 47 ESP tunn 104 0
6 22 (172.17.100.2,*)-(172.16.1.2,*) 47 ESP tunn 0 8368
7 23 (172.17.100.2,*)-(172.16.1.2,*) 47 ESP tunn 8848 0
8 24 (172.16.100.2,*)-(172.16.1.2,*) 47 ESP tunn 8544 7448
```

```
Spoke1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 7 (172.16.1.2,500)-(172.16.100.2,500) active 2360 2328
2 8 (172.16.1.2,500)-(172.17.100.2,500) active 2360 2264

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 9 (172.16.1.2,*)-(172.16.100.2,*) 47 ESP tunn 208 104
2 10 (172.16.1.2,*)-(172.17.100.2,*) 47 ESP tunn 0 12632
3 11 (172.16.1.2,*)-(172.16.100.2,*) 47 ESP tunn 11312 12408
4 12 (172.16.1.2,*)-(172.17.100.2,*) 47 ESP tunn 12232 0
```

```
Spoke2#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 6 (172.16.2.2,500)-(172.17.100.2,500) active 2328 2232
2 7 (172.16.2.2,500)-(172.16.100.2,500) active 2328 2232

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 9 (172.16.2.2,*)-(172.16.100.2,*) 47 ESP tunn 184 184
2 10 (172.16.2.2,*)-(172.17.100.2,*) 47 ESP tunn 184 184
3 11 (172.16.2.2,*)-(172.16.100.2,*) 47 ESP tunn 14536 13816
4 12 (172.16.2.2,*)-(172.17.100.2,*) 47 ESP tunn 14536 13176
```

Вывод команды свидетельствует о том, что на криптошлюзах установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка динамической маршрутизации

1. Убедитесь в том, что на криптошлюзах Hub1, Spoke1 и Spoke2 имеются маршруты до защищаемой подсети соседа, полученные по протоколу BGP.

```
Hub1#run ip route show
```

```
default via 172.16.100.1 dev eth0 proto multiwan metric 800
```

```
10.10.10.0/30 dev gre_1 proto kernel scope link src 10.10.10.1
10.10.10.4/30 dev gre_2 proto kernel scope link src 10.10.10.5
10.10.10.8/30 dev gre_3 proto kernel scope link src 10.10.10.9
10.10.10.12/30 dev gre_4 proto kernel scope link src 10.10.10.13
172.16.100.0/24 dev eth0 proto kernel scope link src 172.16.100.2
172.17.100.0/24 dev eth1 proto kernel scope link src 172.17.100.2
192.168.1.0/24 nhid 175 proto bgp metric 20
      nexthop via 10.10.10.2 dev gre_1 weight 1
      nexthop via 10.10.10.6 dev gre_2 weight 1
192.168.2.0/24 nhid 172 proto bgp metric 20
      nexthop via 10.10.10.10 dev gre_3 weight 1
      nexthop via 10.10.10.14 dev gre_4 weight 1
192.168.100.0/24 dev eth2 proto kernel scope link src 192.168.100.1
```

```
Spoke1#run ip route show
```

```
default via 172.16.1.1 dev eth0 proto st_policy
10.10.10.0/30 dev gre_1 proto kernel scope link src 10.10.10.2
10.10.10.4/30 dev gre_2 proto kernel scope link src 10.10.10.6
172.16.1.0/24 dev eth0 proto kernel scope link src 172.16.1.2
192.168.1.0/24 dev eth1 proto kernel scope link src 192.168.1.1
192.168.100.0/24 nhid 24 proto bgp metric 20
      nexthop via 10.10.10.1 dev gre_1 weight 1
      nexthop via 10.10.10.5 dev gre_2 weight 1
```

