Построение отказоустойчивого решения на базе протокола VRRP с резервированием каналов связи

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 защищаемыми подсетями центрального офиса и филиала. 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». Криптошлюзы в центральном офисе и филиале версии 5.0.

Для «С-Терра Шлюз» версии 5.0 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.

Для отказоустойчивости Шлюзов используется протокол VRRP, для резервирования каналов связи используется сервис `multiwan`.

Краткое описание сервиса multiwan

Сервис `multiwan` предназначен для резервирования каналов связи (Интернет-провайдеров) в режиме активный/резервный.

Сервис `multiwan` конфигурируется через файл `/etc/multiwan/config.yaml`.

Доступность канала определяется по доступности заданных хостов посредством протокола ICMP. В зависимости от режима, доступность канала определяется по доступности одного или всех хостов.

Сервис `multiwan` считает канал неработоспособным или работоспособным, если канал был недоступен или доступен несколько проверок подряд соответственно. Сервис `multiwan` считает состояние канала неизвестным, если со старта сервиса канал не определил свою доступность.

Если установлено определение доступности канала по всем хостам, то сервис `multiwan` считает канал неработоспособным, если при каждой проверке хотя бы один хост был недоступен. Если

установлено определение доступности канала по любому хосту, то сервис `multiwan` считает канал неработоспособным, если при каждой проверке были недоступны все хосты.

Сервис переключит на другой работоспособный канал в следующих случаях:

1) Текущий активный канал неработоспособен.
2) Если включен режим preempt и есть работоспособный канал с большим приоритетом.

Ограничения и особенности сервиса multiwan

1. Сервис `multiwan` может резервировать произвольные маршруты;
2. Маршруты других сервисов до той же подсети, что заданы в сервисе `multiwan`, могут быть удалены, если находятся в той же таблице маршрутизации;
3. При переключении канала автоматически происходит очистка текущих ISAKMP/IPsec туннелей, перезагрузка LSP и очистка таблицы трансляций NAT.
4. Каждый канал связи проверяется отдельно. Для проверки доступности каналов используются ICMP запросы. Для ICMP запросов сервиса `multiwan` используются отдельные таблицы маршрутизации. Данные таблицы будут содержать маршруты по умолчанию через адрес следующего перехода и будут очищаться от посторонних маршрутов.
5. Reverse path filtering должен быть отключен. Для проверки доступности хостов используются собственные таблицы отличные от основной таблицы маршрутизации. Поэтому проверки доступности будут неудачными при включенном reverse path filtering.

   По умолчанию данный параметр отключен на криптошлюзах «С-Терра Шлюз» ОС Debian 12 классы защиты КС1, КС2 и КС3.

   Пример проверки состояния reverse path filtering на интерфейсе `eth0`:

   ```
   root@Hub:~# cat /proc/sys/net/ipv4/conf/eth0/rp_filter
   0
   ```

   Ноль в выводе команды означает, что reverse path filtering отключен.

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, switching.

Администратор должен быть ознакомлен с базовым сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля», «Резервирование каналов связи при помощи сервиса MultiWAN» и «Построение кластера на базе протокола VRRP с настройкой при помощи CLI».

Требования к инфраструктуре

1. Требования к устройствам.

   1.1. «С-Терра Шлюз».

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

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

1.2. Центр выпуска сертификатов.

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

1.2.2 Для выпуска цифровых сертификатов допускается использование встроенного в ОС Windows Server 2016 (или новее) удостоверяющего центра совместно с сертифицированным СКЗИ «КриптоПро» CSP 5.0 R2 (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

