Построение VPN соединения при наличии подсетей с пересекающимся адресным пространством и отказоустойчивостью

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 описывает настройку криптошлюзов «С-Терра Шлюз» для обеспечения безопасного взаимодействия между центральным офисом, филиалами и удаленными VPN клиентами в условиях наличия в центральном офисе и филиалов нескольких подсетей, чье адресное пространство частично или полностью пересекается.

Защищаемые подсети в LAN сегменте и недоверенные подсети в WAN сегменте – IPv4. Обеспечение безопасного взаимодействия достигается путем шифрования и туннелирования трафика с применением отечественных отраслевых стандартов ГОСТ и протокола IPsec.

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, VPN клиент – «С-Терра Клиент» 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, VRF, 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 https://doc.s-terra.ru раздел С-Терра Шлюз -> С-Терра Шлюз 5.0 -> «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

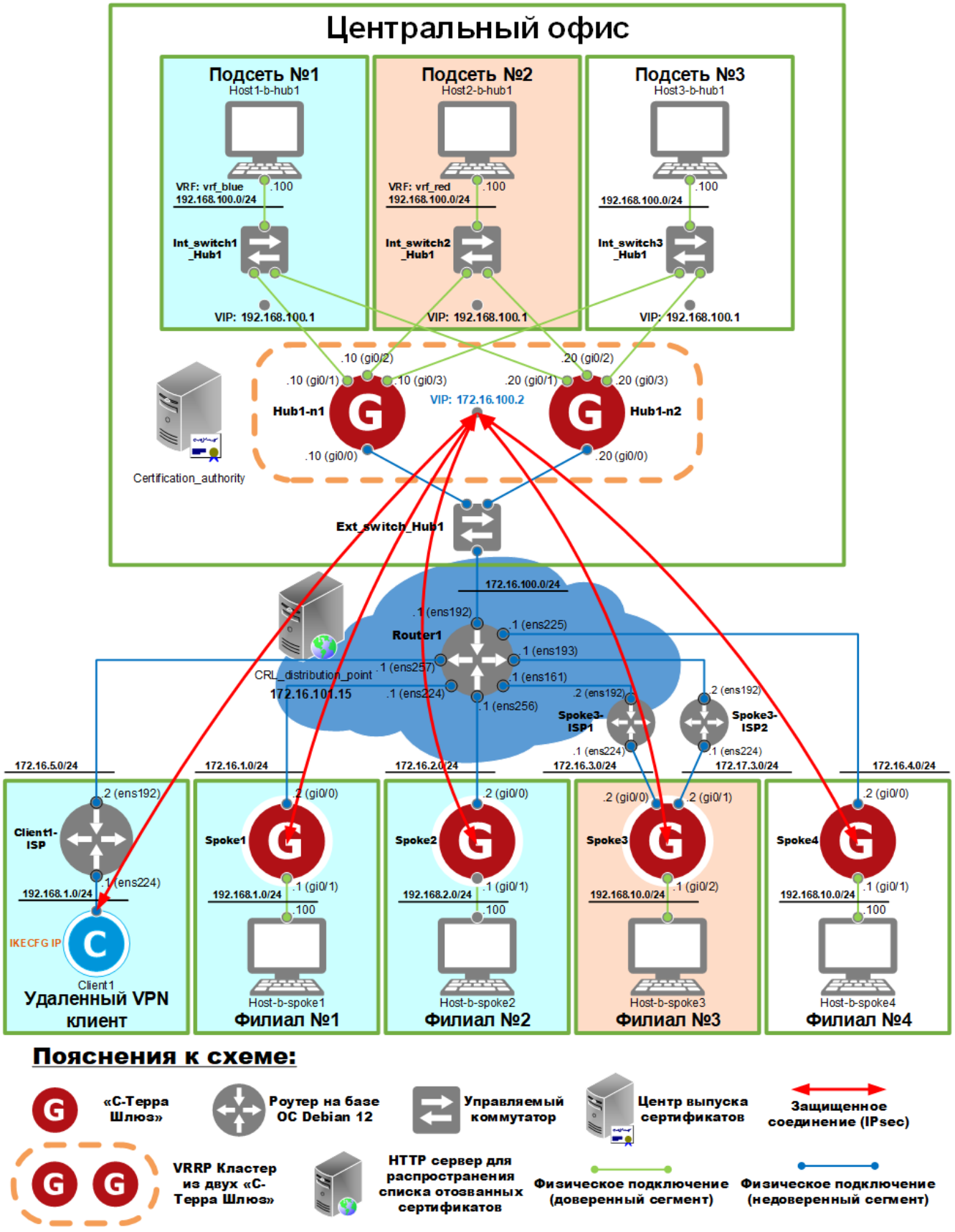


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), криптошлюзы «С-Терра Шлюз» (Hub1-n1 и Hub1-2), коммутатор (Ext-switch1-Hub1), а также:

      1.1.1 В подсети №1 размещаются: коммутатор (Int_switch1_Hub1) и персональный компьютер (Host1-b-hub1).

      1.1.2 В подсети №2 размещаются: коммутатор (Int_switch2_Hub1) и персональный компьютер (Host2-b-hub1).

      1.1.3 В подсети №3 размещаются: коммутатор (Int_switch3_Hub1) и персональный компьютер (Host3-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 филиале №3 размещаются: криптошлюз «С-Терра Шлюз» (Spoke3) и персональный компьютер (Host-b-spoke3).

   1.5. В филиале №4 размещаются: криптошлюз «С-Терра Шлюз» (Spoke4) и персональный компьютер (Host-b-spoke4).

   1.6. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point), маршрутизаторы (Router1, Client1-ISP, Spoke3-ISP1 и Spoke3-ISP2) и персональный компьютер (Client1).

2. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизатор Router1.

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

   2.1. Криптошлюзы Hub1-n1 и Hub1-n2 подключаются к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router1).

   2.2. Криптошлюзы Spoke1, Spoke2 и Spoke4 подключаются к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router1).

   2.3. Криптошлюз Spoke3 подключается к сети Интернет через двух провайдеров (Spoke3-ISP1 и Spoke3-ISP2) с использованием технологии `multiwan`.

3. Пересекающиеся подсети.

   В центральном офисе находятся три подсети с одинаковым адресным пространством – 192.168.100.0/24. Изолирование этих подсетей осуществляется за счет использования технологии VRF (Virtual routing and forwarding) на центральных шлюзах.

   Интерфейс, подключенный к подсети №1, находится в VRF инстансе с именем vrf_blue, а интерфейс, подключенный к подсети №2, находится в VRF инстансе с именем vrf_red. Каждый VRF инстанс использует свою собственную таблицу маршрутизации. В данном сценарии vrf_blue использует таблицу маршрутизации под номером 10, vrf_red – под номером 20.

   В таблицах маршрутизации каждого VRF инстанса присутствует unreachable маршрут по умолчанию. Делается это для того, чтобы трафик, для которого в данном VRF инстансе отсутствует подходящий маршрут, не маршрутизировался в соответствии с другими таблицами и не покидал данный VRF инстанс.

   Интерфейс, подключенный к подсети №3 не входит в какой-либо VRF инстанс, и трафик этой подсети маршрутизируется согласно основной (main) таблице маршрутизации (номер 254).

В силу особенностей настройки маршрутизации, среди всех подсетей центрального офиса выход в интернет можно настроить только для хостов из подсети №3.

Адресные пространства подсетей филиалов пересекаются у филиалов №3 и №4. VRF на филиальных шлюзах не используется.

Допускается пересечение адресных пространств между подсетями других филиалов, но со следующим ограничением: нельзя, чтобы две или более подсетей филиалов с пересекающимся адресным пространством взаимодействовали с одними и теми же подсетями центрального офиса (то есть с подсетями одного VRF инстанса или с одними и теми же подсетями, не входящими в VRF инстанс).

4. Взаимодействие между центральным офисом и филиалами.

   В данном сценарии описывается настройка для обеспечения безопасного взаимодействия в следующих направлениях:

   4.1. Из филиала №1, филиала №2 и с удаленного VPN Клиента Client1 до подсети №1 центрального офиса (и в обратном направлении);

   4.2. Из филиала №3 до подсети №2 центрального офиса (и в обратном направлении);

   4.3. Из филиала №4 до подсети №3 центрального офиса (и в обратном направлении).

   Другие взаимодействия между подсетями центрального офиса и филиалов не предусматриваются. Например, нельзя попасть из подсети филиала №1 в подсеть №2 центрального офиса, или из подсети №3 центрального офиса в подсети филиалов №1 и №2.

5. VTI интерфейсы.

   Для построения защищенного соединения между центром и филиальными криптошлюзами Spoke1, Spoke2 и Spoke3, а также до удаленного VPN клиента используются виртуальные интерфейсы (в данном сценарии они называются VTI интерфейсами), к которым прикрепляются криптокарты. На центральных криптошлюзах VTI интерфейсов несколько – по одному для каждого из партнеров. Криптокарты на VTI интерфейсах в данном сценарии содержат максимально широкие ACL (permit ip any any) для того, чтобы минимизировать зависимость настроек шифрования от настроек адресации подсетей.

   Для защищенного соединения между центром и криптошлюзом Spoke4 VTI интерфейсы не используются, и конфигурация этого соединения соответствует обычному Site-to-Site сценарию.

6. Динамическая маршрутизация.

   Для обмена информацией о подсетях за шлюзами используется протокол динамической маршрутизации BGP. В качестве адресов BGP пиров используются адреса VTI интерфейсов.

7. Отказоустойчивый VRRP кластер.

   Криптошлюзы Hub1-n1 и Hub1-n2 объединяются в отказоустойчивый кластер, работающий на базе протокола VRRP в режиме Master/Backup. За реализацию протокола VRRP отвечает ПО «keepalived».

   Криптошлюзы Hub1-n1 и Hub1-n2 являются нодами кластера.

   Конфигурация между нодами кластера не синхронизируется. При изменении настроек на одной ноде нужно выполнить аналогичные настройки на другой.

   Нода, находящаяся в состоянии Master, владеет VIP адресами в доверенном и недоверенном сегментах и, соответственно, обрабатывает трафик. Нода, находящаяся в состоянии Backup или Fault, не владеет VIP адресами. По умолчанию все VRRP интерфейсы синхронизированы в группу номер ноль. Соответственно, одновременно меняют свое состояние.

   На нодах кластера выключен режим preempt mode – это значит, что нода с более высоким приоритетом не будет захватывать роль Master у рабочей ноды с более низким приоритетом. Данная настройка помогает избавиться от лишних переключений.

8. Source NAT.

Source NAT выполняется в данном сценарии на нодах кластера в центральном офисе. Трансляция адресов осуществляется в кластерный VIP адрес 172.16.100.100. Source NAT не выполняется для трафика, принадлежащего VRF инстансам.

В общем случае предполагается, что HTTP сервер, распространяющий список отзыва сертификатов, располагается в Интернет. С учетом того, что требуется обеспечить периодическое обновление списка отзыва сертификатов на резервной ноде кластера, нужно, чтобы она имела вход в Интернет (примечание: нода, находящаяся в состоянии Backup или Fault, выходит в Интернет через ноду, которая находится в состоянии Master). Следовательно, на нодах должен быть настроен source NAT. Настройка осуществляется при помощи утилит `iptables` и `netfilter-persistent`, входящих в состав ОС.

Source NAT настраивается для неподлежащего шифрованию трафика из подсети №3 и для транзитного IPsec трафика (не принадлежащего какому либо VRF инстансу).

9. Параметры межсетевого экранирования.

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

   9.1. На LAN интерфейсах IPv4 трафик не контролируется.

   9.2. На LAN интерфейсах IPv6 трафик заблокирован.

   9.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

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

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

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

      9.3.4 Протокол VRRP (номер 112).

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

   9.4. На WAN интерфейсах IPv6 трафик заблокирован.

   9.5. 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).

10. Параметры шифрования IPsec.

    Весь IPv4 трафик между подсетями центрального офиса (подсеть №1, №2 и №3) и филиалами (192.168.1.0/24, 192.168.2.0/24, 192.168.10.0/24 (филиал №3) и 192.168.10.0/24 (филиал №4)) и удаленным VPN клиентом защищается с использованием алгоритмов ГОСТ и протокола IPsec в туннельном IPv4 режиме.

    Инициировать защищенное соединение между криптошлюзами центрального офиса и филиалов может трафик, идущий как из центра в филиалы, так и в обратном направлении.

    Инициировать защищенное соединение между криптошлюзами центрального офиса и удаленным VPN клиентом может только трафик от VPN клиента в центр.

    10.1. Параметры протокола IKE:

    - Аутентификация при помощи цифровых сертификатов, алгоритм подписи – GOST R 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 GOST R 34.10-2012 (ключ 256 бит).

    10.2. 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 Router1

1. Настройте IP адрес – 172.16.100.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.16.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Настройте IP адрес – 172.16.2.1 и маску – 255.255.255.0 на сетевом интерфейсе ens256.
4. Настройте IP адрес – 172.16.255.1 и маску – 255.255.255.0 на сетевом интерфейсе ens161.
5. Настройте IP адрес – 172.17.255.1 и маску – 255.255.255.0 на сетевом интерфейсе ens193.
6. Настройте IP адрес – 172.16.4.1 и маску – 255.255.255.0 на сетевом интерфейсе ens225.
7. Настройте IP адрес – 172.16.5.1 и маску – 255.255.255.0 на сетевом интерфейсе ens257.
8. Задайте маршруты до подсетей 172.16.3.0/24 и 172.17.3.0/24 через 172.16.255.2 и 172.17.255.2 соответственно.
9. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Client1-ISP

1. Настройте IP адрес – 172.16.5.2 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 192.168.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Задайте маршрут по умолчанию через 172.16.5.1.
4. Настройте выдачу IP адресов по DHCP для устройств из подсети 192.168.1.0/24.
5. Настройте Source NAT на сетевом интерфейсе ens192 для трафика из подсети 192.168.1.0/24.
6. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Spoke3-ISP1

1. Настройте IP адрес – 172.16.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.16.3.1 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 Spoke3-ISP2

1. Настройте IP адрес – 172.17.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.17.3.1 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 Host1-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 Host2-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 Host3-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 устройства Host-b-spoke3

1. Настройте IP адрес – 192.168.10.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.10.1.
3. Разрешите прием и отправку ICMP пакетов.

Настройка устройства Host-b-spoke4

1. Настройте IP адрес – 192.168.10.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.10.1.
3. Разрешите прием и отправку ICMP пакетов.

Настройка криптошлюза Hub1-n1

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. Доверенным каналом связи также считается канал, защищенный при помощи протокола IPsec.

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 разделе «Начальные настройки» базового сценария.

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 разделе «Настройки PKI (запросы и сертификаты)» базового сценария.