```
Spoke2#run ip route show
```

```
default via 172.16.2.1 dev eth0 proto st_policy
10.10.10.8/30 dev gre_1 proto kernel scope link src 10.10.10.10
10.10.10.12/30 dev gre_2 proto kernel scope link src 10.10.10.14
172.16.2.0/24 dev eth0 proto kernel scope link src 172.16.2.2
192.168.2.0/24 dev eth1 proto kernel scope link src 192.168.2.1
192.168.100.0/24 nhid 24 proto bgp metric 20
      nexthop via 10.10.10.9 dev gre_1 weight 1
      nexthop via 10.10.10.13 dev gre_2 weight 1
```

Если маршруты в системе отсутствуют, проверьте правильность конфигурации FRR.

Проверка наличия балансировки трафика

1. Создайте на host-b-hub1 вторичные IP адреса из сети 192.168.100.0/24. Для этого измените файл `/etc/network/interfaces` на host-b-hub1 и перезапустите сервис networking:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto ens160
iface ens160 inet static
address 192.168.100.100
netmask 255.255.255.0
gateway 192.168.100.1

auto ens160:0
iface ens160:0 inet static
address 192.168.100.101
netmask 255.255.255.0

auto ens160:1
iface ens160:1 inet static
address 192.168.100.102
```

```
netmask 255.255.255.0

auto ens160:2
iface ens160:2 inet static
address 192.168.100.103
netmask 255.255.255.0

auto ens160:3
iface ens160:3 inet static
address 192.168.100.104
netmask 255.255.255.0
```

2. При помощи ICMP трафика, посылаемого с устройства host-b-spoke1 на IP адреса host-b-hub1, проанализируйте сетевой трафик в GRE-туннелях. Для этого на криптошлюзе Spoke1 с помощью `tcpdump` перехватите пакеты с интерфейсов GRE1 и GRE2:

```
Spoke1#run tcpdump -i gre_1 -c 5 src 192.168.1.100
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on gre_1, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
13:18:17.937476 IP 192.168.1.100 > 192.168.100.100: ICMP echo request, id 985, seq
43576
13:18:17.937490 IP 192.168.1.100 > 192.168.100.104: ICMP echo request, id 989, seq
25097
13:18:17.937498 IP 192.168.1.100 > 192.168.100.103: ICMP echo request, id 988, seq
30495
13:18:17.939141 IP 192.168.1.100 > 192.168.100.100: ICMP echo request, id 985, seq
43577
13:18:17.939160 IP 192.168.1.100 > 192.168.100.104: ICMP echo request, id 989, seq
25098
5 packets captured
7 packets received by filter
0 packets dropped by kernel
```

```
Spoke1# run tcpdump -i gre_1 -c 5 dst 192.168.1.100
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on gre_1, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
13:18:24.937438 IP 192.168.100.102 > 192.168.1.100: ICMP echo reply, id 987, seq 39232,
13:18:24.938215 IP 192.168.100.104 > 192.168.1.100: ICMP echo reply, id 989, seq 31508,
13:18:24.939151 IP 192.168.100.102 > 192.168.1.100: ICMP echo reply, id 987, seq 39233,
13:18:24.939634 IP 192.168.100.104 > 192.168.1.100: ICMP echo reply, id 989, seq 31509,
13:18:24.940609 IP 192.168.100.102 > 192.168.1.100: ICMP echo reply, id 987, seq 39234,
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

```
Spoke1#run tcpdump -i gre_2 -c 5 src 192.168.1.100
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on gre_2, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
13:18:30.285790 IP 192.168.1.100 > 192.168.100.102: ICMP echo request, id 987, seq
44200
13:18:30.285797 IP 192.168.1.100 > 192.168.100.101: ICMP echo request, id 986, seq
47882
13:18:30.287592 IP 192.168.1.100 > 192.168.100.102: ICMP echo request, id 987, seq
44201
13:18:30.287627 IP 192.168.1.100 > 192.168.100.101: ICMP echo request, id 986, seq
47883
13:18:30.288838 IP 192.168.1.100 > 192.168.100.101: ICMP echo request, id 986, seq
47884
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

```
Spoke1# run tcpdump -i gre_2 -c 5 dst 192.168.1.100
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on gre_2, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
13:18:34.609100 IP 192.168.100.100 > 192.168.1.100: ICMP echo reply, id 985, seq 59019,
13:18:34.609500 IP 192.168.100.103 > 192.168.1.100: ICMP echo reply, id 988, seq 45985,
13:18:34.609587 IP 192.168.100.101 > 192.168.1.100: ICMP echo reply, id 986, seq 51878,
13:18:34.610752 IP 192.168.100.100 > 192.168.1.100: ICMP echo reply, id 985, seq 59020,
13:18:34.610775 IP 192.168.100.103 > 192.168.1.100: ICMP echo reply, id 988, seq 45986,
5 packets captured
9 packets received by filter
0 packets dropped by kernel
```