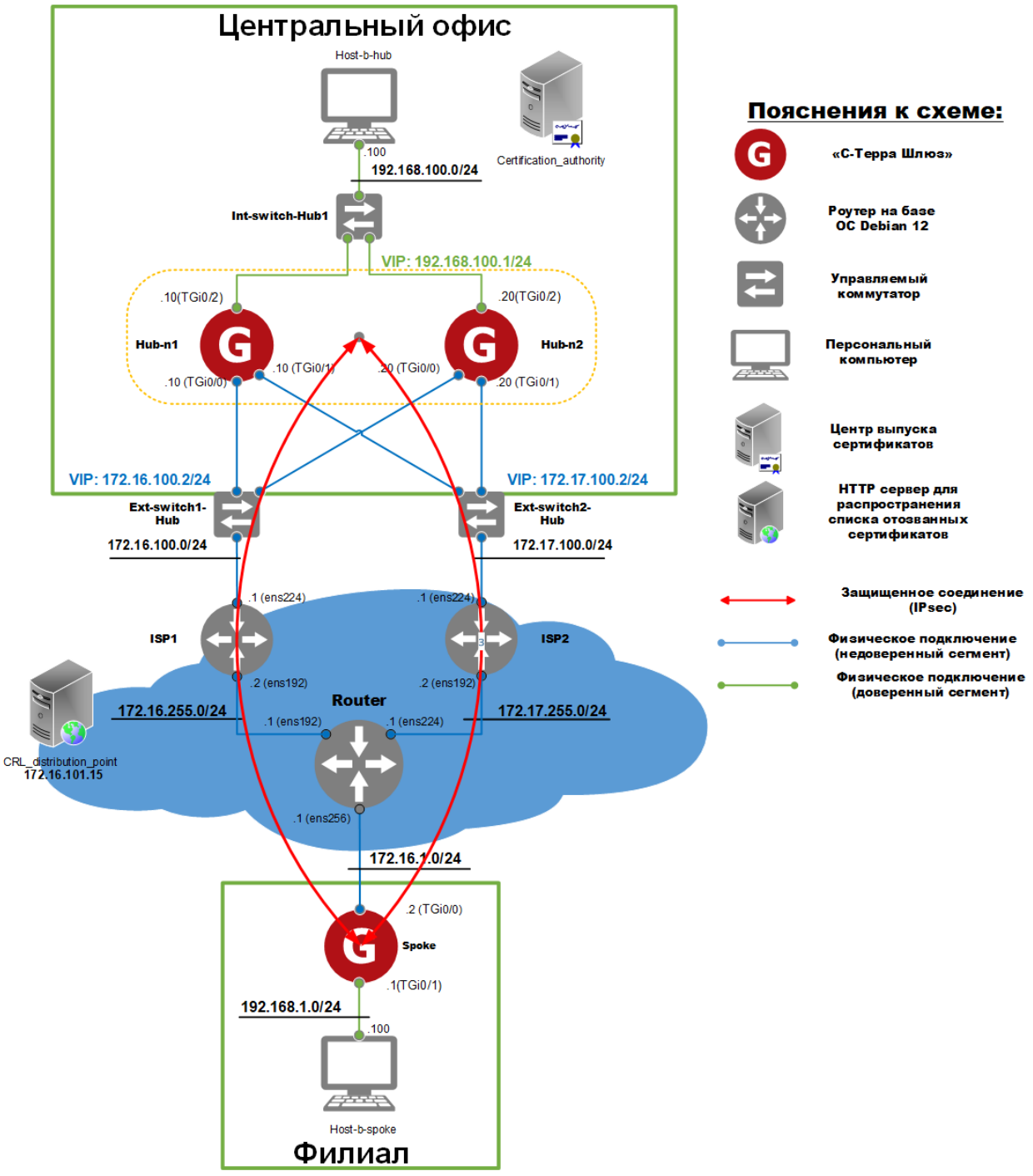


Рисунок 1. Общая схема

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), криптошлюзы «С-Терра Шлюз» (Hub-n1 и Hub-n2) версии 5.0 и персональный компьютер (Host-b-hub).

1.2. В филиале размещаются: криптошлюз «С-Терра Шлюз» (Spoke) версии 5.0 и персональный компьютер (Host-b-spoke).

1.3. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point) и 3 маршрутизатора (ISP1, ISP2 и Router).

2. Отказоустойчивость.

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

Конфигурация между нодами кластера не синхронизируется. При изменении настроек на одной ноде нужно выполнить аналогичные настройки на другой. Отличие между настройками в приоритетах и отслеживаемых интерфейсах.

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 (данное поведение отличается от Cisco IOS, где по умолчанию VRRP интерфейсы не синхронизированы).

Обнаружение отказа происходит благодаря служебным пакетам протокола VRRP (VRRP Advertisement messages). Служебные пакеты отправляются только нодой, находящейся в состоянии Master (адрес источника пакетов – основной IP адрес на интерфейсе, где включен VRRP; IP адрес назначения - 224.0.0.18). Если резервная нода в состоянии Backup перестает получать на всех синхронизированных интерфейсах пакеты VRRP Advertisement messages от основной ноды или получает пакеты с меньшим приоритетом, то она захватывает роль Master.

На нодах есть маршрут по умолчанию через адрес доверенного сегмента другой ноды (для доступа резервной ноды в сеть интернет). Метрика данного маршрута должна быть больше, чем метрика маршрутов, добавляемых с помощью `multiwan`.

Кластер обрабатывает следующие типы отказов:

- отключение питания;
- выход из строя аппаратной платформы;
- отказ сетевого порта на аппаратной платформе;
- отказ сетевого порта на коммутационном оборудовании;
- отказ сервиса `vpngate`, обеспечивающего построение IPsec туннелей.

За резервирования каналов связи отвечает сервис `multiwan`. `multiwan` добавляет маршрут по умолчанию через ISP1 или ISP2. Выбор зависит от доступности каналов. Если канал через ISP1 недоступен, то переключается на работу через ISP2, если он доступен. Если `multiwan` работает на канале через ISP2 и канал через ISP1 становится доступен, то `multiwan` переключится на работу через ISP1, если включен preempt режим.

Для проверки доступности каналов используются ICMP запросы. Для каждого проверяемого хоста и для каждого интерфейса создаётся своё правило для ICMP запроса. Для каждого канала создаётся своя таблица с маршрутом по умолчанию через ISP1 или через ISP2 с указанием src адреса. Данная таблица будет содержать только маршрут по умолчанию, все остальные маршруты из таблиц для каналов будут удалены.

Для маршрутов указывается src адрес: VIP адрес интерфейса. Если нода переходит в состояние BACKUP/FAULT, то маршрутов не будет в таблице маршрутизации.

В случае отказа обоих каналов связи предполагается, что они будут недоступны и на резервной ноде. Поэтому переключение на неё не выполняется.