Настройка сетевых параметров

1. Войдите в cisco-like консоль из CLI разграничения доступа:

```
administrator@sterragate] configure
sterragate login: cscons
```

```
Password:
S-Terra Gate 5.0.XXXXX (amd64)
sterragate#
```

2. Войдите в режим конфигурации:

```
sterragate#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

3. Задайте имя устройства:

```
sterragate(config)#hostname Hub1-n1
```

4. Включите внешний (`GigabitEthernet0/0`) и внутренние (`GigabitEthernet0/1`, `GigabitEthernet0/2` и `GigabitEthernet0/3`) интерфейсы:

```
Hub1-n1(config)#interface range gigabitEthernet 0/0 - 3
Hub1-n1(config-if-range)#no shutdown
Hub1-n1(config-if-range)#exit
```

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

```
Hub1-n1(config)#do show interfaces
```

```
GigabitEthernet0/0 is up, line protocol is up
  Hardware address is 0050.569e.94c4
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.7f79
  MTU 1500 bytes
...
```

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

6. Настройте IP адреса на внешнем и внутренних интерфейсах в соответствии со схемой стенда:

```
Hub1-n1(config)#interface gigabitEthernet 0/0
Hub1-n1(config-if)#ip address 172.16.100.10 255.255.255.0
Hub1-n1(config-if)#exit
Hub1-n1(config)#interface gigabitEthernet 0/1
Hub1-n1(config-if)#description VRF BLUE LAN
Hub1-n1(config-if)#ip address 192.168.100.10 255.255.255.0
Hub1-n1(config-if)#exit
Hub1-n1(config)#interface gigabitEthernet 0/2
Hub1-n1(config-if)#description VRF RED LAN
Hub1-n1(config-if)#ip address 192.168.100.10 255.255.255.0
Hub1-n1(config-if)#exit
Hub1-n1(config)#interface gigabitEthernet 0/3
Hub1-n1(config-if)#ip address 192.168.100.10 255.255.255.0
Hub1-n1(config-if)#exit
```

7. Задайте маршрут по умолчанию через собственный IP адрес LAN интерфейса `gigabitEthernet0/3` шлюза Hub1-n2 с метрикой 100:

```
Hub1-n1(config)#ip route 0.0.0.0 0.0.0.0 192.168.100.20 100
```

8. Проверьте доступность устройства Router1:

```
Hub1-n1(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=0.503 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.340 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.512 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.449 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.414 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.340/0.443/0.512/0.063 ms
```

9. Сохраните текущую конфигурацию и выйдите из cisco-like консоли:

```
Hub1-n1(config)#end
Hub1-n1#exit
```

```
administrator@Hub1-n1]
```

10. Перейдите в linux консоль:

```
administrator@Hub1-n1] system
```

```
Entering system shell...
root@Hub1-n1:~#
```

11. Создайте VTI и VRF интерфейсы. Настройка осуществляется средствами ОС Linux.

Важно! Интерфейсы, которые входят в VRF, должны существовать в системе на момент включения VRF интерфейса, в противном случае VRF инстанс будет сконфигурирован неправильно.
По умолчанию, интерфейсы, описанные в конфигурационных файлах в директории `/etc/network/interfaces.d/`, включаются **до** интерфейсов, описанных в `/etc/network/interfaces`.
В связи с этим, если в VRF требуется добавить интерфейсы (например, VLAN или GRE), которые создаются в конфигурационном файле `/etc/network/interfaces` и которые еще не существуют на момент включения соответствующего VRF интерфейса, то следует либо перенести описание VRF интерфейса в файл `/etc/network/interfaces` (после секции, обозначенной строками `###netifcfg-begin###` и `###netifcfg-end###`), либо в файле `/etc/network/interfaces` перенести строку `source /etc/network/interfaces.d/*` в конец файла (после строки `###netifcfg-end###`).

11.1. Создайте файл `/etc/network/interfaces.d/st_vti0`, содержащий описание VTI интерфейса st_vti0 (этот интерфейс будет использоваться для построения соединения с удаленным VPN клиентом Client1):

> Имя создаваемого VTI интерфейса должно быть вида st_vtiN, в противном случае входящие IPsec пакеты на таком интерфейсе будут отброшены VPN драйвером.
> Интерфейсы, описанные в конфигурационных файлах в директории `/etc/network/interfaces.d/` будут включаться в лексикографическом порядке. Имена для файлов с описаниями интерфейсов в этой директории следует выбирать так, чтобы сначала включались VTI интерфейсы, а после них – VRF интерфейсы. В данном сценарии файлы с описанием VTI интерфейсов имеют названия вида st_vtiN, файлы с описанием VRF интерфейсов имеют названия вида vrf_xxx, что соответствует этому правилу.

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

11.2. Создайте файл `/etc/network/interfaces.d/st_vti1`, содержащий описание VTI интерфейса st_vti1 (этот интерфейс будет использоваться для построения соединения с криптошлюзом Spoke1):

```
auto st_vti1
iface st_vti1 inet static
address 10.10.11.1/24
pre-up ip link del st_vti1 type dummy 2>/dev/null || true
pre-up ip link add st_vti1 type dummy
post-down ip link set dev st_vti1 down
```

11.3. Создайте файл `/etc/network/interfaces.d/st_vti2`, содержащий описание VTI интерфейса st_vti2 (этот интерфейс будет использоваться для построения соединения с криптошлюзом Spoke2):

```
auto st_vti2
iface st_vti2 inet static
address 10.10.12.1/24
pre-up ip link del st_vti2 type dummy 2>/dev/null || true
pre-up ip link add st_vti2 type dummy
post-down ip link set dev st_vti2 down
```

11.4. Создайте файл `/etc/network/interfaces.d/st_vti3`, содержащий описание VTI интерфейса st_vti3 (этот интерфейс будет использоваться для построения соединения с криптошлюзом Spoke3):

```
auto st_vti3
iface st_vti3 inet static
address 10.10.13.1/24
pre-up ip link del st_vti3 type dummy 2>/dev/null || true
pre-up ip link add st_vti3 type dummy
post-down ip link set dev st_vti3 down
```

11.5. Создайте файл `/etc/network/interfaces.d/vrf_blue`, содержащий описание VRF интерфейса vrf_blue (этот интерфейс формирует VRF инстанс `vrf_blue`), и внесите в него следующие строки:

11.5.1 Задайте для интерфейса имя `vrf_blue`:

```
auto vrf_blue
iface vrf_blue inet manual
```

11.5.2 Настройте удаление интерфейса в pre-up скрипте:

```
pre-up ip link del vrf_blue type vrf 2>/dev/null || true
```

11.5.3 Настройте очистку таблицы маршрутизации под номером 10, которая будет использоваться для маршрутизации в данном VRF инстансе:

```
pre-up ip route flush table 10 2>/dev/null || true
```

11.5.4 Укажите для интерфейса тип `vrf` и таблицу маршрутизации 10:

```
pre-up ip link add vrf_blue type vrf table 10
pre-up ip link set dev vrf_blue up
```

11.5.5 Задайте unreachable маршрут по умолчанию с метрикой 4278198272 (чтобы трафик, для которого в данном VRF инстансе отсутствуют маршруты, не покидал VRF инстанс):

```
pre-up ip route add table 10 unreachable default metric 4278198272
```

11.5.6 Укажите какие интерфейсы будут принадлежать данному VRF инстансу:

```
pre-up ip link set dev eth1 master vrf_blue
pre-up ip link set dev st_vti0 master vrf_blue
pre-up ip link set dev st_vti1 master vrf_blue
pre-up ip link set dev st_vti2 master vrf_blue
post-down ip link set dev vrf_blue down
```

11.6. Создайте файл `/etc/network/interfaces.d/vrf_red`, содержащий описание VRF интерфейса vrf_red (этот интерфейс формирует VRF инстанс `vrf_red`):

```
auto vrf_red
iface vrf_red inet manual
pre-up ip link del vrf_red type vrf 2>/dev/null || true
pre-up ip route flush table 20 2>/dev/null || true
pre-up ip link add vrf_red type vrf table 20
pre-up ip link set dev vrf_red up
pre-up ip route add table 20 unreachable default metric 4278198272
pre-up ip link set dev eth2 master vrf_red
pre-up ip link set dev st_vti3 master vrf_red
post-down ip link set dev vrf_red down
```

12. Отредактируйте файл `/etc/ifaliases.cf`, добавив туда строки для VTI интерфейсов:

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="GigabitEthernet0/3" pattern="eth3")
interface (name="GigabitEthernet0/4" pattern="eth4")
interface (name="stVTI0" pattern="st_vti0")
interface (name="stVTI1" pattern="st_vti1")
interface (name="stVTI2" pattern="st_vti2")
interface (name="stVTI3" pattern="st_vti3")
interface (name="default" pattern="*")
```

13. Пересчитайте контрольную сумму для файла `/etc/ifaliases.cf`:

```
root@Hub1-n1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```
SUCCESS:  Operation was successful.
```

14. Перезагрузите шлюз, чтобы все изменения вступили в силу.

Настройка шифрования

1. Войдите в cisco-like консоль из CLI разграничения доступа:

```
administrator@Hub1-n1] configure
```

```
Hub1-n1 login: cscons
Password:
S-Terra Gate 5.0.XXXXX (amd64)
```

```
Hub1-n1#
```

2. Войдите в режим конфигурации:

```
Hub1-n1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1-n1(config)#
```

3. Создайте IKECFG пул адресов, из которого будут назначаться IKECFG адреса для удаленных VPN клиентов:

```
Hub1-n1(config)# ip local pool IKECFG_POOL 192.168.255.1 192.168.255.254
```

4. Настройте параметры IKE.

   4.1. Задайте общие параметры IKE:

```
Hub1-n1(config)#crypto isakmp identity dn
Hub1-n1(config)#crypto isakmp keepalive 3 2
Hub1-n1(config)#crypto isakmp keepalive retry-count 5
Hub1-n1(config)#crypto isakmp fragmentation
Hub1-n1(config)#crypto isakmp security-association lifetime delta 50
Hub1-n1(config)#crypto isakmp initiator-sessions-max 100
Hub1-n1(config)#crypto isakmp responder-sessions-max 100
```

   4.2. Создайте политику, описывающую параметры IKE туннеля:

```
Hub1-n1(config)#crypto isakmp policy 1
Hub1-n1(config-isakmp)#encryption magma
Hub1-n1(config-isakmp)#hash stribog-256
Hub1-n1(config-isakmp)#authentication gost-sig
Hub1-n1(config-isakmp)#group vko2
Hub1-n1(config-isakmp)#exit
```

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

```
Hub1-n1(config)#crypto isakmp profile NEW_ALGS
Hub1-n1(conf-isa-prof)#match identity dn
Hub1-n1(conf-isa-prof)#set identity mapping
Hub1-n1(conf-isa-prof)#set policy 1
```

5. Настройте параметры IPsec.

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

```
Hub1-n1(config)#crypto ipsec security-association lifetime kilobytes unlimited
```

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

```
Hub1-n1(config)# crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
Hub1-n1(cfg-crypto-trans)#exit
```

   5.3. Создайте список доступа (ACL) для трафика, который нужно защищать между центральным шлюзом и шлюзом Spoke1:

```
Hub1-n1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE1
Hub1-n1(config-ext-nacl)#permit ip any any
Hub1-n1(config-ext-nacl)#exit
```

   5.4. Создайте список доступа (ACL) для трафика, который нужно защищать между центральным шлюзом и шлюзом Spoke2:

```
Hub1-n1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE2
Hub1-n1(config-ext-nacl)#permit ip any any
Hub1-n1(config-ext-nacl)#exit
```

   5.5. Создайте список доступа (ACL) для трафика, который нужно защищать между центральным шлюзом и шлюзом Spoke3:

```
Hub1-n1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE3
Hub1-n1(config-ext-nacl)#permit ip any any
Hub1-n1(config-ext-nacl)#exit
```

5.6. Создайте список доступа (ACL) для трафика, который нужно защищать между центральным шлюзом и шлюзом Spoke4:

```
Hub1-n1(config)#ip access-list extended IPSEC_ACL_HUB1_SPOKE4
Hub1-n1(config-ext-nacl)#permit ip 192.168.100.0 0.0.0.255 192.168.10.0 0.0.0.255
Hub1-n1(config-ext-nacl)#exit
```

5.7. Создайте список доступа (ACL) для трафика, который нужно защищать между центральным шлюзом и удаленными VPN клиентами:

```
Hub1-n1(config)#ip access-list extended IPSEC_ACL_HUB1_CLIENTS
Hub1-n1(config-ext-nacl)#permit ip any 192.168.255.0 0.0.0.255
Hub1-n1(config-ext-nacl)#exit
```

5.8. Создайте крипто-карту (имя VPN_VRF_BLUE_SPOKE1, раздел 1) для построения IPsec туннеля между подсетью №1 центрального офиса и филиалом №1:

```
Hub1-n1(config)#crypto map VPN_VRF_BLUE_SPOKE1 1 ipsec-isakmp
```

5.8.1 Добавьте описание криптокарты:

```
Hub1-n1(config-crypto-map)#description VPN between Hub1 and Spoke1
```

5.8.2 Укажите список доступа для защищаемого трафика:

```
Hub1-n1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE1
```

5.8.3 Укажите при помощи какого набора алгоритмов нужно защищать трафик:

```
Hub1-n1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
```

5.8.4 Укажите IP адрес партнера по IPsec, в данном сценарии это внешний IP адрес устройства Spoke1:

```
Hub1-n1(config-crypto-map)#set peer 172.16.1.2
```

5.8.5 Задайте локальный IP адрес для взаимодействия с партнером, в данном сценарии это внешний VRRP IP адрес центрального шлюза:

```
Hub1-n1(config-crypto-map)#set local-address 172.16.100.2
```

В данном сценарии для всех криптокарт (только на центральных шлюзах), которые прикрепляются к VTI интерфейсам, необходимо указывать local-address. Это делается для того, чтобы предотвратить возможность построения защищенного соединения с резервной ноды VRRP кластера (на резервной ноде инициатором соединения может быть локальный трафик с VTI интерфейсов, например, от FRR или др.)

5.8.6 Прикрепите ISAKMP профиль `NEW_ALGS` к криптокарте:

```
Hub1-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
```

5.8.7 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-n1(config-crypto-map)#set pre-fragmentation
```

5.8.8 Включите автоматическую и интеллектуальную корректировку значения MSS (величина подбирается в зависимости от алгоритма шифрования и MTU трассы) для исходящего TCP трафика, чтобы его пакеты имели оптимальный размер и не фрагментировались криптошлюзом:

```
Hub1-n1(config-crypto-map)#set security-association mss-fix out
```

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

```
Hub1-n1(config-crypto-map)#set dead-connection history off
Hub1-n1(config-crypto-map)#exit
```

5.9. Создайте крипто-карту (имя VPN_VRF_BLUE_SPOKE2, раздел 1) для построения IPsec туннеля между подсетью №1 центрального офиса и филиалом №2:

```
Hub1-n1(config)#crypto map VPN_VRF_BLUE_SPOKE2 1 ipsec-isakmp
Hub1-n1(config-crypto-map)#description VPN between Hub1 and Spoke2
Hub1-n1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE2
Hub1-n1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1-n1(config-crypto-map)#set peer 172.16.2.2
Hub1-n1(config-crypto-map)#set local-address 172.16.100.2
Hub1-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1-n1(config-crypto-map)#set pre-fragmentation
Hub1-n1(config-crypto-map)#set security-association mss-fix out
Hub1-n1(config-crypto-map)#set dead-connection history off
Hub1-n1(config-crypto-map)#exit
```

5.10. Создайте крипто-карту (имя VPN_VRF_RED_SPOKE3, раздел 1) для построения IPsec туннеля между подсетью №2 центрального офиса и филиалом №3 (в качестве адреса партнера укажите внешние IP адреса устройства Spoke3):

```
Hub1-n1(config)#crypto map VPN_VRF_RED_SPOKE3 1 ipsec-isakmp
Hub1-n1(config-crypto-map)#description VPN between Hub1 and Spoke3
Hub1-n1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE3
Hub1-n1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1-n1(config-crypto-map)#set peer 172.16.3.2
Hub1-n1(config-crypto-map)#set peer 172.17.3.2
Hub1-n1(config-crypto-map)#set local-address 172.16.100.2
Hub1-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1-n1(config-crypto-map)#set pre-fragmentation
Hub1-n1(config-crypto-map)#set security-association mss-fix out
Hub1-n1(config-crypto-map)#set dead-connection history off
Hub1-n1(config-crypto-map)#exit
```

5.11. Создайте крипто-карту (имя VPN, раздел 1) для построения IPsec туннеля между подсетью №3 центрального офиса и филиалом №4:

```
Hub1-n1(config)#crypto map VPN 1 ipsec-isakmp
Hub1-n1(config-crypto-map)#description VPN between Hub1 and Spoke4
Hub1-n1(config-crypto-map)#match address IPSEC_ACL_HUB1_SPOKE4
Hub1-n1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1-n1(config-crypto-map)#set peer 172.16.4.2
Hub1-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1-n1(config-crypto-map)#set pre-fragmentation
Hub1-n1(config-crypto-map)#set security-association mss-fix out
Hub1-n1(config-crypto-map)#set dead-connection history off
Hub1-n1(config-crypto-map)#exit
```

5.12. Создайте динамическую крипто-карту (имя DMAP, раздел 1) для построения IPsec туннеля между подсетью №1 центрального офиса и удаленным VPN клиентом:

```
Hub1-n1(config)#crypto dynamic-map DMAP 1
Hub1-n1(config-crypto-map)#description VPN between Hub1 and remote Clients
Hub1-n1(config-crypto-map)#match address IPSEC_ACL_HUB1_CLIENTS
Hub1-n1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1-n1(config-crypto-map)#set pool IKECFG_POOL
Hub1-n1(config-crypto-map)#set local-address 172.16.100.2
Hub1-n1(config-crypto-map)#reverse-route
Hub1-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1-n1(config-crypto-map)#set pre-fragmentation
Hub1-n1(config-crypto-map)#set security-association mss-fix out
Hub1-n1(config-crypto-map)#set dead-connection history off
Hub1-n1(config-crypto-map)#exit
```

5.13. Создайте крипто-карту (имя VPN_VRF_BLUE_CLIENTS, раздел 1) и добавьте туда динамическую крипто-карту DMAP:

```
Hub1-n1(config)#crypto map VPN_VRF_BLUE_CLIENTS 1 ipsec-isakmp dynamic DMAP
Hub1-n1(config-crypto-map)#exit
```

5.14. Прикрепите крипто-карту `VPN_VRF_BLUE_CLIENTS` к интерфейсу `stVTI0` и добавьте описание:

```
Hub1-n1(config)#interface stVTI0
Hub1-n1(config-if)#description VRF BLUE VTI for VPN Clients
Hub1-n1(config-if)#crypto map VPN_VRF_BLUE_CLIENTS
Hub1-n1(config-if)#exit
```

5.15. Прикрепите крипто-карту `VPN_VRF_BLUE_SPOKE1` к интерфейсу `stVTI1` и добавьте описание:

```
Hub1-n1(config)#interface stVTI1
Hub1-n1(config-if)#description VRF BLUE VTI for Spoke1
Hub1-n1(config-if)#crypto map VPN_VRF_BLUE_SPOKE1
Hub1-n1(config-if)#exit
```

5.16. Прикрепите крипто-карту `VPN_VRF_BLUE_SPOKE2` к интерфейсу `stVTI2` и добавьте описание:

```
Hub1-n1(config)#interface stVTI2
Hub1-n1(config-if)#description VRF BLUE VTI for Spoke2
Hub1-n1(config-if)#crypto map VPN_VRF_BLUE_SPOKE2
Hub1-n1(config-if)#exit
```

5.17. Прикрепите крипто-карту `VPN_VRF_RED_SPOKE3` к интерфейсу `stVTI3` и добавьте описание:

```
Hub1-n1(config)#interface stVTI3
Hub1-n1(config-if)#description VRF RED VTI for Spoke3
Hub1-n1(config-if)#crypto map VPN_VRF_RED_SPOKE3
Hub1-n1(config-if)#exit
```

5.18. Прикрепите крипто-карту `VPN` к внешнему интерфейсу `GigabitEthernet0/0`:

```
Hub1-n1(config)#interface GigabitEthernet0/0
Hub1-n1(config-if)#crypto map VPN
Hub1-n1(config-if)#exit
```

Настройка VRRP

1. Включите VRRPv3:

```
Hub1-n1(config)#fhrp version vrrp v3
```

2. Войдите в режим конфигурирования VRRP инстанса на внешнем интерфейсе `gigabitEthernet0/0`:

```
Hub1-n1(config)#interface gigabitEthernet0/0
Hub1-n1(config-if)#vrrp 1 address-family ipv4
Hub1-n1(config-if-vrrp)#
```

3. Задайте VRRP IP адрес:

```
Hub1-n1(config-if-vrrp)#address 172.16.100.2 255.255.255.0
```

4. Установите приоритет для данного интерфейса:

```
Hub1-n1(config-if-vrrp)#priority 200
```

5. Задайте интервал отправки VRRP-объявлений:

```
Hub1-n1(config-if-vrrp)#timers advertise 2
```

6. Задайте интервал отправки gratuitous ARP-сообщений:

```
Hub1-n1(config-if-vrrp)#timers garp 5
```

7. Выйдите из режима конфигурирования интерфейса:

```
Hub1-n1(config-if-vrrp)#exit
Hub1-n1(config-if)#exit
```

8. Аналогичным образом настройте VRRP на интерфейсах `GigabitEthernet0/1`, `GigabitEthernet0/2`, `GigabitEthernet0/3`:

```
Hub1-n1(config)#interface GigabitEthernet0/1
Hub1-n1(config-if)#vrrp 2 address-family ipv4
Hub1-n1(config-if-vrrp)#timers advertise 2
Hub1-n1(config-if-vrrp)#timers garp 5
Hub1-n1(config-if-vrrp)#no preempt
Hub1-n1(config-if-vrrp)#priority 200
Hub1-n1(config-if-vrrp)#address 192.168.100.1 255.255.255.0
Hub1-n1(config-if-vrrp)#exit
Hub1-n1(config-if)#exit
Hub1-n1(config)#interface GigabitEthernet0/2
Hub1-n1(config-if)#vrrp 3 address-family ipv4
Hub1-n1(config-if-vrrp)#timers advertise 2
Hub1-n1(config-if-vrrp)#timers garp 5
Hub1-n1(config-if-vrrp)#no preempt
Hub1-n1(config-if-vrrp)#priority 200
Hub1-n1(config-if-vrrp)#address 192.168.100.1 255.255.255.0
Hub1-n1(config-if-vrrp)#exit
Hub1-n1(config-if)#exit
Hub1-n1(config)#interface GigabitEthernet0/3
Hub1-n1(config-if)#vrrp 4 address-family ipv4
Hub1-n1(config-if-vrrp)#timers advertise 2
Hub1-n1(config-if-vrrp)#timers garp 5
Hub1-n1(config-if-vrrp)#no preempt
Hub1-n1(config-if-vrrp)#priority 200
Hub1-n1(config-if-vrrp)#address 192.168.100.1 255.255.255.0
Hub1-n1(config-if-vrrp)#exit
Hub1-n1(config-if)#exit
```

9. Задайте VRRP маршрут по умолчанию, который будет устанавливаться в системе при переключении ноды в состояние Master и удаляться при переходе в состояние Backup или Fault (метрика этого маршрута должна быть меньше, чем у маршрута по умолчанию через внутренний VRRP IP адрес):

```
Hub1-n1(config)#vrrp ip route 0.0.0.0 0.0.0.0 172.16.100.1 src 172.16.100.2
```

10. Настройте вызов VRRP скрипта `notify_common`:

```
Hub1-n1(config)#vrrp notify common
```

11. Сохраните текущую конфигурацию и выйдите из cisco-like консоли:

```
Hub1-n1(config)#end
Hub1-n1#exit
```

```
administrator@Hub1-n1]
```

12. Перейдите в linux консоль:

```
administrator@Hub1-n1] system
```

```
Entering system shell...
root@Hub1-n1:~#
```

13. Откройте на редактирование файл `/etc/keepalived/notify_common.conf`; установите значение true для параметра `FLAG_MANAGE_SERVICES`, в параметре `SERVICES_LIST` укажите `frr`:

```
# Flag indicates that services which are set in SERVICES_LIST variable will be started or stopped in certain VRRP state.
FLAG_MANAGE_SERVICES="true"
# Services (separated by space, example: SERVICES_LIST="l2svc changeroutes") which will be started,
# when VRRP state changes to MASTER and stopped, when VRRP state changes to FAULT or BACKUP.
SERVICES_LIST="frr"
```

Данная настройка будет останавливать сервис FRR при переходе ноды кластера в состояние Backup или Fault и запускать его при переходе в состояние Master. Это нужно для того, чтобы с

резервной VRRP ноды не производились попытки построения защищенного соединения из-за наличия служебного трафика FRR.

Настройка динамической маршрутизации

Настройка динамической маршрутизации выполняется средствами ОС Linux с использованием пакета FRR. В качестве протокола динамической маршрутизации используется BGP.

1. Включите сервис FRR и добавьте его в автозапуск:

```
root@Hub1-n1:~# systemctl start frr.service
root@Hub1-n2:~# systemctl enable frr
```

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

2. Перейдите в консоль FRR:

```
root@Hub1-n1:~# vtysh
```

```
Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
Hub1-n1#
```

3. Настройте BGP:

```
Hub1-n1# configure terminal
Hub1-n1(config)# router bgp 65000 vrf vrf_blue
Hub1-n1(config-router)# bgp router-id 172.16.100.2
Hub1-n1(config-router)# neighbor 10.10.11.2 remote-as 65000
Hub1-n1(config-router)# neighbor 10.10.12.2 remote-as 65000
Hub1-n1(config-router)# address-family ipv4 unicast
Hub1-n1(config-router-af)# network 192.168.100.0/24
Hub1-n1(config-router-af)# exit
Hub1-n1(config-router)# exit
Hub1-n1(config)# router bgp 65000 vrf vrf_red
Hub1-n1(config-router)# bgp router-id 172.16.100.2
Hub1-n1(config-router)# neighbor 10.10.13.2 remote-as 65000
Hub1-n1(config-router)# address-family ipv4 unicast
Hub1-n1(config-router-af)# network 192.168.100.0/24
Hub1-n1(config-router-af)# exit
Hub1-n1(config-router)# exit
```

4. Сохраните текущую конфигурацию:

```
Hub1-n1# write
```

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

5. Выйдите из консоли FRR:

```
Hub1-n1# exit
```

```
root@Hub1-n1:~#
```

Настройка source NAT

Настройка source NAT осуществляется при необходимости доступа в Интернет резервной ноде кластера. Трансляция адресов осуществляется в кластерный VIP адрес 172.16.100.2 для трафика, выходящего с внешнего интерфейса `GigabitEthernet0/0`, который соответствует `eth0` (соответствие между именем интерфейса в нотации cisco-like и linux можно посмотреть в файле `/etc/ifaliases.cf`).

Подробное описание настроек приведено в документе «Инструкция по настройке межсетевого экрана на «С-Терра Шлюз» при помощи утилиты «iptables».

Обратите внимание:

1. Порядок обработки пакетов: сначала source NAT, потом IPsec (зашифрование);
2. Если требуется использовать метки (marks) в `iptables`, то на криптошлюзе запрещено изменять старшие 16 бит значения метки, так как они используются для внутренних нужд Продукта. Для работы с метками используйте маску (подробнее см. документ «Инструкция по настройке межсетевого экрана на «С-Терра Шлюз» при помощи утилиты «iptables»).

1. Очистите настройки `iptables`:

```
root@Hub1-n1:~# netfilter-persistent flush
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables flush
```

2. Добавьте две цепочки в таблицу NAT:

```
root@Hub1-n1:~# iptables -N NO_SOURCE_NAT -t nat
root@Hub1-n1:~# iptables -N SOURCE_NAT -t nat
```

В цепочку NO_SOURCE_NAT будет направляться трафик, для которого **не нужно** делать source NAT. В цепочку SOURCE_NAT будет направляться трафик, для которого **нужно** делать source NAT.

3. Добавьте правила в цепочку POSTROUTING для перенаправления пакетов в цепочки NO_SOURCE_NAT, SOURCE_NAT:

```
root@Hub1-n1:~# iptables -I POSTROUTING -t nat -j NO_SOURCE_NAT -m comment --comment
'pass traffic to chain to bypass source NAT'
root@Hub1-n1:~# iptables -A POSTROUTING -t nat -j SOURCE_NAT -m comment --comment
'pass traffic to chain to do source NAT'
```

Логика следующая: все пакеты из цепочки POSTROUTING сначала направляются в цепочку NO_SOURCE_NAT, в которой определяется трафик, для которого **не нужно** делать source NAT. После того как трафик был определен, он прекращает свою обработку (действие ACCEPT) и, соответственно, не попадает в цепочку SOURCE_NAT. То есть в цепочку SOURCE_NAT попадает все, что не «отсеялось» в цепочке NO_SOURCE_NAT.

4. Добавьте `iptables` правила в цепочку NO_SOURCE_NAT для трафика, для которого не нужно делать source NAT:

4.1. Выключите source NAT для локальных IKE/ESP/NAT-T пакетов:

```
root@Hub1-n1:~# iptables -t nat -A NO_SOURCE_NAT -m mark --mark 0x8000000/0x8000000 -
j ACCEPT -m comment --comment 'local IKE packets'
root@Hub1-n1:~# iptables -t nat -A NO_SOURCE_NAT -m mark --mark 0x40000000/0x40000000
-j ACCEPT -m comment --comment 'local ESP/NAT-T packets'
```

Метки 0x8000000 и 0x40000000 являются внутренними метками Продукта для локальных IKE и ESP/NAT-T пакетов соответственно.

4.2. Выключите source NAT для пакетов протокола VRPP:

```
root@Hub1-n1:~# iptables -t nat -A NO_SOURCE_NAT -p 112 -j ACCEPT -m comment --
comment 'local VRRP packets'
```

4.3. Выключите source NAT для трафика, который должен шифроваться (здесь следует указывать только трафик из подсети №3 центрального офиса до подсети филиала №4, то есть трафик, не принадлежащий VRF. Трафик VRF инстансов по умолчанию не будет попадать под правила Source NAT, так как натирование выполняется на WAN интерфейсе eth0, а трафик VRF инстансов будет уже зашифрован к моменту его прохождения черз WAN интерфейс):

```
root@Hub1-n1:~# iptables -t nat -A NO_SOURCE_NAT -s 192.168.100.0/24 -d
192.168.10.0/24 -j ACCEPT -m comment --comment 'traffic which must be protected by
IPsec'
```

5. Добавьте `iptables` правила в цепочку SOURCE_NAT для трафика, для которого нужно делать source NAT:

5.1. Включите source NAT для транзитного IPsec трафика, использующего порты 500/4500 протокола UDP:

```
root@Hub1-n1:~# iptables -t nat -A SOURCE_NAT -p udp --sport 500 -o eth0 -j SNAT --
to-source 172.16.100.2:40500-40540 -m comment --comment 'transit IKE packets'
root@Hub1-n1:~# iptables -t nat -A SOURCE_NAT -p udp --sport 4500 -o eth0 -j SNAT --
to-source 172.16.100.2:44500-44540 -m comment --comment 'transit ESP/NAT-T packets'
```

Нельзя, чтобы транзитный IPsec трафик использовал те же source UDP порты (500/4500), что и Продукт, так как в этом случае Продукт будет считать входящие транзитные IPsec пакеты за свои.

5.2. Включите source NAT для трафика из подсети 192.168.100.0/24:

```
root@Hub1-n1:~# iptables -t nat -A SOURCE_NAT -s 192.168.100.0/24 -o eth0 -j SNAT --
to-source 172.16.100.2
```

6. Сохраните добавленные `iptables` правила (правила сохраняются в файл `/etc/iptables/rules.v4`):

```
root@Hub1-n1:~# netfilter-persistent save
```

```
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
```

7. Добавьте сервис `netfilter-persistent.service` в автозапуск, чтобы правила `iptables` применялись во время старта криптошлюза:

```
root@Hub1-n1:~# systemctl enable netfilter-persistent.service
```

```
Synchronizing state of netfilter-persistent.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable netfilter-persistent
```

8. Для просмотра текущих NAT правил воспользуйтесь командой:

```
root@Hub1-n1:~# iptables -L -n -v -t nat
```

9. Для просмотра текущей таблицы NAT трансляций воспользуйтесь командой:

```
root@Hub1-n1:~# conntrack -L
```

Настройка политики обработки COC (CRL)

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

1. Войдите в cisco-like консоль из CLI разграничения доступа:

```
administrator@Hub1-n1] configure
```

```
Hub1-n1 login: cscons
Password:
S-Terra Gate 5.0.XXXXX (amd64)
Hub1-n1#
```

2. Войдите в режим конфигурации:

```
Hub1-n1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1-n1(config)#
```

3. Включите проверку COC:

```
Hub1-n1#configure terminal
Hub1-n1(config)# crypto pki trustpoint s-terra_technological_trustpoint
Hub1-n1(ca-trustpoint)# revocation-check crl
```

Если по обоснованным причинам использование СОС невозможно, то выключите проверку СОС и **не включайте автоматическую загрузку СОС**:

```
Hub1-n1(ca-trustpoint)# revocation-check none
```

4. Включите автоматическую загрузку и импортирование в базу Продукта списка отозванных сертификатов с HTTP сервера CRL_distribution_point:

```
Hub1-n1(ca-trustpoint)# crl download group ROOTCA http://172.16.101.15/certcrl.crl
```

5. Настройте периодичность загрузки СОС в 60 минут (по умолчанию 24 часа):

```
Hub1-n1(ca-trustpoint)# crl download time 60
```

6. Примените настройки:

```
Hub1-n1(ca-trustpoint)#end
```

7. Проверьте загружен ли СОС в базу Продукта:

```
Hub1-n1#run cert_mgr show
```

```
Found 2 certificates. Found 1 CRL.
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
2 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1-n1
3 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

Если СОС не загрузился, то проверьте файл журнала, например:

```
Hub1-n1#run grep getcrls_daemon /var/log/cspvpngate.log
```

Примечание 1: чтобы не ждать следующего периода загрузки СОС можно перезапустить сервис getcrls вручную:

```
Hub1-n1#run systemctl restart getcrls.service
```

Примечание 2: начиная с версии 5.0 СОС можно удалить из базы Продукта:

```
Hub1-n1#run cert_mgr remove -i <N>
```

```
3 OK   CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

8. Выполните проверку статуса сертификатов в базе Продукта:

```
Hub1-n1#run cert_mgr check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
2 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1-n1
```

Настройка межсетевого экрана

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

```
Hub1-n1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Hub1-n1(config)#object-group network SELF
Hub1-n1(config-network-group)#remark WAN IP
Hub1-n1(config-network-group)#host 172.16.100.10
Hub1-n1(config-network-group)#host 172.16.100.2
Hub1-n1(config-network-group)#exit
```

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

```
Hub1-n1(config)#ip access-list extended FW_INPUT_WAN
Hub1-n1(config-ext-nacl)#permit udp any object-group SELF eq 500
Hub1-n1(config-ext-nacl)#permit udp any object-group SELF eq 4500
Hub1-n1(config-ext-nacl)#permit icmp any object-group SELF
Hub1-n1(config-ext-nacl)#permit esp any object-group SELF
Hub1-n1(config-ext-nacl)#permit vrrp any any
Hub1-n1(config-ext-nacl)#exit
```

Данный ACL разрешает входящие пакеты протоколов IKE, ESP, ICMP, VRRP а остальные блокирует только для случая, когда криптошлюз является ответчиком **впервые**, если криптошлюз является инициатором или ответчиком в рамках ранее созданной сессии, то ответные пакеты будут разрешены в силу принципов работы stateful фильтрации (которая будет настроена далее) – динамические разрешающие правила более приоритетны чем статические, данные правила в cisco-like консоли НЕ отображаются, их можно посмотреть при помощи утилиты `fwconn_mgr` с параметром `show`.

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

```
Hub1-n1(config)#ip inspect name INSPECT tcp
Hub1-n1(config)#ip inspect name INSPECT udp
Hub1-n1(config)#ip inspect tcp sessions-max 256000
Hub1-n1(config)#ip inspect udp sessions-max 65536
```

Максимальное количество сессий настраивается в зависимости от условий, потребностей и производительности аппаратной платформы. Суммарное максимальное рекомендуемое значение для количества сессий – 10 000 000 (для Gate 7000+).

4. Прикрепите правила межсетевого экранирования на WAN интерфейс `GigabitEthernet0/0` (обратите особое внимание на направление правил фильтрации, список доступа `FW_INPUT_WAN` прикрепляется для входящих в интерфейс пакетов – **in**, а правило `INSPECT` stateful фильтрации прикрепляется для исходящих с интерфейса пакетов – **out**):

```
Hub1-n1(config)#interface GigabitEthernet0/0
Hub1-n1(config-if)#ip access-group FW_INPUT_WAN in
Hub1-n1(config-if)#ip inspect INSPECT out
Hub1-n1(config-if)#exit
```

5. Примените настройки:

```
Hub1-n1(config)#end
```

Настройка криптошлюза Hub1-n1 завершена.

В Приложении представлены тексты конфигураций для криптошлюза Hub1-n1:

- текст конфигурации консоли cisco-like;
- текст LSP;
- содержимое файла `/etc/keepalived/notify_common.conf`;
- текст конфигурации FRR;
- тексты конфигураций VRF и VTI интерфейсов.

Настройка криптошлюза Hub1-n2

Настройка криптошлюза происходит аналогично настройке Hub1-n1.

В Приложении представлены тексты конфигураций для криптошлюза Hub1-2:

- текст конфигурации консоли cisco-like;
- текст LSP;
- содержимое файла `/etc/keepalived/notify_common.conf`;
- текст конфигурации FRR;
- тексты конфигураций VRF и VTI интерфейсов.

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 разделе «Начальные настройки» базового сценария.

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 разделе «Настройки PKI (запросы и сертификаты)» базового сценария.

Настройка сетевых параметров

1. Войдите в cisco-like консоль из CLI разграничения доступа:

```
administrator@sterragate] configure
sterragate login: cscons
```

```
Password:
S-Terra Gate 5.0.XXXXX (amd64)
sterragate#
```

2. Войдите в режим конфигурации:

```
sterragate#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

3. Задайте имя устройства:

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

4. 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
```

5. 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.94c4
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.7f79
  MTU 1500 bytes
...
```

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

6. Настройте IP адреса на внешнем и внутреннем интерфейсах в соответствии со схемой стенда:

```
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
```

7. Задайте маршрут по умолчанию через устройство Router1:

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

8. 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=0.326 ms
108 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.263 ms
108 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.300 ms
108 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.271 ms
108 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.271 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4092ms
rtt min/avg/max/mdev = 0.263/0.286/0.326/0.023 ms
```

9. Сохраните текущую конфигурацию и выйдите из cisco-like консоли:

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

```
administrator@Spoke1]
```

10. Перейдите в linux консоль:

```
administrator@Spoke1] system
```

```
Entering system shell...
root@Spoke1:~#
```

11. Создайте VTI интерфейс. Настройка осуществляется средствами ОС Linux.

    11.1. Создайте файл `/etc/network/interfaces.d/st_vti0`, содержащий описание интерфейса st_vti0:

```
auto st_vti0
iface st_vti0 inet static
address 10.10.11.2/24
pre-up ip link del st_vti0 type dummy 2>/dev/null || true
```

```
pre-up ip link add st_vti0 type dummy
post-down ip link set dev st_vti0 down
```

12. Отредактируйте файл `/etc/ifaliases.cf`, добавив туда строку для VTI интерфейса:

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="stVTI0" pattern="st_vti0")
interface (name="default" pattern="*")
```

13. Пересчитайте контрольную сумму для файла `/etc/ifaliases.cf`:

```
root@Spoke1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```
SUCCESS:  Operation was successful.
```

14. Перезагрузите шлюз, чтобы все изменения вступили в силу.

Настройка шифрования

1. Войдите в cisco-like консоль из CLI разграничения доступа:

```
administrator@Spoke1] configure
```

```
Spoke1 login: cscons
Password:
S-Terra Gate 5.0.XXXXX (amd64)
Spoke1#
```

2. Войдите в режим конфигурации:

```
Spoke1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1-n1(config)#
```

3. Настройте параметры IKE.

   3.1. 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
```

   3.2. 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
```

   3.3. 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)#set policy 1
```

4. Настройте параметры IPsec.

   4.1. Отключите обновление ключа IPsec SA по достижению определенного объема переданного трафика (это позволит увеличить производительность и снизит потери при интенсивном трафике; разрешено для комбинированных алгоритмов шифрования, используемых в настоящем документе, так как в них есть встроенные механизмы обновления ключей по достижению определенного объема переданного трафика):

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

4.2. 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
```

4.3. Создайте список доступа (ACL) для трафика, который нужно защищать между шлюзом Spoke1 и центральным шлюзом:

```
Spoke1(config)#ip access-list extended IPSEC_ACL_SPOKE1_HUB1
Spoke1(config-ext-nacl)#permit ip any any
Spoke1(config-ext-nacl)#exit
```

4.4. Создайте крипто-карту (имя VPN, раздел 1):

```
Spoke1(config)#crypto map VPN 1 ipsec-isakmp
```

4.4.1 Укажите список доступа для защищаемого трафика:

```
Spoke1(config-crypto-map)#match address IPSEC_ACL_SPOKE1_HUB1
```

4.4.2 Укажите при помощи какого набора алгоритмов нужно защищать трафик:

```
Spoke1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
```

4.4.3 Укажите IP адрес партнера по IPsec, в данном сценарии это внешний VRRP IP адрес центрального шлюза:

```
Spoke1(config-crypto-map)#set peer 172.16.100.2
```

4.4.4 Прикрепите ISAKMP профиль `NEW_ALGS` к криптокарте:

```
Spoke1(config-crypto-map)#set isakmp-profile NEW_ALGS
```

4.4.5 Включите префрагментацию:

```
Spoke1(config-crypto-map)#set pre-fragmentation
```

4.4.6 Включите автоматическую и интеллектуальную корректировку значения MSS:

```
Spoke1(config-crypto-map)#set security-association mss-fix out
```

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

```
Spoke1(config-crypto-map)#set dead-connection history off
Spoke1(config-crypto-map)#exit
```

4.5. Прикрепите крипто-карту `VPN` к интерфейсу `stVTI0`:

```
Spoke1(config)#interface stVTI0
Spoke1(config-if)#crypto map VPN
Spoke1(config-if)#exit
```

Важно! В случаях, когда криптокарты прикрепляются к VTI интерфейсам, следует помнить о том, что на WAN интерфейсе шлюза всегда должна присутствовать какая-либо политика, обеспечивающая обработку трафика, будь то криптокарта, список доступа или заданная по умолчанию фильтрация IPv6 трафика (`ipv6 default filtering deny`). В противном случае весь трафик на WAN интерфейсе не будет обрабатываться VPN драйвером, соответственно, не будет возможности построения защищенных соединений.

Настройка политики обработки COC (CRL)

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

1. Включите проверку COC:

```
Spoke1(config)#crypto pki trustpoint s-terra_technological_trustpoint
Spoke1(ca-trustpoint)# revocation-check crl
```

Если по обоснованным причинам использование СОС невозможно, то выключите проверку СОС и **не включайте автоматическую загрузку СОС**:

```
Hub1-n1(ca-trustpoint)# revocation-check none
```

2. Включите автоматическую загрузку и импортирование в базу Продукта списка отозванных сертификатов с HTTP сервера CRL_distribution_point:

```
Spoke1(ca-trustpoint)# crl download group ROOTCA http://172.16.101.15/certcrl.crl
```

3. Настройте периодичность загрузки СОС в 60 минут (по умолчанию 24 часа):

```
Spoke1(ca-trustpoint)# crl download time 60
```

4. Примените настройки:

```
Spoke1(ca-trustpoint)#end
```

5. Проверьте загружен ли СОС в базу Продукта:

```
Spoke#run cert_mgr show
```

```
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
2 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Spoke1
3 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

Если СОС не загрузился, то проверьте файл журнала, например:

```
Spoke1#run grep getcrls_daemon /var/log/cspvpngate.log
```

Примечание 1: чтобы не ждать следующего периода загрузки СОС можно перезапустить сервис getcrls вручную:

```
Spoke1#run systemctl restart getcrls.service
```

Примечание 2: начиная с версии 5.0 СОС можно удалить из базы Продукта:

```
Spoke1#run cert_mgr remove -i <N>
```

```
3 OK   CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

6. Выполните проверку статуса сертификатов в базе Продукта:

```
Spoke1#run cert_mgr check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
2 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Spoke1
```

Настройка межсетевого экрана

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

```
Spoke1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
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
```

2. 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
```

Данный ACL разрешает входящие пакеты протоколов IKE, ESP, ICMP, а остальные блокирует только для случая, когда криптошлюз является ответчиком **впервые**, если криптошлюз является инициатором или ответчиком в рамках ранее созданной сессии, то ответные пакеты будут разрешены в силу принципов работы stateful фильтрации (которая будет настроена далее) – динамические разрешающие правила более приоритетны чем статические, данные правила в cisco-like консоли НЕ отображаются, их можно посмотреть при помощи утилиты `fwconn_mgr` с параметром `show`.

3. 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
```

Максимальное количество сессий настраивается в зависимости от условий, потребностей и производительности аппаратной платформы. Суммарное максимальное рекомендуемое значение для количества сессий – 10 000 000 (для Gate 7000+).

4. Прикрепите правила межсетевого экранирования на WAN интерфейс `GigabitEthernet0/0` (обратите особое внимание на направление правил фильтрации, список доступа `FW_INPUT_WAN` прикрепляется для входящих в интерфейс пакетов – **in**, а правило `INSPECT` stateful фильтрации прикрепляется для исходящих с интерфейса пакетов – **out**):

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

5. Примените настройки:

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

Настройка динамической маршрутизации

Настройка динамической маршрутизации выполняется средствами ОС Linux с использованием пакета FRR. В качестве протокола динамической маршрутизации используется BGP.

1. Включите сервис FRR и добавьте его в автозапуск:

```
root@Spoke1:~# systemctl start frr.service
root@Spoke1:~# systemctl enable frr
```

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

2. Перейдите в консоль FRR:

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

```
Hello, this is FRRouting (version 9.0.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
Hub1-n1#
```

3. Настройте BGP:

```
Spoke1# configure terminal
Spoke1(config)# router bgp 65000
Spoke1(config-router)# bgp router-id 172.16.1.2
Spoke1(config-router)# neighbor 10.10.11.1 remote-as 65000
Spoke1(config-router)# address-family ipv4 unicast
Spoke1(config-router-af)# network 192.168.1.0/24
Spoke1(config-router-af)# exit
```

```
Spoke1(config-router)# exit
```

4. Сохраните текущую конфигурацию:

```
Spoke1# write
```

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

5. Выйдите из консоли FRR:

```
Spoke1# exit
```

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

Настройка криптошлюза Spoke1 завершена.

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст конфигурации консоли cisco-like;
- текст LSP;
- текст конфигурации FRR;
- тексты конфигурации VTI интерфейса.

Настройка криптошлюза Spoke2

Настройка криптошлюза происходит аналогично настройке Spoke1.

В Приложении представлены тексты конфигураций для криптошлюза Spoke2:

- текст консоли cisco-like;
- текст LSP;
- текст конфигурации FRR;
- текст конфигурации VTI интерфейса.

Настройка криптошлюза Spoke3

Настройка криптошлюза происходит аналогично настройке Spoke1.