Видно, что через GRE1-туннель проходят запросы до 192.168.100.100, 192.168.100.104 и 192.168.100.103, а также ответы от 192.168.100.102 и 192.168.100.104. Через GRE2-туннель проходят запросы до 192.168.100.101 и 192.168.100.102, а также ответы от 192.168.100.100, 192.168.100.101 и 192.168.100.103. Такое поведение не постоянно и возможны случаи, когда у одного из криптошлюзов трафик уходит по обоим GRE-туннелям, но у криптошлюза партнера трафик отсылает только по одному GRE-туннелю.

Приложение

Конфигурации криптошлюза Hub1

1. 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
aaa new-model
!
!
hostname Hub1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.100.2
 host 172.17.100.2
!
ip access-list extended FW_OUTPUT_WAN
 deny   gre any any
 deny   ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
 deny   ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
 permit ip any any
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE1
 permit gre host 172.16.100.2 host 172.16.1.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE2
```

```
 permit gre host 172.17.100.2 host 172.16.1.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE2_GRE1
 permit gre host 172.16.100.2 host 172.16.2.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE2_GRE2
 permit gre host 172.17.100.2 host 172.16.2.2
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN_ISP1 1 ipsec-isakmp
 description VPN between HUB1 and Spoke1 via ISP1
 match address IPSEC_ACL_HUB1_SPOKE1_GRE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
crypto map VPN_ISP1 2 ipsec-isakmp
 description VPN between HUB1 and Spoke2 via ISP1
 match address IPSEC_ACL_HUB1_SPOKE2_GRE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set local-address 172.16.100.2
 set peer 172.16.2.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_ISP2 1 ipsec-isakmp
 description VPN between HUB1 and Spoke1 via ISP2
 match address IPSEC_ACL_HUB1_SPOKE1_GRE2
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set local-address 172.17.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
crypto map VPN_ISP2 2 ipsec-isakmp
 description VPN between HUB1 and Spoke2 via ISP2
 match address IPSEC_ACL_HUB1_SPOKE2_GRE2
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set local-address 172.17.100.2
 set peer 172.16.2.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 description ISP1
 ip address 172.16.100.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 ip access-group FW_OUTPUT_WAN out
 crypto map VPN_ISP1
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 description ISP2
 ip address 172.17.100.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
```

```
 ip access-group FW_OUTPUT_WAN out
 crypto map VPN_ISP2
 ip inspect INSPECT out
!
interface GigabitEthernet0/2
 ip address 192.168.100.1 255.255.255.0
!
interface Tunnel1
 description GRE between Hub1 and Spoke1 via ISP1
 ip address 10.10.10.1 255.255.255.252
 mtu 1400
 tunnel source 172.16.100.2
 tunnel destination 172.16.1.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
interface Tunnel2
 description GRE between Hub1 and Spoke1 via ISP2
 ip address 10.10.10.5 255.255.255.252
 mtu 1400
 tunnel source 172.17.100.2
 tunnel destination 172.16.1.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
interface Tunnel3
 description GRE between Hub1 and Spoke2 via ISP1
 ip address 10.10.10.9 255.255.255.252
 mtu 1400
 tunnel source 172.16.100.2
 tunnel destination 172.16.2.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
interface Tunnel4
 description GRE between Hub1 and Spoke2 via ISP2
 ip address 10.10.10.13 255.255.255.252
 mtu 1400
 tunnel source 172.17.100.2
 tunnel destination 172.16.2.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
!
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
...
AD4F8901771632E0A0AF83
```

```
quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Wed Jan 17 13:00:10 2024

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Wed Jan 17 13:00:10 2024 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
    TCPSessionsMax = 256000
    TCPStrictnessLevel = 3
    UDPNotEstablishedTimeout = 30
    UDPEstablishedTimeout = 120
    UDPSessionsMax = 65536
)

IKETransform crypto:isakmp:policy:1
(
    CipherAlg   = "GR341215M-K256-CFB-65526"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

ESPProposal MAGMA_ENCRYPT_AND_INTEGRITY:ESP
(
    Transform* = ESPTransform
    (
        CipherAlg*          = "GR341215M-K288-MGM-33"
        LifetimeSeconds     = 3600
    )
)

FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
```

```
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain FilterChain:FW_OUTPUT_WAN (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        ProtocolID = 47
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        SourceIP = 192.168.100.0/24
```

```
        DestinationIP = 192.168.1.0/24
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.2.0/24
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = "INSPECT"
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    ),
    Filter (
        Label = "INSPECT"
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

FilterChain IPv6Default (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

IKEParameters(
```

```
    FragmentSize = 576
    SALifetimeDelta = 50
    InitiatorSessionsMax = 100
    ResponderSessionsMax = 100
)

AuthMethodGOSTSign GOST:Sign:NEW_ALGS
(
    RemoteID             = Remote:NEW_ALGS
    LocalID              = Local:Cert:NEW_ALGS
    SendRequestMode      = ALWAYS
    SendCertMode         = ALWAYS
)

IKERule IKERule:VPN_ISP1:1
(
    IKEPeerIPFilter = 172.16.1.2
    IKELocalIPFilter = 172.16.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 10
)

IKERule IKERule:VPN_ISP1:2
(
    IKEPeerIPFilter = 172.16.2.2
    IKELocalIPFilter = 172.16.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 20
)

IKERule IKERule:VPN_ISP2:1
(
    IKEPeerIPFilter = 172.16.1.2
    IKELocalIPFilter = 172.17.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 30
)

IKERule IKERule:VPN_ISP2:2
(
    IKEPeerIPFilter = 172.16.2.2
    IKELocalIPFilter = 172.17.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
```

```
    DPDRetries          = 5
    Priority            = 40
)

IPsecAction IPsecAction:VPN_ISP1:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.1.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_ISP1:1
)

IPsecAction IPsecAction:VPN_ISP1:2
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.2.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_ISP1:2
)

IPsecAction IPsecAction:VPN_ISP2:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.17.100.2
        PeerAddress = 172.16.1.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_ISP2:1
)

IPsecAction IPsecAction:VPN_ISP2:2
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.17.100.2
        PeerAddress = 172.16.2.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_ISP2:2
)

FilterChain IPsecPolicy:VPN_ISP1 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
```

```
    ),
    Filter (
        SourceIP = 172.16.100.2
        DestinationIP = 172.16.1.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP1:1 >
        LogEventID = "IPsec:Protect:VPN_ISP1:1:IPSEC_ACL_HUB1_SPOKE1_GRE1"
    ),
    Filter (
        Label = "VPN_ISP1:2:IPSEC_ACL_HUB1_SPOKE2_GRE1"
        SourceIP = 172.16.100.2
        DestinationIP = 172.16.2.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP1:2 >
        LogEventID = "IPsec:Protect:VPN_ISP1:2:IPSEC_ACL_HUB1_SPOKE2_GRE1"
    )
)

FilterChain IPsecPolicy:VPN_ISP2 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 172.17.100.2
        DestinationIP = 172.16.1.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP2:1 >
        LogEventID = "IPsec:Protect:VPN_ISP2:1:IPSEC_ACL_HUB1_SPOKE1_GRE2"
    ),
    Filter (
        Label = "VPN_ISP2:2:IPSEC_ACL_HUB1_SPOKE2_GRE2"
        SourceIP = 172.17.100.2
        DestinationIP = 172.16.2.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP2:2 >
        LogEventID = "IPsec:Protect:VPN_ISP2:2:IPSEC_ACL_HUB1_SPOKE2_GRE2"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = FilterChain:FW_OUTPUT_WAN
    IPsecPolicy = IPsecPolicy:VPN_ISP1
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = FilterChain:FW_OUTPUT_WAN
    IPsecPolicy = IPsecPolicy:VPN_ISP2
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
```

```
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel4"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

3. Содержимое `/etc/multiwan/config.yaml`:

```
config_format_version: 1
links_params:
  check_links_reachability_on_start: true
  links:
    -
      iface_name: eth0
      name: ISP1
      prio: 200
      routes:
        ipv4:
          networks:
            - 0.0.0.0/0
          nexthop: 172.16.100.1
      check:
        hosts_reachability: any
        ipv4:
          hosts:
            - 172.16.101.15
        method:
          icmp:
            interval: 1
            restore_interval: 1
            restore_count: 3
            failure_count: 3

    -
      iface_name: eth1
      name: ISP2
      prio: 100
      routes:
        ipv4:
          networks:
            - 0.0.0.0/0
          nexthop: 172.17.100.1
      check:
```

```
        hosts_reachability: any
        ipv4:
          hosts:
            - 172.16.101.15
        method:
          icmp:
            interval: 1
            restore_interval: 1
            restore_count: 3
            failure_count: 3
```