3. Подключение к сети Интернет.

В данном сценарии для эмуляции сети Интернет используются маршрутизаторы ISP1, ISP2 и Router.

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). Из-за отказоустойчивости резервная нода кластера должна иметь доступ к сети Интернет через основную ноду.

3.1. Криптошлюзы Hub-n1 и Hub-n2 подключаются к сети Интернет с помощью сервиса `multiwan`. Основной маршрут по умолчанию через маршрутизатор ISP1. Резервный маршрут по умолчанию через маршрутизатор ISP2.

3.2. Криптошлюз Spoke подключается к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router).

4. Параметры шифрования IPsec.

Весь IPv4 трафик между подсетями центрального офиса (192.168.100.0/24) и филиала (192.168.1.0/24) защищается с использованием алгоритмов ГОСТ и протокола IPsec в туннельном режиме.

Инициировать защищенное соединение может как трафик из подсети 192.168.1.0/24 в 192.168.100.0/24 (из филиала в центр), так и наоборот, из центра в филиал.

4.1. Параметры протокола IKE для версии 5.0:

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

4.2. Параметры протокола ESP для версии 5.0:

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

5. Параметры межсетевого экранирования.

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

5.1. На LAN интерфейсах IPv4 трафик не контролируется.

5.2. На LAN интерфейсах IPv6 трафик заблокирован.

5.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

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

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

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

5.3.4 Multicast VRRP (номер 112) – для VRRP.

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

5.4. На WAN интерфейсах IPv6 трафик заблокирован.

5.5. 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).

Настройка стенда

Настройка устройства Host-b-hub

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 (если заблокировано).

Настройка устройства Host-b-spoke

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 (если заблокировано).

Настройка устройства ISP1

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

Настройка устройства ISP2

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

Настройка устройства Router

1. Настройте IP адрес – 172.16.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.17.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Настройте IP адрес – 172.16.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens256.
4. Задайте маршрут до сети 172.16.100.0/24 через 172.16.255.1.
5. Задайте маршрут до сети 172.17.100.0/24 через 172.17.255.1.
6. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Hub-n1

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

Начальные настройки

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

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

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройки PKI (запросы и сертификаты)

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

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

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка паролей доступа к консоли cisco-like

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка сетевых параметров

Изменение состояния интерфейсов и назначение IP адресов в cisco-like консоли применяются сразу после ввода соответствующих команд. Политика безопасности применяются после выхода из режима конфигурирования.

1. Задайте имя устройства:
   ```
   sterragate(config)#hostname Hub-n1
   ```
2. Включите WAN и LAN интерфейсы и настройте IP адрес на WAN интерфейсе в соответствии со схемой стенда:
   ```
   Hub-n1(config)#interface TenGigabitEthernet0/0
   Hub-n1(config-if)# ip address 172.16.100.10 255.255.255.0
   Hub-n1(config-if)# no shutdown
   Hub-n1(config-if)#interface TenGigabitEthernet0/1
   Hub-n1(config-if)# ip address 172.17.100.10 255.255.255.0
   Hub-n1(config-if)# no shutdown
   Hub-n1(config-if)#interface TenGigabitEthernet0/2
   Hub-n1(config-if)# ip address 192.168.100.10 255.255.255.0
   Hub-n1(config-if)# no shutdown
   Hub-n1(config-if)#exit
   ```

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка шифрования

1. Параметры IKE.

   1.1. Задайте общие параметры IKE:

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

   1.2. Создайте политику **1**, описывающую алгоритмы IKE туннеля для совместимости с версией 5.0:

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

   1.3. Опционально: создайте политику **2**, описывающую алгоритмы IKE туннеля для совместимости с версией 4.3:

```
Hub-n1(config)#crypto isakmp policy 2
Hub-n1(config-isakmp)# encryption gost
Hub-n1(config-isakmp)# hash gost341112-256-tc26
Hub-n1(config-isakmp)# authentication gost-sig
Hub-n1(config-isakmp)# group vko2
Hub-n1(config-isakmp)# exit
```

   1.4. Создайте профиль настроек ISAKMP (алгоритм IKE) под именем `NEW_ALGS` для совместимости с версией 5.0, чтобы далее его прикрепить к соответствующему разделу крипто-карты:

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

   1.5. Опционально: для совместимости с версией 4.3 создайте профиль настроек ISAKMP (алгоритм IKE) под именем `OLD_ALGS`, чтобы далее его прикрепить к соответствующему разделу крипто-карты:

```
Hub-n1(config)#crypto isakmp profile OLD_ALGS
Hub-n1(conf-isa-prof)#match identity dn
Hub-n1(conf-isa-prof)#set identity mapping
Hub-n1(conf-isa-prof)#set policy 2
```

2. Параметры IPsec.