Настройка multiwan

1. Создайте конфигурационный файл `/etc/multiwan/config.yaml` со следующим содержимым:

```
config_format_version: 1
links_params:
  check_links_reachability_on_start: true
  preempt: no
  links:
    -
      iface_name: eth0
      name: ISP1
      prio: 200
      routes:
        ipv4:
          networks:
            - 0.0.0.0/0
          nexthop: 172.16.3.1
      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: 172.17.3.1
    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 автозагрузку:

```
root@Spoke3:~# systemctl enable multiwan.service
root@Spoke3:~# systemctl start multiwan.service
```

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

В Приложении представлены тексты конфигураций для криптошлюза Spoke3:

- текст консоли cisco-like;
- текст LSP;
- текст конфигурации FRR;
- текст конфигурации multiwan;
- текст конфигурации VTI интерфейса.

Настройка криптошлюза Spoke4

Настройка криптошлюза выполняется на основе базового сценария. Настраивать VRF, VTI интерфейсы или динамическую маршрутизацию на криптошлюзе Spoke4 не требуется.

В Приложении представлены тексты конфигураций для криптошлюза Spoke4:

- текст консоли cisco-like;
- текст LSP;

Настройка удаленного VPN клиента Client1

1. Настройте получение IP адреса по DHCP на сетевом интерфейсе.
2. Разрешите прием и отправку IP пакетов.

Развертывание VPN клиента «С-Терра Клиент» выполняется с помощью «С-Терра ГСУ» 5.0 или «С-Терра AdminTool» 5.0. Подробная настройка приводится в соответствующем сценарии.

В Приложении представлен текст конфигурации для Client1:

- текст LSP

Проверка работоспособности стенда

Проверка IP связности

1. Проверьте, что с криптошлюзов Hub1-n1, Hub1-n2, Spoke1, Spoke2, Spoke3, Spoke4 доступен маршрутизатор Router1. Для этого выполните команду ping из cisco-like консоли криптошлюзов (пример вывода для Hub1-n1):

```
Hub1-n1#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=0.290 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.433 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.404 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.616 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.501 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4076ms
rtt min/avg/max/mdev = 0.290/0.448/0.616/0.107 ms
```

2. Проверьте, что со всех защищаемых устройств доступен их соответствующий шлюз по умолчанию. Для этого выполните команду ping из консоли защищаемых устройств (пример вывода для Host1-b-hub1):

```
root@Host1-b-hub1:~# ping 192.168.100.1 -c 4
```

```
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=0.118 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.337 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.238 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=0.268 ms

--- 192.168.100.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.118/0.240/0.337/0.079 ms
```

Проверка PKI

1. Проверьте, что на всех криптошлюзах присутствуют локальный и доверенный сертификаты, а также CRL (если загрузка списка отозванных сертификатов была настроена). Для этого выполните следующие команды из linux консоли криптошлюзов (пример вывода для Hub1-n1):

```
root@Hub1-n1:~# cert_mgr show
```

```
Found 2 certificates. Found 1 CRL.
1 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1-n1
2 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
3 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

```
root@Hub1-n1:~# cert_mgr check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1-n1
2 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

Проверка состояния VRRP кластера

1. Проверьте, что основная нода Hub1-n1 находится в состоянии Master, а резервная Hub1-n2 в состоянии Backup. Для этого выполните следующую команду из cisco-like консоли криптошлюзов:

```
Hub1-n1#show vrrp
```

```
Interface                     VRID  State   AddrFamily
-----------------------------------------------------
GigabitEthernet0/0            1     Master  IPv4
GigabitEthernet0/1            2     Master  IPv4
```

```
GigabitEthernet0/2              3     Master  IPv4
GigabitEthernet0/3              4     Master  IPv4
```

```
Hub1-n2#show vrrp
```

```
Interface                       VRID  State   AddrFamily
-------------------------------------------------------
GigabitEthernet0/0              1     Backup  IPv4
GigabitEthernet0/1              2     Backup  IPv4
GigabitEthernet0/2              3     Backup  IPv4
GigabitEthernet0/3              4     Backup  IPv4
```

2. Убедитесь в том, что на основной ноде Hub1-n1 маршрут по умолчанию через маршрутизатор Router1, а на резервной ноде Hub1-n2 через VIP адрес доверенного сегмента:

```
Hub1-n1#show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.16.100.1 to network 0.0.0.0

S*   0.0.0.0/0 [0/0] via 172.16.100.1
               [100/0] via 192.168.100.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.100.0 [0/0] is directly connected, GigabitEthernet0/0
C    192.168.100.0/24 [0/0] is directly connected, GigabitEthernet0/3
```

```
Hub1-n2#show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.100.10 to network 0.0.0.0

S*   0.0.0.0/0 [100/0] via 192.168.100.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.100.0 [0/0] is directly connected, GigabitEthernet0/0
C    192.168.100.0/24 [0/0] is directly connected, GigabitEthernet0/3
```

Проверка построения защищенного соединения

1. Убедитесь, что на основной ноде Hub1-n1 построены защищенные соединения до филиальных криптошлюзов и удаленного VPN клиента Client1. Защищенные соединения с криптошлюзами Spoke1, Spoke2 и Spoke3 инициируются автоматически за счет служебного трафика протокола динамической маршрутизации. Для построения защищенного соединения до филиала №4 отправьте ICMP запросы с устройства Host3-b-hub1 на устройство Host-b-spoke4 (или в обратном направлении). Для построения защищенного соединения с устройством Client1, отправьте ICMP запросы с Client1 (предварительно выполнив авторизацию в окне Login в «С-Терра Клиент») на устройство Host1-b-hub1.

   Для просмотра установленных соединений выполните следующую команду из linux консоли криптошлюза Hub1-n1:

```
root@Hub1-n1:~# sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.100.2,500)-(172.16.1.2,500) active 2428 2396
2 2 (172.16.100.2,500)-(172.16.2.2,500) active 1928 1960
3 3 (172.16.100.2,500)-(172.16.3.2,500) active 1928 1960
4 4 (172.16.100.2,500)-(172.16.4.2,500) active 1928 1960
5 5 (172.16.100.2,4500)-(172.16.5.2,4500) active 2016 2416

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 152 124
2 2 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 1076 1264
3 3 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 1348 1772
4 4 (192.168.100.0-192.168.100.255,*)-(192.168.10.0-192.168.10.255,*) * ESP tunn 528
528
5 5 (192.168.100.0-192.168.100.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 512 512
```

Если защищенные соединения не построились, проверьте файл журнала `/var/log/cspvpngate.log` (при необходимости увеличьте уровень логирования командой `logging trap debugging` в cisco-like консоли) и состояние сервиса FRR.

Проверка связности между VTI интерфейсами внутри VRF

1. Убедитесь, что с основной ноды Hub1-n1 доступны криптошлюзы Spoke1, Spoke2 и Spoke3 через VTI интерфейсы из соответствующих VRF инстансов. Для этого выполните следующие команды из linux консоли криптошлюза:

```
root@Hub1-n1:~# ip vrf exec vrf_blue ping 10.10.11.2 -I 10.10.11.1 -c 4
```

```
PING 10.10.11.2 (10.10.11.2) from 10.10.11.1 : 56(84) bytes of data.
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=0.564 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=0.621 ms
64 bytes from 10.10.11.2: icmp_seq=3 ttl=64 time=0.643 ms
64 bytes from 10.10.11.2: icmp_seq=4 ttl=64 time=0.610 ms

--- 10.10.11.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3055ms
rtt min/avg/max/mdev = 0.564/0.609/0.643/0.028 ms
```

```
root@Hub1-n1:~# ip vrf exec vrf_blue ping 10.10.12.2 -I 10.10.12.1 -c 4
```

```
PING 10.10.12.2 (10.10.12.2) from 10.10.12.1 : 56(84) bytes of data.
64 bytes from 10.10.12.2: icmp_seq=1 ttl=64 time=0.610 ms
64 bytes from 10.10.12.2: icmp_seq=2 ttl=64 time=0.717 ms
64 bytes from 10.10.12.2: icmp_seq=3 ttl=64 time=0.621 ms
64 bytes from 10.10.12.2: icmp_seq=4 ttl=64 time=0.626 ms

--- 10.10.12.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3022ms
rtt min/avg/max/mdev = 0.610/0.643/0.717/0.042 ms
```

```
root@Hub1-n1:~# ip vrf exec vrf_red ping 10.10.13.2 -I 10.10.13.1 -c 4
```

```
PING 10.10.13.2 (10.10.13.2) from 10.10.13.1 : 56(84) bytes of data.
64 bytes from 10.10.13.2: icmp_seq=1 ttl=64 time=0.686 ms
64 bytes from 10.10.13.2: icmp_seq=2 ttl=64 time=0.768 ms
64 bytes from 10.10.13.2: icmp_seq=3 ttl=64 time=0.801 ms
64 bytes from 10.10.13.2: icmp_seq=4 ttl=64 time=0.818 ms

--- 10.10.13.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 0.686/0.768/0.818/0.050 ms
```

Проверка динамической маршрутизации

1. Убедитесь, что на основной ноде Hub1-n1 в таблицах маршрутизации под номерами 10 и 20, которые соответствуют VRF инстансам `vrf_blue` и `vrf_red`, присутствуют маршруты до защищаемых подсетей филиалов, полученные по протоколу BGP. В таблице маршрутизации 10 также должен присутствовать RRI маршрут до удаленного VPN клиента (если с Client1 установлено соединение). Для этого выполните следующие команды из linux консоли криптошлюза:

```
root@Hub1-n1:~# ip r show table 10
```

```
unreachable default metric 4278198272
10.10.10.0/24 dev st_vti0 proto kernel scope link src 10.10.10.1
local 10.10.10.1 dev st_vti0 proto kernel scope host src 10.10.10.1
broadcast 10.10.10.255 dev st_vti0 proto kernel scope link src 10.10.10.1
10.10.11.0/24 dev st_vti1 proto kernel scope link src 10.10.11.1
local 10.10.11.1 dev st_vti1 proto kernel scope host src 10.10.11.1
broadcast 10.10.11.255 dev st_vti1 proto kernel scope link src 10.10.11.1
10.10.12.0/24 dev st_vti2 proto kernel scope link src 10.10.12.1
local 10.10.12.1 dev st_vti2 proto kernel scope host src 10.10.12.1
broadcast 10.10.12.255 dev st_vti2 proto kernel scope link src 10.10.12.1
192.168.1.0/24 nhid 106 via 10.10.11.2 dev st_vti1 proto bgp metric 20
192.168.2.0/24 nhid 110 via 10.10.12.2 dev st_vti2 proto bgp metric 20
192.168.100.0/24 dev eth1 proto kernel scope link src 192.168.100.10
local 192.168.100.1 dev eth1 proto kernel scope host src 192.168.100.10
local 192.168.100.10 dev eth1 proto kernel scope host src 192.168.100.10
broadcast 192.168.100.255 dev eth1 proto kernel scope link src 192.168.100.10
192.168.255.1 dev st_vti0 proto st_rri
```

```
root@Hub1-n1:~# ip r show table 20
```

```
unreachable default metric 4278198272
10.10.13.0/24 dev st_vti3 proto kernel scope link src 10.10.13.1
local 10.10.13.1 dev st_vti3 proto kernel scope host src 10.10.13.1
broadcast 10.10.13.255 dev st_vti3 proto kernel scope link src 10.10.13.1
192.168.10.0/24 nhid 108 via 10.10.13.2 dev st_vti3 proto bgp metric 20
192.168.100.0/24 dev eth2 proto kernel scope link src 192.168.100.10
local 192.168.100.1 dev eth2 proto kernel scope host src 192.168.100.10
local 192.168.100.10 dev eth2 proto kernel scope host src 192.168.100.10
broadcast 192.168.100.255 dev eth2 proto kernel scope link src 192.168.100.10
```

Если BGP маршруты в таблицах маршрутизации отсутствуют, проверьте конфигурацию FRR.

2. Убедитесь, что на основной ноде Hub1-n1 в FRR установлены отношения соседства с BGP пирами. Для этого выполните следующие команды из консоли FRR криптошлюза:

```
Hub1-n1# show ip bgp vrf vrf_blue summary
```

```
IPv4 Unicast Summary (VRF vrf_blue):
BGP router identifier 172.16.100.2, local AS number 65000 vrf-id 11
BGP table version 3
RIB entries 5, using 960 bytes of memory
Peers 2, using 41 KiB of memory

Neighbor          V            AS    MsgRcvd    MsgSent    TblVer   InQ  OutQ   Up/Down
State/PfxRcd   PfxSnt Desc
10.10.11.2        4       65000         164         164          3     0     0 02:40:51
1        1 N/A
10.10.12.2        4       65000         163         163          3     0     0 02:39:21
1        1 N/A

Total number of neighbors 2
```

```
Hub1-n1# show ip bgp vrf vrf_red summary
```

```
IPv4 Unicast Summary (VRF vrf_red):
BGP router identifier 172.16.100.2, local AS number 65000 vrf-id 12
```

```
BGP table version 2
RIB entries 3, using 576 bytes of memory
Peers 1, using 20 KiB of memory

Neighbor          V           AS    MsgRcvd    MsgSent    TblVer   InQ  OutQ   Up/Down
State/PfxRcd   PfxSnt Desc
10.10.13.2        4        65000         160         160          2     0     0 02:36:00
1        1 N/A

Total number of neighbors 1
```

Проверка доступности хостов

1. Убедитесь, что с устройства Host1-b-hub1 по протоколу ICMP доступны устройства Host-b-spoke1 и Host-b-spoke2. Для этого выполните следующие команды из консоли Host1-b-hub1:

```
root@Host1-b-hub1:~# ping 192.168.1.100 -c 4
```

```
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=62 time=1.55 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=62 time=3.18 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=62 time=1.25 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=62 time=1.72 ms

--- 192.168.1.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.252/1.928/3.184/0.744 ms
```

```
root@Host1-b-hub1:~# ping 192.168.2.100 -c 4
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_seq=1 ttl=62 time=1.28 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=62 time=1.78 ms
64 bytes from 192.168.2.100: icmp_seq=3 ttl=62 time=1.46 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=62 time=1.46 ms

--- 192.168.2.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.275/1.491/1.775/0.179 ms
```

2. Убедитесь, что с устройства Host1-b-hub1 по протоколу ICMP **не доступны** устройства Host-b-spoke3 и Host-b-spoke4. Для этого выполните следующую команду из консоли Host1-b-hub1:

```
root@Host1-b-hub1:~# ping 192.168.10.100 -c 4
```

```
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
From 192.168.100.10 icmp_seq=1 Destination Host Unreachable
From 192.168.100.10 icmp_seq=2 Destination Host Unreachable
From 192.168.100.10 icmp_seq=3 Destination Host Unreachable
From 192.168.100.10 icmp_seq=4 Destination Host Unreachable

--- 192.168.10.100 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3005ms
```

3. Убедитесь, что с устройства Host2-b-hub1 по протоколу ICMP доступно устройство Host-b-spoke3. Для этого выполните следующую команду из консоли Host2-b-hub1:

```
root@Host2-b-hub1:~# ping 192.168.10.100 -c 4
```

```
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=62 time=1.45 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=62 time=1.17 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=62 time=1.17 ms
64 bytes from 192.168.10.100: icmp_seq=4 ttl=62 time=1.49 ms

--- 192.168.10.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
```

```
rtt min/avg/max/mdev = 1.168/1.318/1.487/0.149 ms
```

4. Убедитесь, что с устройства Host2-b-hub1 по протоколу ICMP **не доступны** устройства Host-b-spoke1 и Host-b-spoke2. Для этого выполните следующие команды из консоли Host1-b-hub1:

```
root@Host2-b-hub1:~# ping 192.168.1.100 -c 4
```

```
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
From 192.168.100.10 icmp_seq=1 Destination Host Unreachable
From 192.168.100.10 icmp_seq=2 Destination Host Unreachable
From 192.168.100.10 icmp_seq=3 Destination Host Unreachable
From 192.168.100.10 icmp_seq=4 Destination Host Unreachable

--- 192.168.1.100 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3005ms
```

```
root@Host2-b-hub1:~# ping 192.168.2.100 -c 4
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
From 192.168.100.10 icmp_seq=1 Destination Host Unreachable
From 192.168.100.10 icmp_seq=2 Destination Host Unreachable
From 192.168.100.10 icmp_seq=3 Destination Host Unreachable
From 192.168.100.10 icmp_seq=4 Destination Host Unreachable

--- 192.168.2.100 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3005ms
```

5. Убедитесь, что с устройства Host3-b-hub1 по протоколу ICMP доступно устройство Host-b-spoke4. Для этого выполните следующую команду из консоли Host3-b-hub1:

```
root@Host3-b-hub1:~# ping 192.168.10.100 -c 4
```

```
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=62 time=1.15 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=62 time=1.48 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=62 time=1.09 ms
64 bytes from 192.168.10.100: icmp_seq=4 ttl=62 time=1.53 ms

--- 192.168.10.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.092/1.314/1.533/0.194 ms
```

6. Убедитесь, что с устройства Host3-b-hub1 по протоколу ICMP **не доступны** устройства Host-b-spoke1 и Host-b-spoke2. Для этого выполните следующие команды из консоли Host1-b-hub1:

```
root@Host3-b-hub1:~# ping 192.168.1.100 -c 4
```

```
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.

--- 192.168.1.100 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3073ms
```

```
root@Host3-b-hub1:~# ping 192.168.2.100 -c 4
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.