4. Конфигурации FRR:

```
frr version 9.0.2
frr defaults traditional
hostname Hub1
log syslog informational
service integrated-vtysh-config
!
router bgp 65000
 bgp router-id 172.16.100.2
 timers bgp 2 8
 neighbor 10.10.10.2 remote-as 65000
 neighbor 10.10.10.6 remote-as 65000
 neighbor 10.10.10.10 remote-as 65000
 neighbor 10.10.10.14 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.100.0/24
 exit-address-family
exit
!
```

5. Содержимое `/etc/network/pbr_common`:

```
SPOKES_IP_ADDRESSES="172.16.1.2 172.16.2.2"

WAN_0_INTERFACE="eth0"
WAN_0_IP_ADDRESS="172.16.100.2"
WAN_0_NEXTHOP_ADDRESS="172.16.100.1"

WAN_1_INTERFACE="eth1"
WAN_1_IP_ADDRESS="172.17.100.2"
WAN_1_NEXTHOP_ADDRESS="172.17.100.1"

GRE1_TABLE_ID="100"
GRE2_TABLE_ID="101"
PRIO="100"
```

6. Содержимое `/etc/network/if-pre-up.d/pbr`:

```
#!/bin/bash
# if-pre-up script

source /etc/network/pbr_common

if [[ $IFACE == $WAN_0_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_0="from $WAN_0_IP_ADDRESS to $SPOKE_IP lookup $GRE1_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_0 2>/dev/null || true
        ip rule add $RULE_WLAN_0
    done
fi

if [[ $IFACE == $WAN_1_INTERFACE ]]; then
```

```
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_1="from $WAN_1_IP_ADDRESS to $SPOKE_IP lookup $GRE2_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_1 2>/dev/null || true
        ip rule add $RULE_WLAN_1
    done
fi

exit 0
```

7. Содержимое `/etc/network/if-up.d/pbr`:

```
#!/bin/bash
# if-up script

source /etc/network/pbr_common

if [[ $IFACE == $WAN_0_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        ip route add "$SPOKE_IP/32" via "$WAN_0_NEXTHOP_ADDRESS" dev "$WAN_0_INTERFACE"
table $GRE1_TABLE_ID
    done
fi

if [[ $IFACE == $WAN_1_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        ip route add "$SPOKE_IP/32" via "$WAN_1_NEXTHOP_ADDRESS" dev "$WAN_1_INTERFACE"
table $GRE2_TABLE_ID
    done
fi

exit 0
```

8. Содержимое `/etc/network/if-post-down.d/pbr`:

```
#!/bin/bash
# if-post-down script

source /etc/network/pbr_common

if [[ $IFACE == $WAN_0_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_0="from $WAN_0_IP_ADDRESS to $SPOKE_IP lookup $GRE1_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_0 2>/dev/null || true
    done
fi

if [[ $IFACE == $WAN_1_INTERFACE ]]; then
    for SPOKE_IP in $SPOKES_IP_ADDRESSES
    do
        RULE_WLAN_1="from $WAN_1_IP_ADDRESS to $SPOKE_IP lookup $GRE2_TABLE_ID pref
$PRIO"
        ip rule del $RULE_WLAN_1 2>/dev/null || true
    done
fi

exit 0
```

Конфигурации криптошлюза Spoke1

1. 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
aaa new-model
!
!
hostname Spoke1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.1.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE1
 permit gre host 172.16.1.2 host 172.16.100.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE1_GRE2
 permit gre host 172.16.1.2 host 172.17.100.2
!
ip access-list extended FW_OUTPUT_WAN
 deny   gre any any
 deny   ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
 permit ip any any
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
```