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

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

   2.2. Создайте IPsec трансформ, задающий комбинированный алгоритм шифрования и имитозащиты ГОСТ 34.12-2015 Магма в режиме MGM для трафика (для версий 5.0):

```
Hub-n1(config)# crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-
magma-mgm
```

```
Hub-n1(cfg-crypto-trans)#exit
```

2.3. Опционально: для совместимости с 4.3 создайте IPsec трансформ, задающий комбинированный алгоритм шифрования и имитозащиты на базе ГОСТ 28147-89 для трафика:

```
Hub-n1(config)# crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
Hub-n1(cfg-crypto-trans)#exit
```

2.4. Создайте список доступа (ACL) для IPv4 трафика, который нужно защищать между центральным офисом и филиалом:

```
Hub-n1(config)#ip access-list extended IPSEC_ACL_HUB_AND_SPOKE
Hub-n1(config-ext-nacl)#permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub-n1(config-ext-nacl)#exit
Hub-n1(config)#
```

2.5. Создайте крипто-карту (имя VPN_ISP1, раздел 1) для построения IPsec туннеля через провайдера ISP1:

```
Hub-n1(config)#crypto map VPN_ISP1 1 ipsec-isakmp
Hub-n1(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Hub-n1(config-crypto-map)#set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
Hub-n1(config-crypto-map)#set local-address 172.16.100.2
```

Указание локального адреса служит для отправки шифрованного трафика с адресом интерфейса `TenGigabitEthernet0/0`.

```
Hub-n1(config-crypto-map)#set peer 172.16.1.2
Hub-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub-n1(config-crypto-map)#set dead-connection history off
Hub-n1(config-crypto-map)#set pre-fragmentation
Hub-n1(config-crypto-map)#set security-association mss-fix out
Hub-n1(config-crypto-map)#exit
```

2.6. Опционально: для совместимости с 4.3 используйте криптокарту со следующими параметрами:

```
Hub-n1(config)#crypto map VPN_ISP1 1 ipsec-isakmp
Hub-n1(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Hub-n1(config-crypto-map)#set transform-set GOST_ENCRYPT_AND_INTEGRITY
Hub-n1(config-crypto-map)#set local-address 172.16.100.2
Hub-n1(config-crypto-map)#set peer 172.16.1.2
Hub-n1(config-crypto-map)#set isakmp-profile OLD_ALGS
Hub-n1(config-crypto-map)#set dead-connection history off
Hub-n1(config-crypto-map)#set pre-fragmentation
Hub-n1(config-crypto-map)#set security-association mss-fix out
Hub-n1(config-crypto-map)#exit
```

2.7. Создайте крипто-карту (имя VPN_ISP2, раздел 1) для построения IPsec туннеля через провайдера ISP2:

```
Hub-n1(config)#crypto map VPN_ISP2 1 ipsec-isakmp
Hub-n1(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Hub-n1(config-crypto-map)#set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
Hub-n1(config-crypto-map)#set local-address 172.17.100.2
```

Указание локального адреса служит для отправки шифрованного трафика с адресом интерфейса `TenGigabitEthernet0/1`.

```
Hub-n1(config-crypto-map)#set peer 172.16.1.2
Hub-n1(config-crypto-map)#set isakmp-profile NEW_ALGS
```

```
Hub-n1(config-crypto-map)#set dead-connection history off
Hub-n1(config-crypto-map)#set pre-fragmentation
Hub-n1(config-crypto-map)#set security-association mss-fix out
Hub-n1(config-crypto-map)#exit
```

2.8. Опционально: для совместимости с 4.3 используйте криптокарту со следующими параметрами:

```
Hub-n1(config)#crypto map VPN_ISP2 1 ipsec-isakmp
Hub-n1(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Hub-n1(config-crypto-map)#set transform-set GOST_ENCRYPT_AND_INTEGRITY
Hub-n1(config-crypto-map)#set local-address 172.17.100.2
Hub-n1(config-crypto-map)#set peer 172.16.1.2
Hub-n1(config-crypto-map)#set isakmp-profile OLD_ALGS
Hub-n1(config-crypto-map)#set dead-connection history off
Hub-n1(config-crypto-map)#set pre-fragmentation
Hub-n1(config-crypto-map)#set security-association mss-fix out
Hub-n1(config-crypto-map)#exit
```

3. Прикрепите созданные крипто-карты к внешним интерфейсам `TenGigabitEthernet0/0` и `TenGigabitEthernet0/1`:

```
Hub-n1(config)#interface TenGigabitEthernet0/0
Hub-n1(config-if)#crypto map VPN_ISP1
Hub-n1(config-if)#interface TenGigabitEthernet0/1
Hub-n1(config-if)#crypto map VPN_ISP2
Hub-n1(config-if)#exit
```

4. Примените введенные настройки:

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка политики обработки COC (CRL)

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

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Из-за отсутствия маршрута по умолчанию COC будет не загружен в базу Продукта. Проверка осуществится после настройки сервиса `multiwan`.

Настройка межсетевого экрана

Настройка межсетевого экрана осуществляется из cisco-like консоли.

По умолчанию на всех сетевых интерфейсах IPv6 трафик заблокирован – команда `ipv6 default filtering deny`, если требуется разрешить IPv6 трафик, то нужно выполнить команду `ipv6 default filtering permit` и далее уже фильтровать IPv6 трафик по аналогии с IPv4.