--- 192.168.2.100 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3053ms
```

7. Убедитесь, что с устройства Client1 по протоколу ICMP доступно устройство Host1-b-hub1. Для этого выполните следующую команду из консоли устройства Client1 (предварительно выполнив авторизацию в окне Login в «С-Терра Клиент»):

```
C:\Users\Administrator>ping 192.168.100.100
```

```
Pinging 192.168.100.100 with 32 bytes of data:
Reply from 192.168.100.100: bytes=32 time=1ms TTL=62
Reply from 192.168.100.100: bytes=32 time=1ms TTL=62
Reply from 192.168.100.100: bytes=32 time=1ms TTL=62
```

```
Reply from 192.168.100.100: bytes=32 time=1ms TTL=62

Ping statistics for 192.168.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

8. Убедитесь, что с устройств Host2-b-hub1 и Host3-b-hub1 по протоколу ICMP **не доступно** устройство Client1. Для этого выполните следующую команду из консоли Host2-b-hub1 и Host3-b-hub1:

```
root@Host2-b-hub1:~# ping 192.168.255.1 -c 4
```

```
PING 192.168.255.1 (192.168.255.1) 56(84) bytes of data.
From 192.168.100.10 icmp_seq=1 Destination Host Unreachable
From 192.168.100.10 icmp_seq=2 Destination Host Unreachable
From 192.168.100.10 icmp_seq=3 Destination Host Unreachable
From 192.168.100.10 icmp_seq=4 Destination Host Unreachable

--- 192.168.255.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3005ms
```

```
root@Host3-b-hub1:~# ping 192.168.255.1 -c 4
```

```
PING 192.168.255.1 (192.168.255.1) 56(84) bytes of data.

--- 192.168.255.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3078ms
```

где 192.168.255.1 – IKECFG IP адрес VPN клиента Client1.

Проверка отказоустойчивого решения

1. Смоделируйте отказ на основной ноде Hub1-n1, например отключите сетевой кабель от интерфейса `GigabitEthernet0/0` и убедитесь, что роль мастера перешла к Hub1-n2:

```
Hub1-n1#show ip interface gigabitEthernet 0/0
```

```
GigabitEthernet0/0 is up, line protocol is down
  Internet address is 172.16.100.10/24
  MTU is 1500 bytes
  Outgoing ip access list is not set
  Inbound ip access list is FW_INPUT_WAN
```

```
Hub1-n1#show vrrp
```

```
Interface                     VRID  State   AddrFamily
------------------------------------------------------
GigabitEthernet0/0            1     Fault   IPv4
GigabitEthernet0/1            2     Fault   IPv4
GigabitEthernet0/2            3     Fault   IPv4
GigabitEthernet0/3            4     Fault   IPv4
```

```
Hub1-n2#show vrrp
```

```
Interface                     VRID  State   AddrFamily
------------------------------------------------------
GigabitEthernet0/0            1     Master  IPv4
GigabitEthernet0/1            2     Master  IPv4
GigabitEthernet0/2            3     Master  IPv4
GigabitEthernet0/3            4     Master  IPv4
```

2. Убедитесь, что на ноде Hub1-n1 сервис FRR был остановлен автоматически, и в основной таблице маршрутизации остался единственный маршрут по умолчанию через внутренний VRRP IP адрес. Для этого выполните следующие команды из linux консоли криптошлюза:

```
root@Hub1-n1:~# systemctl status frr.service
```

```
○ frr.service - FRRouting
     Loaded: loaded (/lib/systemd/system/frr.service; enabled; preset: disabled)
```

```
     Active: inactive (dead) since Mon 2024-02-19 15:21:22 MSK; 16min ago
   Duration: 3h 17min 57.512s
       Docs: https://frrouting.readthedocs.io/en/latest/setup.html
    Process:     1532     ExecStart=/usr/lib/frr/frrinit.sh     start     (code=exited,
status=0/SUCCESS)
    Process:     2005     ExecStop=/usr/lib/frr/frrinit.sh     stop     (code=exited,
status=0/SUCCESS)
   Main PID: 1541 (code=exited, status=0/SUCCESS)
        CPU: 7.258s
```

```
root@Hub1-n1:~# ip r
```

```
default via 192.168.100.1 dev eth3 proto st_policy metric 100
172.16.100.0/24 dev eth0 proto kernel scope link src 172.16.100.10 linkdown
192.168.100.0/24 dev eth3 proto kernel scope link src 192.168.100.10
```

3. Убедитесь, что на ноде Hub1-n2 построены защищенные соединения. Для построения защищенного соединения до филиала №4 отправьте ICMP запросы с устройства Host3-b-hub1 на устройство Host-b-spoke4 (или в обратном направлении). Для построения защищенного соединения с устройством Client1, отправьте ICMP запросы с Client1 (предварительно выполнив авторизацию в окне Login в «С-Терра Клиент») на устройство Host1-b-hub1.

   Для просмотра установленных соединений выполните следующую команду из linux консоли криптошлюза Hub1-n2:

```
root@Hub1-n2:~# sa_mgr show
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.100.2,500)-(172.16.1.2,500) active 1960 1928
2 2 (172.16.100.2,500)-(172.16.2.2,500) active 2412 2296
3 3 (172.16.100.2,500)-(172.16.3.2,500) active 2260 2228
4 4 (172.16.100.2,500)-(172.16.4.2,500) active 1928 1960
5 5 (172.16.100.2,4500)-(172.16.5.2,4500) active 2016 2416

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 5748 6536
2 2 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 7012 7896
3 3 (0.0.0.0-255.255.255.255,*)-(0.0.0.0-255.255.255.255,*) * ESP tunn 604 660
4 4 (192.168.100.0-192.168.100.255,*)-(192.168.10.0-192.168.10.255,*) * ESP tunn 792
792
5 5 (192.168.100.0-192.168.100.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 512 512
```

4. Убедитесь, что на ноде Hub1-n2 в таблицах маршрутизации под номерами 10 и 20 появились маршруты до защищаемых подсетей филиалов, полученные по протоколу BGP. В таблице маршрутизации 10 также должен присутствовать RRI маршрут до удаленного VPN клиента (если с Client1 установлено соединение).

   Для этого выполните следующие команды из linux консоли криптошлюза:

```
root@Hub1-n2:~# ip r show table 10
```

```
unreachable default metric 4278198272
10.10.10.0/24 dev st_vti0 proto kernel scope link src 10.10.10.1
local 10.10.10.1 dev st_vti0 proto kernel scope host src 10.10.10.1
broadcast 10.10.10.255 dev st_vti0 proto kernel scope link src 10.10.10.1
10.10.11.0/24 dev st_vti1 proto kernel scope link src 10.10.11.1
local 10.10.11.1 dev st_vti1 proto kernel scope host src 10.10.11.1
broadcast 10.10.11.255 dev st_vti1 proto kernel scope link src 10.10.11.1
10.10.12.0/24 dev st_vti2 proto kernel scope link src 10.10.12.1
local 10.10.12.1 dev st_vti2 proto kernel scope host src 10.10.12.1
broadcast 10.10.12.255 dev st_vti2 proto kernel scope link src 10.10.12.1
192.168.1.0/24 nhid 108 via 10.10.11.2 dev st_vti1 proto bgp metric 20
192.168.2.0/24 nhid 110 via 10.10.12.2 dev st_vti2 proto bgp metric 20
```

```
192.168.100.0/24 dev eth1 proto kernel scope link src 192.168.100.20
local 192.168.100.1 dev eth1 proto kernel scope host src 192.168.100.20
local 192.168.100.20 dev eth1 proto kernel scope host src 192.168.100.20
broadcast 192.168.100.255 dev eth1 proto kernel scope link src 192.168.100.20
192.168.255.1 dev st_vti0 proto st_rri
```

```
root@Hub1-n2:~# ip r show table 20
```

```
unreachable default metric 4278198272
10.10.13.0/24 dev st_vti3 proto kernel scope link src 10.10.13.1
local 10.10.13.1 dev st_vti3 proto kernel scope host src 10.10.13.1
broadcast 10.10.13.255 dev st_vti3 proto kernel scope link src 10.10.13.1
192.168.10.0/24 nhid 112 via 10.10.13.2 dev st_vti3 proto bgp metric 20
192.168.100.0/24 dev eth2 proto kernel scope link src 192.168.100.20
local 192.168.100.1 dev eth2 proto kernel scope host src 192.168.100.20
local 192.168.100.20 dev eth2 proto kernel scope host src 192.168.100.20
broadcast 192.168.100.255 dev eth2 proto kernel scope link src 192.168.100.20
```

5. Исправьте причину отказа на основной ноде Hub1-n1 и убедитесь, что она перешла из состояния Fault в Backup:

```
Hub1-n1#show vrrp
```

```
Interface                VRID  State   AddrFamily
-------------------------------------------------------
GigabitEthernet0/0       1     Backup  IPv4
GigabitEthernet0/1       2     Backup  IPv4
GigabitEthernet0/2       3     Backup  IPv4
GigabitEthernet0/3       4     Backup  IPv4
```

Нода Hub1-n1 не захватит роль Master (пока нода Hub1-n2 работоспособна), так как выключен `preempt mode`.

Проверка multiwan

Проверьте отказоустойчивость провайдеров на филиальном криптошлюзе Spoke3 в соответствии со сценарием «Резервирование каналов связи при помощи сервиса MultiWAN».

Приложение

Конфигурации криптошлюза Hub1-n1

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
!
!
fhrp version vrrp v3
hostname Hub1-n1
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
!
ip local pool IKECFG_POOL 192.168.255.1 192.168.255.254
!
crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.100.2
 host 172.16.100.10
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE1
 permit ip any any
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE2
 permit ip any any
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE3
 permit ip any any
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE4
 permit ip 192.168.100.0 0.0.0.255 192.168.10.0 0.0.0.255
!
ip access-list extended IPSEC_ACL_HUB1_CLIENTS
 permit ip any 192.168.255.0 0.0.0.255
!
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
 permit 112 any any
!
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 dynamic-map DMAP 1
 match address IPSEC_ACL_HUB1_CLIENTS
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set pool IKECFG_POOL
 set security-association mss-fix out
 set local-address 172.16.100.2
 reverse-route
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_BLUE_SPOKE1 1 ipsec-isakmp
```

```
 description VPN between Hub1 and Spoke1
 match address IPSEC_ACL_HUB1_SPOKE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_BLUE_SPOKE2 1 ipsec-isakmp
 description VPN between Hub1 and Spoke2
 match address IPSEC_ACL_HUB1_SPOKE2
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.2.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_RED_SPOKE3 1 ipsec-isakmp
 description VPN between Hub1 and Spoke3
 match address IPSEC_ACL_HUB1_SPOKE3
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.3.2
 set peer 172.17.3.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN 1 ipsec-isakmp
 description VPN between Hub1 and Spoke4
 match address IPSEC_ACL_HUB1_SPOKE4
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.4.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_BLUE_CLIENTS 1 ipsec-isakmp dynamic DMAP
!
interface GigabitEthernet0/0
 ip address 172.16.100.10 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
 vrrp 1 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  priority 200
  address 172.16.100.2 255.255.255.0
  exit-vrrp
!
interface GigabitEthernet0/1
 description VRF BLUE LAN
 ip address 192.168.100.10 255.255.255.0
 vrrp 2 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  priority 200
```

```
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface GigabitEthernet0/2
 description VRF RED LAN
 ip address 192.168.100.10 255.255.255.0
 vrrp 3 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  priority 200
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface GigabitEthernet0/3
 ip address 192.168.100.10 255.255.255.0
 vrrp 4 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  priority 200
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface stVTI0
 description VRF BLUE VTI for VPN Clients
 ip address 10.10.10.1 255.255.255.0
 crypto map VPN_VRF_BLUE_CLIENTS
!
interface stVTI1
 description VRF BLUE VTI for Spoke1
 ip address 10.10.11.1 255.255.255.0
 crypto map VPN_VRF_BLUE_SPOKE1
!
interface stVTI2
 description VRF BLUE VTI for Spoke2
 ip address 10.10.12.1 255.255.255.0
 crypto map VPN_VRF_BLUE_SPOKE2
!
interface stVTI3
 description VRF RED VTI for Spoke3
 ip address 10.10.13.1 255.255.255.0
 crypto map VPN_VRF_RED_SPOKE3
!
!
ip route 0.0.0.0 0.0.0.0 192.168.100.20 100
!
!
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
!
vrrp ip route 0.0.0.0 0.0.0.0 172.16.100.1 src 172.16.100.2
vrrp notify common
end
```

2. Конфигурация LSP:

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

```
#   Conversion Date/Time:   Fri Oct 31 15:13:21 2025

GlobalParameters(
    Title                              = "This LSP was automatically generated by CSP
Converter at Fri Oct 31 15:13:21 2025 (user: cscons)"
    Version                     = LSP_5_0
    CRLHandlingMode             = ENABLE
    PreserveIPsecSA             = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 192.168.100.20
            Metric = 100
        )

)

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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
```

```
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.10
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.10
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.10
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.10
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 112
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
```

```
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain InspectChain:INSPECT (
    Filters = Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

AddressPool IKECFG_POOL
(
    IPAddresses = 192.168.255.1..192.168.255.254
)
```

```
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 = 172.16.4.2
    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_VRF_BLUE_CLIENTS:1:DMAP:1
(
    IKELocalIPFilter = 172.16.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    IKECFGPool          = IKECFG_POOL
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 100
)

IKERule IKERule:VPN_VRF_BLUE_SPOKE1:1
(
    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            = 20
)

IKERule IKERule:VPN_VRF_BLUE_SPOKE2:1
(
    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            = 30
)

IKERule IKERule:VPN_VRF_RED_SPOKE3:1
(
    IKEPeerIPFilter = 172.16.3.2, 172.17.3.2
    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            = 40
)

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.4.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:1
)

IPsecAction IPsecAction:VPN_VRF_BLUE_CLIENTS:1:DMAP:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    ReverseRoute = TRUE
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_BLUE_CLIENTS:1:DMAP:1
)

IPsecAction IPsecAction:VPN_VRF_BLUE_SPOKE1:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.1.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_BLUE_SPOKE1:1
)

IPsecAction IPsecAction:VPN_VRF_BLUE_SPOKE2:1
(
    TunnelingParameters = TunnelEntry(
```

```
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.2.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_BLUE_SPOKE2:1
)

IPsecAction IPsecAction:VPN_VRF_RED_SPOKE3:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.3.2
        Prefragmentation = ENABLE
    ),
    TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.17.3.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_RED_SPOKE3:1
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.10.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_SPOKE4"
    )
)

FilterChain IPsecPolicy:VPN_VRF_BLUE_CLIENTS (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 192.168.255.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_BLUE_CLIENTS:1:DMAP:1 >
        LogEventID                                                              =
"IPsec:Protect:VPN_VRF_BLUE_CLIENTS:1:DMAP:1:IPSEC_ACL_HUB1_CLIENTS"
    )
)
```

```
FilterChain IPsecPolicy:VPN_VRF_BLUE_SPOKE1 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_BLUE_SPOKE1:1 >
        LogEventID = "IPsec:Protect:VPN_VRF_BLUE_SPOKE1:1:IPSEC_ACL_HUB1_SPOKE1"
    )
)

FilterChain IPsecPolicy:VPN_VRF_BLUE_SPOKE2 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_BLUE_SPOKE2:1 >
        LogEventID = "IPsec:Protect:VPN_VRF_BLUE_SPOKE2:1:IPSEC_ACL_HUB1_SPOKE2"
    )
)

FilterChain IPsecPolicy:VPN_VRF_RED_SPOKE3 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_RED_SPOKE3:1 >
        LogEventID = "IPsec:Protect:VPN_VRF_RED_SPOKE3:1:IPSEC_ACL_HUB1_SPOKE3"
    )
)

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

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

```
NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/4"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "stVTI0"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_BLUE_CLIENTS
)

NetworkInterface (
    LogicalName = "stVTI1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_BLUE_SPOKE1
)

NetworkInterface (
    LogicalName = "stVTI2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_BLUE_SPOKE2
)

NetworkInterface (
    LogicalName = "stVTI3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_RED_SPOKE3
)
```