```
!
crypto map VPN 1 ipsec-isakmp
 description VPN between Spoke1 and HUB1 via ISP1
 match address IPSEC_ACL_HUB1_SPOKE1_GRE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set peer 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
crypto map VPN 2 ipsec-isakmp
 description VPN between Spoke1 and HUB1 via ISP2
 match address IPSEC_ACL_HUB1_SPOKE1_GRE2
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set peer 172.17.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.1.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 ip access-group FW_OUTPUT_WAN out
 crypto map VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
!
interface Tunnel1
 description GRE between Spoke1 and Hub1 via ISP1
 ip address 10.10.10.2 255.255.255.252
 mtu 1400
 tunnel source 172.16.1.2
 tunnel destination 172.16.100.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
interface Tunnel2
 description GRE between Spoke1 and Hub1 via ISP2
 ip address 10.10.10.6 255.255.255.252
 mtu 1400
 tunnel source 172.16.1.2
 tunnel destination 172.17.100.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 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 crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
...
AD4F8901771632E0A0AF83

quit
```

```
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Wed Jan 17 13:01:52 2024

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Wed Jan 17 13:01:52 2024 (user: cscons)"
    Version                       = LSP_5_0
    CRLHandlingMode               = ENABLE
    PreserveIPsecSA               = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 172.16.1.1
        )

)

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
    TCPSessionsMax = 256000
    TCPStrictnessLevel = 3
    UDPNotEstablishedTimeout = 30
    UDPEstablishedTimeout = 120
    UDPSessionsMax = 65536
)

IKETransform crypto:isakmp:policy:1
(
    CipherAlg   = "GR341215M-K256-CFB-65526"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

ESPProposal MAGMA_ENCRYPT_AND_INTEGRITY:ESP
(
    Transform* = ESPTransform
    (
        CipherAlg*          = "GR341215M-K288-MGM-33"
        LifetimeSeconds     = 3600
    )
)

FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
```

```
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain FilterChain:FW_OUTPUT_WAN (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        ProtocolID = 47
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        SourceIP = 192.168.1.0/24
        DestinationIP = 192.168.100.0/24
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = "INSPECT"
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    ),
    Filter (
        Label = "INSPECT"
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
```

```
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

FilterChain IPv6Default (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

IKEParameters(
    FragmentSize = 576
    SALifetimeDelta = 50
    InitiatorSessionsMax = 100
    ResponderSessionsMax = 100
)

AuthMethodGOSTSign GOST:Sign:NEW_ALGS
(
    RemoteID             = Remote:NEW_ALGS
    LocalID              = Local:Cert:NEW_ALGS
    SendRequestMode      = ALWAYS
    SendCertMode         = ALWAYS
)

IKERule IKERule:VPN:1
(
    IKEPeerIPFilter = 172.16.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 10
)
```

```
IKERule IKERule:VPN:2
(
    IKEPeerIPFilter = 172.17.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 20
)

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:1
)

IPsecAction IPsecAction:VPN:2
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.17.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:2
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 172.16.1.2
        DestinationIP = 172.16.100.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_SPOKE1_GRE1"
    ),
    Filter (
        Label = "VPN:2:IPSEC_ACL_HUB1_SPOKE1_GRE2"
        SourceIP = 172.16.1.2
        DestinationIP = 172.17.100.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2 >
        LogEventID = "IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_SPOKE1_GRE2"
    )
)
```

```
NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = FilterChain:FW_OUTPUT_WAN
    IPsecPolicy = IPsecPolicy:VPN
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

3. Конфигурация FRR:

```
frr version 9.0.2
frr defaults traditional
hostname Spoke1
log syslog informational
service integrated-vtysh-config
!
router bgp 65000
 bgp router-id 172.16.1.2
 timers bgp 2 8
 neighbor 10.10.10.1 remote-as 65000
 neighbor 10.10.10.5 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.1.0/24
 exit-address-family
exit
!
```

Конфигурации криптошлюза Spoke2

1. Конфигурация 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
aaa new-model
```

```
!
!
hostname Spoke2
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.2.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE2_GRE1
 permit gre host 172.16.2.2 host 172.16.100.2
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE2_GRE2
 permit gre host 172.16.2.2 host 172.17.100.2
!
ip access-list extended FW_OUTPUT_WAN
 deny   gre any any
 deny   ip 192.168.2.0 0.0.0.255 192.168.100.0 0.0.0.255
 permit ip any any
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN 1 ipsec-isakmp
 description VPN between Spoke2 and HUB1 via ISP1
 match address IPSEC_ACL_HUB1_SPOKE2_GRE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set peer 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
crypto map VPN 2 ipsec-isakmp
```