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

```
Hub-n1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Hub-n1(config)#object-group network SELF
Hub-n1(config-network-group)# remark WAN IP
Hub-n1(config-network-group)# host 172.16.100.2
Hub-n1(config-network-group)# host 172.17.100.2
Hub-n1(config-network-group)# host 172.16.100.10
Hub-n1(config-network-group)# host 172.17.100.10
Hub-n1(config-network-group)#exit
```

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

```
Hub-n1(config)#ip access-list extended FIREWALL_IN
Hub-n1(config-ext-nacl)# permit udp any object-group SELF eq 500
Hub-n1(config-ext-nacl)# permit udp any object-group SELF eq 4500
Hub-n1(config-ext-nacl)# permit icmp any object-group SELF
Hub-n1(config-ext-nacl)# permit esp any object-group SELF
Hub-n1(config-ext-nacl)# permit vrrp any host 224.0.0.18
Hub-n1(config-ext-nacl)# exit
```

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

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

4. Прикрепите правила межсетевого экранирования на WAN интерфейсы `TenGigabitEthernet0/0` и `TenGigabitEthernet0/1`:

```
Hub-n1(config)# interface TenGigabitEthernet 0/0
Hub-n1(config-if)# ip access-group FIREWALL_IN in
Hub-n1(config-if)# ip inspect INSPECT out
Hub-n1(config-if)#exit
Hub-n1(config)# interface TenGigabitEthernet 0/1
Hub-n1(config-if)# ip access-group FIREWALL_IN in
Hub-n1(config-if)# ip inspect INSPECT out
Hub-n1(config-if)#exit
```

5. Примените настройки:

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка VRRP

1. Настройки VRRP осуществляются из cisco-like консоли шлюза. Настройте конфигурирование третьей версии VRRP.

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

2. Включите VRRP на `TenGigabitEthernet0/0` (создает VRRP инстанс):

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

3. Задайте кластерный IP адрес:

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

4. Выключите отслеживание состояния интерфейса:

```
Hub-n1(config-if-vrrp)# no track interface primary
```

5. Установите интервал отправки advertisement-сообщений:

```
Hub-n1(config-if-vrrp)# timers advertise 3
```

6. Установите интервал отправки gratuitous ARP-сообщения:

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

7. Установите приоритет для VRRP инстанса:

```
Hub-n1(config-if-vrrp)# priority 100
Hub-n1(config-if-vrrp)# exit
```

8. Аналогично включите VRRP на `TenGigabitEthernet0/1` и `TenGigabitEthernet0/2`. У каждого VRRP ID и свой кластерный адрес:

```
Hub-n1(config-if)# interface TenGigabitEthernet 0/1
Hub-n1(config-if)# vrrp 2 address-family ipv4
Hub-n1(config-if-vrrp)# address 172.17.100.2 255.255.255.0
Hub-n1(config-if-vrrp)# no track interface primary
Hub-n1(config-if-vrrp)# timers advertise 3
Hub-n1(config-if-vrrp)# timers garp 5
Hub-n1(config-if-vrrp)# priority 100
Hub-n1(config-if-vrrp)# exit
Hub-n1(config-if)# interface TenGigabitEthernet 0/2
Hub-n1(config-if)# vrrp 3 address-family ipv4
Hub-n1(config-if-vrrp)# address 192.168.100.1 255.255.255.0
Hub-n1(config-if-vrrp)# no track interface primary
Hub-n1(config-if-vrrp)# timers advertise 3
Hub-n1(config-if-vrrp)# timers garp 5
Hub-n1(config-if-vrrp)# priority 100
```

9. Для `TenGigabitEthernet0/2` включите отслеживание внешних интерфейсов с весом -40. Данный вес будет уменьшать приоритет ноды на 40 при переходе внешнего интерфейса в состояние DOWN:

Приоритет изменится только для пакетов, отправленных с `TenGigabitEthernet 0/2`. Для изменения приоритета на других интерфейса, следует также включить отслеживание интерфейсов с весом.

```
Hub-n1(config-if-vrrp)# track interface TenGigabitEthernet0/0 weight -40
Hub-n1(config-if-vrrp)# track interface TenGigabitEthernet0/1 weight -40
Hub-n1(config-if-vrrp)# exit
Hub-n1(config-if)# exit
```

При отказе одного интерфейса приоритет ноды станет 60, при отказе второго интерфейса 20. Приоритет не может стать ниже 1 и больше 255.

10. Установите маршрут по умолчанию с метрикой 1001 через адрес внутреннего интерфейса другой ноды. Данная метрика больше, чем метрика `multiwan` маршрутов. В случае перехода ноды в состояние BACKUP/FAILT доступ в Интернет будет происходить через ноду в состоянии MASTER:

```
Hub-n1(config)# ip route 0.0.0.0 0.0.0.0 192.168.100.20 1001
Hub-n1(config)# end
```

11. Убедитесь, что нода перешла в состояние MASTER. Для этого войдите в cisco-like консоль шлюза и посмотрите настройки VRRP:

```
Hub-n1#show vrrp
```

```
Interface                     VRID  State   AddrFamily
------------------------------------------------------
TenGigabitEthernet0/0         1     Master  IPv4
TenGigabitEthernet0/1         2     Master  IPv4
TenGigabitEthernet0/2         3     Master  IPv4
```