3. Содержимое файла `/etc/keepalived/notify_common.conf`:

```
# Flag indicates that IP addresses which are set in RESERVE_IPADDRESSES variable will
be added or removed in certain VRRP state.
FLAG_MANAGE_IPADDRESSES="false"
# IP addresses (separated by space, example: RESERVE_IPADDRESSES="192.168.1.1/24
192.168.2.1/24", RESERVE_IPV6ADDRESSES="2001:db8:0:4003::1/64 2001:db8:0:4003::2/64")
# which will be set on interface RESERVE_IPADDRESSES_INTERFACE, when VRRP state
changes to FAULT or BACKUP.
RESERVE_IPADDRESSES="192.168.1.1/24 192.168.2.1/24"
RESERVE_IPV6ADDRESSES="2001:db8:0:4003::1/64 2001:db8:0:4003::2/64"
# Interface on which addreses will be set.
# INTERFACE MUST NOT PARTICIPATE IN VRRP CLUSTER
RESERVE_IPADDRESSES_INTERFACE="eth1"

# Flag indicates that routes which are set in RESERVE_ROUTES variable will be added
or removed in certain VRRP state.
```

```
FLAG_MANAGE_ROUTES="false"
#    Routes    (separated    by    space,    example:    RESERVE_ROUTES="192.168.1.0/24
192.168.2.0/24",    RESERVE_IPV6ROUTES="2001:db8:0:4003::/64    2001:db8:0:4003::/64")
which will be added in to route table,
# when VRRP state changes to FAULT or BACKUP.
RESERVE_ROUTES="0.0.0.0/0"
RESERVE_IPV6ROUTES="::/0"
# Nexthop IP for route in RESERVE_NEXTHOP or/and RESERVE_IPV6NEXTHOP variable.
RESERVE_NEXTHOP="192.168.100.1"
RESERVE_IPV6NEXTHOP="2001:db8:0:4003::100"
# Routes metric (NOTE: do not use 0 value).
RESERVE_METRIC="1"
RESERVE_IPV6METRIC="1"
# Flag indicates that netfilter sessions will be synchronized upon changing VRRP
state from BACKUP/FAULT to MASTER
FLAG_SYNC_NETFILTER_SESSIONS="false"

# Flag indicates that services which are set in SERVICES_LIST variable will be
started or stopped in certain VRRP state.
FLAG_MANAGE_SERVICES="true"
# Services (separated by space, example: SERVICES_LIST="l2svc changeroutes") which
will be started,
# when VRRP state changes to MASTER and stopped, when VRRP state changes to FAULT or
BACKUP.
SERVICES_LIST="frr"


################################################    specific    to    DMVPN    scenario
############################################################
# Flag indicates that DMVPN scenario is using on clustered spokes.
# Used to manage BGP neighbors shutdown state or OSPF interfaces passive mode when
VRRP node changes its STATE.
FLAG_DMVPN_SCENARIO_ON_CLUSTERED_SPOKE="false"

### IBGP specific options ###
# Internal routing protocol for DMVPN (IBGP or OSPF).
DMVPN_INTERNAL_ROUTING_PROTOCOL="IBGP"

# Local BGP AS Number. Set it if internal routing protocol is IBGP.
DMVPN_LOCAl_ASN="65000"

# BGP neighbors (separated by space) which must be shut down if VRRP node transits to
BACKUP or FAULT state,
# and must not be shut down if VRRP node transits to MASTER state.
# Set it if internal routing protocol is IBGP.
DMVPN_BGP_NEIGHBORS="10.10.10.100 10.10.20.200"


### OSPF specific options ###
# mGRE OSPF interfaces (separated by space) which must be in passive mode if VRRP
node transits to BACKUP or FAULT state
# and must not be in passive mode if VRRP node transits to MASTER state.
# Set it if internal routing protocol is OSPF.
DMVPN_OSPF_INTERFACES="mgre0 mgre1"
```

4. Конфигурация FRR:

```
frr version 9.0.2
frr defaults traditional
hostname Hub1
log syslog informational
service integrated-vtysh-config
!
```

```
router bgp 65000 vrf vrf_blue
 bgp router-id 172.16.100.2
 neighbor 10.10.11.2 remote-as 65000
 neighbor 10.10.12.2 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.100.0/24
 exit-address-family
exit
!
router bgp 65000 vrf vrf_red
 bgp router-id 172.16.100.2
 neighbor 10.10.13.2 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.100.0/24
 exit-address-family
exit
!
```

5. Конфигурация сетевых интерфейсов.

5.1. Файл `/etc/network/interfaces.d/vrf_blue`:

```
auto vrf_blue
iface vrf_blue inet manual
pre-up ip link del vrf_blue type vrf 2>/dev/null || true
pre-up ip route flush table 10 2>/dev/null || true
pre-up ip link add vrf_blue type vrf table 10
pre-up ip link set dev vrf_blue up
pre-up ip route add table 10 unreachable default metric 4278198272
pre-up ip link set dev eth1 master vrf_blue
pre-up ip link set dev st_vti0 master vrf_blue
pre-up ip link set dev st_vti1 master vrf_blue
pre-up ip link set dev st_vti2 master vrf_blue
post-down ip link set dev vrf_blue down
```

5.2. Файл `/etc/network/interfaces.d/vrf_red`:

```
auto vrf_red
iface vrf_red inet manual
pre-up ip link del vrf_red type vrf 2>/dev/null || true
pre-up ip route flush table 20 2>/dev/null || true
pre-up ip link add vrf_red type vrf table 20
pre-up ip link set dev vrf_red up
pre-up ip route add table 20 unreachable default metric 4278198272
pre-up ip link set dev eth2 master vrf_red
pre-up ip link set dev st_vti3 master vrf_red
post-down ip link set dev vrf_red down
```

5.3. Файл `/etc/network/interfaces.d/st_vti0`:

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

5.4. Файл `/etc/network/interfaces.d/st_vti1`:

```
auto st_vti1
iface st_vti1 inet static
address 10.10.11.1/24
pre-up ip link del st_vti1 type dummy 2>/dev/null || true
pre-up ip link add st_vti1 type dummy
```

```
post-down ip link set dev st_vti1 down
```

5.5. Файл `/etc/network/interfaces.d/st_vti2`:

```
auto st_vti2
iface st_vti2 inet static
address 10.10.12.1/24
pre-up ip link del st_vti2 type dummy 2>/dev/null || true
pre-up ip link add st_vti2 type dummy
post-down ip link set dev st_vti2 down
```

5.6. Файл `/etc/network/interfaces.d/st_vti3`:

```
auto st_vti3
iface st_vti3 inet static
address 10.10.13.1/24
pre-up ip link del st_vti3 type dummy 2>/dev/null || true
pre-up ip link add st_vti3 type dummy
post-down ip link set dev st_vti3 down
```

Конфигурации криптошлюза Hub1-n2

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
!
!
fhrp version vrrp v3
hostname Hub1-n2
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
!
ip local pool IKECFG_POOL 192.168.255.1 192.168.255.254
!
crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
```

```
object-group network SELF
 remark WAN IP
 host 172.16.100.2
 host 172.16.100.20
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE1
 permit ip any any
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE2
 permit ip any any
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE3
 permit ip any any
!
ip access-list extended IPSEC_ACL_HUB1_SPOKE4
 permit ip 192.168.100.0 0.0.0.255 192.168.10.0 0.0.0.255
!
ip access-list extended IPSEC_ACL_HUB1_CLIENTS
 permit ip any 192.168.255.0 0.0.0.255
!
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
 permit 112 any any
!
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 dynamic-map DMAP 1
 match address IPSEC_ACL_HUB1_CLIENTS
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set pool IKECFG_POOL
 set security-association mss-fix out
 set local-address 172.16.100.2
 reverse-route
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_BLUE_SPOKE1 1 ipsec-isakmp
 description VPN between Hub1 and Spoke1
 match address IPSEC_ACL_HUB1_SPOKE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_BLUE_SPOKE2 1 ipsec-isakmp
 description VPN between Hub1 and Spoke2
 match address IPSEC_ACL_HUB1_SPOKE2
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.2.2
 set pre-fragmentation
```

```
 set dead-connection history off
!
crypto map VPN_VRF_RED_SPOKE3 1 ipsec-isakmp
 description VPN between Hub1 and Spoke3
 match address IPSEC_ACL_HUB1_SPOKE3
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.100.2
 set peer 172.16.3.2
 set peer 172.17.3.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN 1 ipsec-isakmp
 description VPN between Hub1 and Spoke4
 match address IPSEC_ACL_HUB1_SPOKE4
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.4.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN_VRF_BLUE_CLIENTS 1 ipsec-isakmp dynamic DMAP
!
interface GigabitEthernet0/0
 ip address 172.16.100.20 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
 vrrp 1 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  address 172.16.100.2 255.255.255.0
  exit-vrrp
!
interface GigabitEthernet0/1
 description VRF BLUE LAN
 ip address 192.168.100.20 255.255.255.0
 vrrp 2 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface GigabitEthernet0/2
 description VRF RED LAN
 ip address 192.168.100.20 255.255.255.0
 vrrp 3 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface GigabitEthernet0/3
 ip address 192.168.100.20 255.255.255.0
 vrrp 4 address-family ipv4
  timers advertise 2
  timers garp 5
  no preempt
  address 192.168.100.1 255.255.255.0
  exit-vrrp
```

```
!
interface stVTI0
 description VRF BLUE VTI for VPN Clients
 ip address 10.10.10.1 255.255.255.0
 crypto map VPN_VRF_BLUE_CLIENTS
!
interface stVTI1
 description VRF BLUE VTI for Spoke1
 ip address 10.10.11.1 255.255.255.0
 crypto map VPN_VRF_BLUE_SPOKE1
!
interface stVTI2
 description VRF BLUE VTI for Spoke2
 ip address 10.10.12.1 255.255.255.0
 crypto map VPN_VRF_BLUE_SPOKE2
!
interface stVTI3
 description VRF RED VTI for Spoke3
 ip address 10.10.13.1 255.255.255.0
 crypto map VPN_VRF_RED_SPOKE3
!
!
ip route 0.0.0.0 0.0.0.0 192.168.100.10 100
!
!
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
!
vrrp ip route 0.0.0.0 0.0.0.0 172.16.100.1 src 172.16.100.2
vrrp notify common
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Fri Oct 31 15:15:26 2025

GlobalParameters(
    Title                              = "This LSP was automatically generated by CSP
Converter at Fri Oct 31 15:15:26 2025 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 192.168.100.10
            Metric = 100
        )

)

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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.20
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.20
```

```
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.20
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.100.20
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 112
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain InspectChain:INSPECT (
    Filters = Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
```

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

AddressPool IKECFG_POOL
(
    IPAddresses = 192.168.255.1..192.168.255.254
)

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 = 172.16.4.2
    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_VRF_BLUE_CLIENTS:1:DMAP:1
(
    IKELocalIPFilter = 172.16.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    IKECFGPool          = IKECFG_POOL
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 100
)

IKERule IKERule:VPN_VRF_BLUE_SPOKE1:1
(
    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            = 20
)

IKERule IKERule:VPN_VRF_BLUE_SPOKE2:1
(
    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            = 30
)

IKERule IKERule:VPN_VRF_RED_SPOKE3:1
(
    IKEPeerIPFilter = 172.16.3.2, 172.17.3.2
    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            = 40
)

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
```

```
        PeerAddress = 172.16.4.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:1
)

IPsecAction IPsecAction:VPN_VRF_BLUE_CLIENTS:1:DMAP:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    ReverseRoute = TRUE
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_BLUE_CLIENTS:1:DMAP:1
)

IPsecAction IPsecAction:VPN_VRF_BLUE_SPOKE1:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.1.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_BLUE_SPOKE1:1
)

IPsecAction IPsecAction:VPN_VRF_BLUE_SPOKE2:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.2.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_BLUE_SPOKE2:1
)

IPsecAction IPsecAction:VPN_VRF_RED_SPOKE3:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.16.3.2
        Prefragmentation = ENABLE
    ),
    TunnelEntry(
        LocalIPAddress = 172.16.100.2
        PeerAddress = 172.17.3.2
```

```
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN_VRF_RED_SPOKE3:1
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.10.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_SPOKE4"
    )
)

FilterChain IPsecPolicy:VPN_VRF_BLUE_CLIENTS (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 192.168.255.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_BLUE_CLIENTS:1:DMAP:1 >
        LogEventID                                                                =
"IPsec:Protect:VPN_VRF_BLUE_CLIENTS:1:DMAP:1:IPSEC_ACL_HUB1_CLIENTS"
    )
)

FilterChain IPsecPolicy:VPN_VRF_BLUE_SPOKE1 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_BLUE_SPOKE1:1 >
        LogEventID = "IPsec:Protect:VPN_VRF_BLUE_SPOKE1:1:IPSEC_ACL_HUB1_SPOKE1"
    )
)

FilterChain IPsecPolicy:VPN_VRF_BLUE_SPOKE2 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
```

```
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_BLUE_SPOKE2:1 >
        LogEventID = "IPsec:Protect:VPN_VRF_BLUE_SPOKE2:1:IPSEC_ACL_HUB1_SPOKE2"
    )
)

FilterChain IPsecPolicy:VPN_VRF_RED_SPOKE3 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_VRF_RED_SPOKE3:1 >
        LogEventID = "IPsec:Protect:VPN_VRF_RED_SPOKE3:1:IPSEC_ACL_HUB1_SPOKE3"
    )
)

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

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

NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/4"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "stVTI0"
    InputFilter = IPv6Default
```

```
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_BLUE_CLIENTS
)

NetworkInterface (
    LogicalName = "stVTI1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_BLUE_SPOKE1
)

NetworkInterface (
    LogicalName = "stVTI2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_BLUE_SPOKE2
)

NetworkInterface (
    LogicalName = "stVTI3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN_VRF_RED_SPOKE3
)
```

3. Содержимое файла `/etc/keepalived/notify_common.conf`:

```
# Flag indicates that IP addresses which are set in RESERVE_IPADDRESSES variable will
be added or removed in certain VRRP state.
FLAG_MANAGE_IPADDRESSES="false"
# IP addresses (separated by space, example: RESERVE_IPADDRESSES="192.168.1.1/24
192.168.2.1/24", RESERVE_IPV6ADDRESSES="2001:db8:0:4003::1/64 2001:db8:0:4003::2/64")
# which will be set on interface RESERVE_IPADDRESSES_INTERFACE, when VRRP state
changes to FAULT or BACKUP.
RESERVE_IPADDRESSES="192.168.1.1/24 192.168.2.1/24"
RESERVE_IPV6ADDRESSES="2001:db8:0:4003::1/64 2001:db8:0:4003::2/64"
# Interface on which addreses will be set.
# INTERFACE MUST NOT PARTICIPATE IN VRRP CLUSTER
RESERVE_IPADDRESSES_INTERFACE="eth1"

# Flag indicates that routes which are set in RESERVE_ROUTES variable will be added
or removed in certain VRRP state.
FLAG_MANAGE_ROUTES="false"
# Routes (separated by space, example: RESERVE_ROUTES="192.168.1.0/24
192.168.2.0/24", RESERVE_IPV6ROUTES="2001:db8:0:4003::/64 2001:db8:0:4003::/64")
which will be added in to route table,
# when VRRP state changes to FAULT or BACKUP.
RESERVE_ROUTES="0.0.0.0/0"
RESERVE_IPV6ROUTES="::/0"
# Nexthop IP for route in RESERVE_NEXTHOP or/and RESERVE_IPV6NEXTHOP variable.
RESERVE_NEXTHOP="192.168.100.1"
RESERVE_IPV6NEXTHOP="2001:db8:0:4003::100"
# Routes metric (NOTE: do not use 0 value).
RESERVE_METRIC="1"
RESERVE_IPV6METRIC="1"
# Flag indicates that netfilter sessions will be synchronized upon changing VRRP
state from BACKUP/FAULT to MASTER
FLAG_SYNC_NETFILTER_SESSIONS="false"

# Flag indicates that services which are set in SERVICES_LIST variable will be
started or stopped in certain VRRP state.
FLAG_MANAGE_SERVICES="true"
```

```
# Services (separated by space, example: SERVICES_LIST="l2svc changeroutes") which
will be started,
# when VRRP state changes to MASTER and stopped, when VRRP state changes to FAULT or
BACKUP.
SERVICES_LIST="frr"


############################################    specific    to    DMVPN    scenario
############################################################
# Flag indicates that DMVPN scenario is using on clustered spokes.
# Used to manage BGP neighbors shutdown state or OSPF interfaces passive mode when
VRRP node changes its STATE.
FLAG_DMVPN_SCENARIO_ON_CLUSTERED_SPOKE="false"

### IBGP specific options ###
# Internal routing protocol for DMVPN (IBGP or OSPF).
DMVPN_INTERNAL_ROUTING_PROTOCOL="IBGP"

# Local BGP AS Number. Set it if internal routing protocol is IBGP.
DMVPN_LOCAl_ASN="65000"

# BGP neighbors (separated by space) which must be shut down if VRRP node transits to
BACKUP or FAULT state,
# and must not be shut down if VRRP node transits to MASTER state.
# Set it if internal routing protocol is IBGP.
DMVPN_BGP_NEIGHBORS="10.10.10.100 10.10.20.200"


### OSPF specific options ###
# mGRE OSPF interfaces (separated by space) which must be in passive mode if VRRP
node transits to BACKUP or FAULT state
# and must not be in passive mode if VRRP node transits to MASTER state.
# Set it if internal routing protocol is OSPF.
DMVPN_OSPF_INTERFACES="mgre0 mgre1"
```

4. Конфигурация FRR:

```
frr version 9.0.2
frr defaults traditional
hostname Hub1
log syslog informational
service integrated-vtysh-config
!
router bgp 65000 vrf vrf_blue
 bgp router-id 172.16.100.2
 neighbor 10.10.11.2 remote-as 65000
 neighbor 10.10.12.2 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.100.0/24
 exit-address-family
exit
!
router bgp 65000 vrf vrf_red
 bgp router-id 172.16.100.2
 neighbor 10.10.13.2 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.100.0/24
 exit-address-family
exit
!
```

5. Конфигурация сетевых интерфейсов.