```
 description VPN between Spoke2 and HUB1 via ISP2
 match address IPSEC_ACL_HUB1_SPOKE2_GRE2
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set peer 172.17.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.2.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 ip access-group FW_OUTPUT_WAN out
 crypto map VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.2.1 255.255.255.0
!
interface Tunnel1
 description GRE between Spoke2 and Hub1 via ISP1
 ip address 10.10.10.10 255.255.255.252
 mtu 1400
 tunnel source 172.16.2.2
 tunnel destination 172.16.100.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
interface Tunnel2
 description GRE between Spoke2 and Hub1 via ISP2
 ip address 10.10.10.14 255.255.255.252
 mtu 1400
 tunnel source 172.16.2.2
 tunnel destination 172.17.100.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl 255
 tunnel tos inherit
!
!
ip route 0.0.0.0 0.0.0.0 172.16.2.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
```

```
#   Conversion Date/Time:   Wed Jan 17 13:03:32 2024

GlobalParameters(
    Title                     = "This LSP was automatically generated by CSP Converter
at Wed Jan 17 13:03:32 2024 (user: cscons)"
    Version                     = LSP_5_0
    CRLHandlingMode             = ENABLE
    PreserveIPsecSA             = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 172.16.2.1
        )

)

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
    TCPSessionsMax = 256000
    TCPStrictnessLevel = 3
    UDPNotEstablishedTimeout = 30
    UDPEstablishedTimeout = 120
    UDPSessionsMax = 65536
)

IKETransform crypto:isakmp:policy:1
(
    CipherAlg   = "GR341215M-K256-CFB-65526"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

ESPProposal MAGMA_ENCRYPT_AND_INTEGRITY:ESP
(
    Transform* = ESPTransform
    (
        CipherAlg*          = "GR341215M-K288-MGM-33"
        LifetimeSeconds     = 3600
    )
)

FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
        DestinationIP = 172.16.2.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
```

```
    Filter (
        DestinationIP = 172.16.2.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.2.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.2.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain FilterChain:FW_OUTPUT_WAN (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        ProtocolID = 47
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.100.0/24
        Action = DROP
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = "INSPECT"
        LogEventID = "FW_OUTPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    ),
    Filter (
        Label = "INSPECT"
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
```

```
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

FilterChain IPv6Default (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

IKEParameters(
    FragmentSize = 576
    SALifetimeDelta = 50
    InitiatorSessionsMax = 100
    ResponderSessionsMax = 100
)

AuthMethodGOSTSign GOST:Sign:NEW_ALGS
(
    RemoteID             = Remote:NEW_ALGS
    LocalID              = Local:Cert:NEW_ALGS
    SendRequestMode      = ALWAYS
    SendCertMode         = ALWAYS
)

IKERule IKERule:VPN:1
(
    IKEPeerIPFilter = 172.16.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 10
)

IKERule IKERule:VPN:2
(
    IKEPeerIPFilter = 172.17.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
```

```
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 20
)

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:1
)

IPsecAction IPsecAction:VPN:2
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.17.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:2
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 172.16.2.2
        DestinationIP = 172.16.100.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_SPOKE2_GRE1"
    ),
    Filter (
        Label = "VPN:2:IPSEC_ACL_HUB1_SPOKE2_GRE2"
        SourceIP = 172.16.2.2
        DestinationIP = 172.17.100.2
        ProtocolID = 47
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2 >
        LogEventID = "IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_SPOKE2_GRE2"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = FilterChain:FW_OUTPUT_WAN
    IPsecPolicy = IPsecPolicy:VPN
```

```
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "Tunnel2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

3. Конфигурация FRR:

```
frr version 9.0.2
frr defaults traditional
hostname Spoke2
log syslog informational
service integrated-vtysh-config
!
router bgp 65000
 bgp router-id 172.16.2.2
 timers bgp 2 8
 neighbor 10.10.10.9 remote-as 65000
 neighbor 10.10.10.13 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.2.0/24
 exit-address-family
exit
!
```