5.1. Файл `/etc/network/interfaces.d/vrf_blue`:

```
auto vrf_blue
iface vrf_blue inet manual
pre-up ip link del vrf_blue type vrf 2>/dev/null || true
pre-up ip route flush table 10 2>/dev/null || true
pre-up ip link add vrf_blue type vrf table 10
pre-up ip link set dev vrf_blue up
pre-up ip route add table 10 unreachable default metric 4278198272
pre-up ip link set dev eth1 master vrf_blue
pre-up ip link set dev st_vti0 master vrf_blue
pre-up ip link set dev st_vti1 master vrf_blue
pre-up ip link set dev st_vti2 master vrf_blue
post-down ip link set dev vrf_blue down
```

5.2. Файл `/etc/network/interfaces.d/vrf_red`:

```
auto vrf_red
iface vrf_red inet manual
pre-up ip link del vrf_red type vrf 2>/dev/null || true
pre-up ip route flush table 20 2>/dev/null || true
pre-up ip link add vrf_red type vrf table 20
pre-up ip link set dev vrf_red up
pre-up ip route add table 20 unreachable default metric 4278198272
pre-up ip link set dev eth2 master vrf_red
pre-up ip link set dev st_vti3 master vrf_red
post-down ip link set dev vrf_red down
```

5.3. Файл `/etc/network/interfaces.d/st_vti0`:

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

5.4. Файл `/etc/network/interfaces.d/st_vti1`:

```
auto st_vti1
iface st_vti1 inet static
address 10.10.11.1/24
pre-up ip link del st_vti1 type dummy 2>/dev/null || true
pre-up ip link add st_vti1 type dummy
post-down ip link set dev st_vti1 down
```

5.5. Файл `/etc/network/interfaces.d/st_vti2`:

```
auto st_vti2
iface st_vti2 inet static
address 10.10.12.1/24
pre-up ip link del st_vti2 type dummy 2>/dev/null || true
pre-up ip link add st_vti2 type dummy
post-down ip link set dev st_vti2 down
```

5.6. Файл `/etc/network/interfaces.d/st_vti3`:

```
auto st_vti3
iface st_vti3 inet static
address 10.10.13.1/24
pre-up ip link del st_vti3 type dummy 2>/dev/null || true
pre-up ip link add st_vti3 type dummy
post-down ip link set dev st_vti3 down
```

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_SPOKE1_HUB1
 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
 match address IPSEC_ACL_SPOKE1_HUB1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
```

```
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 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 inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
!
interface stVTI0
 ip address 10.10.11.2 255.255.255.0
 crypto map VPN
!
!
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:   Fri Oct 31 15:17:08 2025

GlobalParameters(
    Title                              = "This LSP was automatically generated by CSP
Converter at Fri Oct 31 15:17:08 2025 (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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.1.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.1.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
```

```
        Action = DROP
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain InspectChain:INSPECT (
    Filters = Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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
)

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

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_SPOKE1_HUB1"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
)

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

```
NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "stVTI0"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN
)
```

3. Конфигурация FRR:

```
frr version 9.0.1
frr defaults traditional
hostname Spoke1
log syslog informational
service integrated-vtysh-config
!
router bgp 65000
 bgp router-id 172.16.1.2
 neighbor 10.10.11.1 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.1.0/24
 exit-address-family
exit
!
```

4. Конфигурация сетевых интерфейсов.

   4.1. Файл `/etc/network/interfaces.d/st_vti0`:

```
auto st_vti0
iface st_vti0 inet static
address 10.10.11.2/24
pre-up ip link del st_vti0 type dummy 2>/dev/null || true
pre-up ip link add st_vti0 type dummy
post-down ip link set dev st_vti0 down
```

Конфигурации криптошлюза Spoke2

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 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_SPOKE2_HUB1
 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
 match address IPSEC_ACL_SPOKE2_HUB1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 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 inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.2.1 255.255.255.0
!
interface stVTI0
 ip address 10.10.12.2 255.255.255.0
 crypto map VPN
!
!
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:   Fri Oct 31 15:18:48 2025

GlobalParameters(
    Title                              = "This LSP was automatically generated by CSP
Converter at Fri Oct 31 15:18:48 2025 (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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.2.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.2.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.2.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.2.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain InspectChain:INSPECT (
    Filters = Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
```

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

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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
)

IPsecAction IPsecAction:VPN:1
```

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

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_SPOKE2_HUB1"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
)

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

NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "stVTI0"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN
)
```

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
```

```
 neighbor 10.10.12.1 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.2.0/24
 exit-address-family
exit
!
```

4. Конфигурация сетевых интерфейсов.

   4.1. Файл `/etc/network/interfaces.d/st_vti0`:

```
auto st_vti0
iface st_vti0 inet static
address 10.10.12.2/24
pre-up ip link del st_vti0 type dummy 2>/dev/null || true
pre-up ip link add st_vti0 type dummy
post-down ip link set dev st_vti0 down
```

Конфигурации криптошлюза Spoke3

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 Spoke3
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 172.16.3.2
 host 172.17.3.2
!
```

```
ip access-list extended IPSEC_ACL_SPOKE3_HUB1
 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
 match address IPSEC_ACL_SPOKE3_HUB1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.3.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 172.17.3.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 ip inspect INSPECT out
!
interface GigabitEthernet0/2
 ip address 192.168.10.1 255.255.255.0
!
interface stVTI0
 ip address 10.10.13.2 255.255.255.0
 crypto map VPN
!
!
!
!
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:   Fri Oct 31 15:20:31 2025

GlobalParameters(
```

```
    Title                              = "This LSP was automatically generated by CSP
Converter at Fri Oct 31 15:20:31 2025 (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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.3.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.17.3.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.3.2
```

```
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.17.3.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.3.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.17.3.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.3.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.17.3.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain InspectChain:INSPECT (
    Filters = Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
```

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

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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
)

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

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_SPOKE3_HUB1"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "stVTI0"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN
)
```

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 172.16.3.2
 neighbor 10.10.13.1 remote-as 65000
 !
 address-family ipv4 unicast
  network 192.168.10.0/24
 exit-address-family
exit
!
```

4. Конфигурация multiwan:

```
config_format_version: 1
links_params:
  check_links_reachability_on_start: true
  preempt: no
  links:
    -
      iface_name: eth0
      name: ISP1
      prio: 200
      routes:
        ipv4:
          networks:
            - 0.0.0.0/0
          nexthop: 172.16.3.1
      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: 172.17.3.1
      check:
        hosts_reachability: any
        ipv4:
          hosts:
            - 172.16.101.15
        method:
          icmp:
            interval: 1
```

```
            restore_interval: 1
            restore_count: 3
            failure_count: 3
```

5. Конфигурация сетевых интерфейсов.

   5.1. Файл `/etc/network/interfaces.d/st_vti0`:

```
auto st_vti0
iface st_vti0 inet static
address 10.10.13.2/24
pre-up ip link del st_vti0 type dummy 2>/dev/null || true
pre-up ip link add st_vti0 type dummy
post-down ip link set dev st_vti0 down
```

Конфигурации криптошлюза Spoke4

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 Spoke4
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 172.16.4.2
!
ip access-list extended IPSEC_ACL_SPOKE4_HUB1
 permit ip 192.168.10.0 0.0.0.255 192.168.100.0 0.0.0.255
!
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
 match address IPSEC_ACL_SPOKE4_HUB1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.4.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.10.1 255.255.255.0
!
!
ip route 0.0.0.0 0.0.0.0 172.16.4.1
!
!
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:   Fri Oct 31 15:22:03 2025

GlobalParameters(
    Title                              = "This LSP was automatically generated by CSP
Converter at Fri Oct 31 15:22:03 2025 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 172.16.4.1
```

```
        )

)

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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.4.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.4.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.4.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
```

```
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 172.16.4.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        DestinationIP = ::/0
        Action = DROP
    ))

FilterChain InspectChain:INSPECT (
    Filters = Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 6
        Action = PASS
        ExtendedAction = inspect_tcp<>
        LogEventID = "Inspect:INSPECT:tcp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        ProtocolID = 17
        Action = PASS
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        SourceIP = ::/0
        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 (
        SourceIP = 0.0.0.0/0
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
```

```
        SourceIP = ::/0
        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
)

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

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.10.0/24
        DestinationIP = 192.168.100.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_SPOKE4_HUB1"
    )
)

NetworkInterface (
```

```
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN
)

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

NetworkInterface (
    LogicalName = "GigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

Конфигурация Client1

1. Конфигурация LSP:

```
GlobalParameters (
    Title = "This LSP was automatically generated by S-Terra Client AdminTool (st) at
2025.11.10 14:04:28"
    Version = LSP_5_0
    CRLHandlingMode = BEST_EFFORT
)
LDAPSettings (
    ResponseTimeout = 200
    HoldConnectTimeout = 60
    DropConnectTimeout = 5
)
IdentityEntry auth_identity_01(
    DistinguishedName *= CertDescription(
        Subject *= COMPLETE,"C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1"
        )
)
CertDescription local_cert_dsc_01(
    Subject *= COMPLETE,"C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1"
    Issuer  *=  COMPLETE,"C=RU,L=Zelenograd,O=S-Terra  CSP,OU=RnD,CN=S-Terra  CSP  Test
Root CA"
    SerialNumber = "2F00005B65EA4888B6F348046B000000005B65"
    FingerprintMD5 = "22BA4AEB2E443BAF5EFEA81FD826A5A6"
)
CertDescription partner_cert_dsc_01(
)
AuthMethodGOSTSign auth_method_01(
    LocalID = auth_identity_01
    LocalCredential = local_cert_dsc_01
    RemoteCredential = partner_cert_dsc_01
    SendRequestMode = AUTO
    SendCertMode = AUTO
)
IKEParameters (
    DefaultPort = 500
    SendRetries = 5
    RetryTimeBase = 1
    RetryTimeMax = 30
    SessionTimeMax = 60
    InitiatorSessionsMax = 30
    ResponderSessionsMax = 20
```

```
    BlacklogSessionsMax = 16
    BlacklogSessionsMin = 0
    BlacklogSilentSessions = 4
    BlacklogRelaxTime = 120
    IKECFGPreferDefaultAddress = FALSE
)
IKETransform ike_trf_01(
    LifetimeSeconds = 28800
    CipherAlg  *= "GR341215M-K256-CFB-65526"
    HashAlg *= "GR341112_256TC26-65128"
    GroupID *= VKO2_1B
)
IKETransform ike_trf_02(
    LifetimeSeconds = 28800
    CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
    HashAlg *= "GR341112_256TC26-65128"
    GroupID *= VKO2_1B
)
IKETransform ike_trf_03(
    LifetimeSeconds = 28800
    CipherAlg  *= "GR341215K-K256-CFB-65528"
    HashAlg *= "GR341112_256TC26-65128"
    GroupID *= VKO2_1B
)
ESPTransform esp_trf_01(
    CipherAlg *= "GR341215M-K288-MGM-33"
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_01(
    Transform *=esp_trf_01
)
ESPTransform esp_trf_02(
    CipherAlg *= "G2814789CPRO2-K288-CNTMAC-253"
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_02(
    Transform *=esp_trf_02
)
ESPTransform esp_trf_03(
    CipherAlg *= "GR341215K-K352-MGM-251"
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_03(
    Transform *=esp_trf_03
)
IKERule ike_rule_with_ikecfg_01(
    DoNotUseDPD = FALSE
    DPDIdleDuration = 60
    DPDResponseDuration = 5
    DPDRetries = 3
    MainModeAuthMethod *= auth_method_01
    Transform *= ike_trf_01,ike_trf_02,ike_trf_03
    IKECFGRequestAddress = TRUE
)
IPsecAction ipsec_action_01(
    PersistentConnection = TRUE
    TunnelingParameters *=
        TunnelEntry(
            PeerAddress = 172.16.100.2
```

```
            Assemble = TRUE
            ReRoute = FALSE
            TCPEncapsulation = FALSE
        )
    ContainedProposals *= (esp_proposal_01),(esp_proposal_02),(esp_proposal_03)
    IKERule = ike_rule_with_ikecfg_01
)
FilterChain filter_chain_input(
    Filters *= Filter(
        ProtocolID *= 17
        DestinationPort *= 500
        Action = PASS
        LogEventID = "pass_action_02_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        ProtocolID *= 17
        DestinationPort *= 4500
        Action = PASS
        LogEventID = "pass_action_02_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        SourceIP *= 172.16.100.2
        ProtocolID *= 50
        Action = PASS
        LogEventID = "pass_action_03_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        SourceIP *= 172.16.100.2
        ProtocolID *= 51
        Action = PASS
        LogEventID = "pass_action_03_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        Action = PASS
        LogEventID = "pass_action_04"
    )
)
FilterChain filter_chain_output(
    Filters *= Filter(
        ProtocolID *= 17
        SourcePort *= 500
        Action = PASS
        LogEventID = "pass_action_05_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        ProtocolID *= 17
        SourcePort *= 4500
        Action = PASS
        LogEventID = "pass_action_05_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 50
        Action = PASS
        LogEventID = "pass_action_06_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 51
        Action = PASS
        LogEventID = "pass_action_06_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
```

```
    ),Filter(
        Action = PASS
        LogEventID = "pass_action_07"
    )
)
FilterChain filter_chain_classification_input(
    Filters *= Filter(
        Action = PASS
        LogEventID = "pass_action_08"
    )
)
FilterChain filter_chain_classification_output(
    Filters *= Filter(
        Action = PASS
        LogEventID = "pass_action_09"
    )
)
FilterChain filter_chain_ipsec(
    Filters *= Filter(
        ProtocolID *= 17
        SourcePort *= 500
        Action = PASS
        LogEventID = "pass_action_10_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        ProtocolID *= 17
        SourcePort *= 4500
        Action = PASS
        LogEventID = "pass_action_10_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 50
        Action = PASS
        LogEventID = "pass_action_11_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 51
        Action = PASS
        LogEventID = "pass_action_11_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 192.168.100.0/24
        Action = PASS
        ExtendedAction *= ipsec<sa=ipsec_action_01>
        LogEventID = "ipsec_action_01"
    ),Filter(
        Action = PASS
        LogEventID = "pass_action_12"
    )
)
NetworkInterface(
    InputFilter = filter_chain_input
    OutputFilter = filter_chain_output
    InputClassification = filter_chain_classification_input
    OutputClassification = filter_chain_classification_output
    IPsecPolicy = filter_chain_ipsec
)
```