12. Убедитесь, что на интерфейсах появился кластерный VIP-адрес:

```
Hub-n1#run ip address show dev eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:50:56:9e:da:3d brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname ens160
    altname TenGigabitEthernet0_0
    inet 172.16.100.10/24 brd 172.16.100.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 172.16.100.2/24 scope global secondary eth0:900
       valid_lft forever preferred_lft forever
```

```
Hub-n1#run ip address show dev eth1
```

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:50:56:9e:ee:35 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname TenGigabitEthernet0_1
    inet 172.17.100.10/24 brd 172.17.100.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet 172.17.100.2/24 scope global secondary eth1:900
       valid_lft forever preferred_lft forever
```

```
Hub-n1#run ip address show dev eth2
```

```
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:50:56:9e:78:4a brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    altname TenGigabitEthernet0_2
    inet 192.168.100.10/24 brd 192.168.100.255 scope global eth2
       valid_lft forever preferred_lft forever
    inet 192.168.100.1/24 scope global secondary eth2:900
       valid_lft forever preferred_lft forever
```

Настройка сервиса multiwan

При конфигурировании сервиса `multiwan` нужно **обязательно** соблюдать следующие правила:

1) Конфигурационный файл должен быть формата YAML.
2) Версия формата конфигурационного файла должны быть 1.
3) Обязательно должны быть указаны как минимум два канала связи, приоритет канала связи, маршруты и хосты для проверки.
4) Приоритеты каналов связи должны отличаться.
5) Каждый канал связи должен содержать название интерфейса, приоритет канала связи, маршруты, адрес следующего перехода и хосты для проверки.
6) Для каждой IP версии хоста для проверки должен быть указан соответствующий IP версии адрес следующего перехода.

Настройка сервиса `multiwan` осуществляются из linux bash, чтобы попасть в linux bash выполните следующую команду в CLI разграничения доступа (предварительно нужно выйти из консоли cisco-like):

```
Hub-n1#exit
administrator@Hub-n1] system
```

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

Настройка `multiwan` осуществляется через файл `/etc/multiwan/config.yaml`. Приведите конфигурационный файл к следующему виду:

Если в хостах указаны domain-name, то механизм `watch_rule` должен быть включен.
При расположении параметра не в своей секции, значение параметра будет проигнорировано.

```
config_format_version: 1
links_params:
  links:
  - name: ISP1
    iface_name: eth0
    prio: 300
    routes:
        ipv4:
            networks:
            - 0.0.0.0/0
            nexthop: 172.16.100.1
            source_address: 172.16.100.2
        watch_foreign_routes: false
        watch_self_routes: true
    check:
        hosts_reachability: any
        ipv4:
            hosts:

            - 172.16.101.15
            - 172.16.1.2
        method:
            icmp:
                max_rtt: 100
                interval: 3
                restore_interval: 1
                restore_count: 10
                failure_count: 10
                tos: 184
                watch_rules: true
  - name: ISP2
    iface_name: eth1
    prio: 200
    routes:
        ipv4:
            networks:
            - 0.0.0.0/0
            nexthop: 172.17.100.1
            source_address: 172.17.100.2
        watch_foreign_routes: false
        watch_self_routes: true
    check:
        hosts_reachability: any
        ipv4:
            hosts:
            - 172.16.1.2
            - 172.16.1.1
        method:
            icmp:
                max_rtt: 100
                interval: 3
                restore_interval: 1
                restore_count: 10
                failure_count: 10
```

```
                tos: 184
                watch_rules: true
```

Задаются параметры для двух каналов с именами ISP1 и ISP2. Для ISP1 задаётся интерфейс eth0, для ISP2 интерфейс eth1. Канал через ISP1 имеет приоритет 300. Канал через ISP2 имеет приоритет 200. Канал через ISP1 считается более приоритетным.

При работе через канал ISP1 будет добавляться маршрут по умолчанию (0.0.0.0/0) через 172.16.100.1 с адресом отправителя 172.16.100.2 с метрикой 700. При работе через канал ISP2 будет добавляться маршрут по умолчанию через 172.17.100.1 с адресом отправителя 172.17.100.2 с метрикой 800. Расчёт метрики основывается от приоритета и равен 1000 минус значение приоритета.

И для работы через канал ISP1, и для работы через канал ISP2:

- включен механизм отслеживания маршрутов `mutliwan`. Отключено отслеживание и удаление внешних маршрутов по умолчанию;
- для проверки доступности каналов используют IP адрес HTTP сервера CRL_distribution_point и внешний IP адрес Spoke. Канал считается работающим, если получен ответ от любого хоста;
- для проверки используют ICMP. Время ожидания ICMP ответа 100 миллисекунды (`max_rtt`). Время ожидания 3 секунды между попыткой получить ICMP ответ от последнего хоста и повторной отправкой ICMP запроса первому хосту (`interval`). Если канал недоступен и был получен ICMP ответ, то время между попытками 1 секунда (`restore_interval`). Если 10 раз подряд канал считался неработающим, то канал считается недоступным (`failure_count`). Если 10 раз подряд канал считался работающим, то канал считается доступным (`restore_count`). Включен механизм отслеживания маршрутов и правил маршрутизации, рассчитанных на ICMP запросы (`watch_rules`).

Ограничения использования параметров:

- Если оба канала оказываются недоступны и при этом включён параметр `clear_routes_if_all_links_unreachable`, на ноде автоматически удаляются все маршруты, установленные сервисом `multiwan`. В таком случае пакеты отправятся на резервную ноду и будут отброшены по причине превышения TTL.
- Если включен параметр `watch_foreign_routes`, маршруты, добавленные другими сервисами будут удалены. Для стабильной работы VRRP рекомендуется использовать `watch_self_routes: true`, а `watch_foreign_routes` оставлять в значении `false`, чтобы не затрагивать маршруты, установленные самим VRRP.

Подробная настройка с описанием команд приведена в сценарии «Резервирование каналов связи при помощи сервиса multiWAN».

Настройка трансляции адресов источника (source NAT)

Настройка source NAT осуществляется при необходимости доступа в Интернет резервной ноде кластера. Трансляция адресов источника осуществляется в кластерный VIP адрес 172.16.100.2 для трафика, выходящего с внешнего интерфейса `TenGigabitEthernet0/0`, который соответствует `eth0`. Трансляция адресов источника осуществляется в кластерный VIP адрес 172.17.100.2 для трафика, выходящего с внешнего интерфейса `TenGigabitEthernet0/1`, который соответствует `eth1`. Соответствие между именем интерфейса в нотации cisco-like и linux можно посмотреть в файле `/etc/ifaliases.cf`.

Важно:
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 «Использование iptables на шлюзе»).

1. Настройка source NAT осуществляются из linux bash. Посмотрите содержимое карты интерфейсов и найдите интерфейс в нотации linux для внешнего интерфейса `TenGigabitEthernet0/0`:

```
root@Hub-n1:~# cat /etc/ifaliases.cf
```

```
interface (name="TenGigabitEthernet0/0" pattern="eth0")
interface (name="TenGigabitEthernet0/1" pattern="eth1")
interface (name="TenGigabitEthernet0/2" pattern="eth2")
interface (name="TenGigabitEthernet0/3" pattern="eth3")
interface (name="default" pattern="*")
```

Видно, что интерфейсу `TenGigabitEthernet0/0` соответствует `eth0`, `TenGigabitEthernet0/1` соответствует `eth1`.

2. Очистите настройки `iptables`:

```
root@Hub-n1:~# netfilter-persistent flush
```

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

3. Добавьте две цепочки в таблицу NAT:

```
root@Hub-n1:~# iptables -N NO_SOURCE_NAT -t nat
root@Hub-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.

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

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

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

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

```
root@Hub-n1:~# iptables -t nat -A NO_SOURCE_NAT -m mark --mark
0x8000000/0x8000000 -j ACCEPT -m comment --comment 'local IKE packets'
root@Hub-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.

5.2. Выключите source NAT для пакетов протокола VRPP:

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

5.3. Выключите source NAT для трафика, который должен шифроваться:

```
root@Hub-n1:~# iptables -t nat -A NO_SOURCE_NAT -s 192.168.100.0/24 -d
192.168.1.0/24 -j ACCEPT -m comment --comment 'traffic which must be protected
by IPsec'
```

6. Добавьте `iptables` правила в цепочку SOURCE_NAT для трафика через eth0, для которого нужно делать source NAT:

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

```
root@Hub-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@Hub-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.

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

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

7. Добавьте `iptables` правила в цепочку SOURCE_NAT для трафика через eth1, для которого нужно делать source NAT:

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

```
root@Hub-n1:~# iptables -t nat -A SOURCE_NAT -p udp --sport 500 -o eth1 -j SNAT
--to-source 172.17.100.2:40500-40540 -m comment --comment 'transit IKE packets'
root@Hub-n1:~# iptables -t nat -A SOURCE_NAT -p udp --sport 4500 -o eth1 -j SNAT
--to-source 172.17.100.2:44500-44540 -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.

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

```
root@Hub-n1:~# iptables -t nat -A SOURCE_NAT -s 192.168.100.0/24 -o eth1 -j SNAT
--to-source 172.17.100.2
```

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

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

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

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

```
root@Hub-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
```

10. Для просмотра текущих NAT правил воспользуйтесь командой:

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

11. Для просмотра текущей таблицы NAT трансляций воспользуйтесь командой:

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

Завершение настройки Hub-n1

1. После настройки запустите сервис `multiwan`, а также добавьте его в автозагрузку:

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

2. Посмотреть лог работы сервиса можно в файле `/var/log/cspvpngate.log`:

```
root@Hub:~# tail -n 1000 /var/log/cspvpngate.log | grep multiwan
```

```
2023-12-04T15:10:14.853421+03:00 Hub-n1 multiwan: INFO: Starting multiwan...
2023-12-04T15:10:15.908671+03:00 Hub-n1 multiwan: INFO: NAT translations table
has been cleared.
2023-12-04T15:10:16.923643+03:00 Hub-n1 multiwan: INFO: LSP was reloaded.
2023-12-04T15:10:16.930806+03:00 Hub-n1 multiwan: INFO: Used interface "eth0".
Nexthop "172.16.100.1".
```

3. Проверьте, что добавлены маршруты объявлены в настройках `multiwan`:

```
root@Hub:~# ip route show proto multiwan
```

```
default via 172.16.100.1 dev eth0 src 172.16.100.2 metric 700
```

4. Проверьте что маршрут до IP адреса HTTP сервера CRL_distribution_point в системе:

```
root@Hub:~# ip route get 172.16.101.15
```

```
172.16.101.15 via 172.16.100.1 dev eth0 src 172.16.100.2 uid 0
    cache
```

5. Если всё настроено верно, то перезагрузите сервис `getcrls` для загрузки СОС в базу Продукта:

```
root@Hub:~# systemctl restart getcrls.service
```

6. Выполните проверку статуса сертификатов в базе Продукта:

```
root@Hub_n1:~# 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=Hub-n1
```

В Приложении представлены тексты конфигураций для криптошлюза Hub-n1:

- текст консоли cisco-like;
- текст LSP;
- Содержимое `/etc/multiwan/config.yaml`.

Настройка криптошлюза Hub-n2

Настройка криптошлюза Hub-n2 производится по аналогии с Hub-n1. В настройках VRRP для внутреннего интерфейса не отслеживаются внешние интерфейсы.

В Приложении представлены тексты конфигураций для криптошлюза Hub-n2:

- текст консоли cisco-like;
- текст LSP;
- Содержимое `/etc/multiwan/config.yaml`.

Настройка криптошлюза Spoke

Настройка криптошлюза Spoke производится по аналогии с Hub-n1 без настроек VRRP, `multiwan` и трансляции адресов. В статической криптокарте должно быть указано два партнера по IPsec соединению:

```
Spoke(config)#crypto map VPN 1 ipsec-isakmp
Spoke(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Spoke(config-crypto-map)#set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
Spoke(config-crypto-map)#set isakmp-profile NEW_ALGS
Spoke(config-crypto-map)#set security-association mss-fix out
Spoke(config-crypto-map)#set peer 172.16.100.2
Spoke(config-crypto-map)#set peer 172.17.100.2
Spoke(config-crypto-map)#set pre-fragmentation
Spoke(config-crypto-map)#set dead-connection history off
```

В Приложении представлены тексты конфигураций для криптошлюза Spoke:

- текст консоли cisco-like;
- текст LSP.

Проверка работоспособности стенда

Проверку работоспособности нужно выполнять после завершения настройки всех устройств стенда.

Проверка IP связности

1. Проверьте, что с криптошлюзов Hub-n1, Hub-n2 и Spoke по ICMP доступны шлюзы по умолчанию (ISP1 и ISP2 для Hub-n1 и Hub-n2, Router для Spoke). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub-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=0.438 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.465 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.362 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.472 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.301 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.301/0.407/0.472/0.066 ms
```

```
Hub-n1#ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 100(128) bytes of data.
108 bytes from 172.17.100.1: icmp_seq=1 ttl=64 time=0.540 ms
108 bytes from 172.17.100.1: icmp_seq=2 ttl=64 time=0.278 ms
108 bytes from 172.17.100.1: icmp_seq=3 ttl=64 time=1.06 ms
108 bytes from 172.17.100.1: icmp_seq=4 ttl=64 time=0.680 ms
108 bytes from 172.17.100.1: icmp_seq=5 ttl=64 time=0.625 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4032ms
rtt min/avg/max/mdev = 0.278/0.637/1.063/0.253 ms
```

```
Hub-n2#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=0.723 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.461 ms
```

```
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.300 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.289 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.302 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.289/0.415/0.723/0.166 ms
```

```
Hub-n2#ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 100(128) bytes of data.
108 bytes from 172.17.100.1: icmp_seq=1 ttl=64 time=0.443 ms
108 bytes from 172.17.100.1: icmp_seq=2 ttl=64 time=0.328 ms
108 bytes from 172.17.100.1: icmp_seq=3 ttl=64 time=0.392 ms
108 bytes from 172.17.100.1: icmp_seq=4 ttl=64 time=0.299 ms
108 bytes from 172.17.100.1: icmp_seq=5 ttl=64 time=0.417 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.299/0.375/0.443/0.054 ms
```

```
Spoke#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=1.11 ms
108 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.503 ms
108 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.366 ms
108 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.448 ms
108 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.283 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4062ms
rtt min/avg/max/mdev = 0.283/0.541/1.107/0.292 ms
```

Видно, что устройства ISP1 и ISP2 доступно по ICMP с Hub-n1 b Hub-n2, а устройство Router со Spoke.

2. Проверьте, что с защищаемых устройств Host-b-hub и Host-b-spoke доступны по ICMP соответствующие криптошлюзы (Hub-n1 и Hub-n2 для Host-b-hub, Spoke для Host-b-spoke), которые могут являться шлюзом по умолчанию. Для этого выполните команду `ping` из linux bash консоли защищаемых устройств.

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

```
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=63 time=1119 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=63 time=132 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=63 time=1.45 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=63 time=1.12 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=63 time=1.19 ms

--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 1.121/250.955/1119.290/437.098 ms, pipe 2
```

```
root@Host-b-hub:~# ping 192.168.100.10 -c 5
```

```
PING 192.168.100.10 (192.168.100.10) 56(84) bytes of data.
64 bytes from 192.168.100.10: icmp_seq=1 ttl=63 time=0.894 ms
64 bytes from 192.168.100.10: icmp_seq=2 ttl=63 time=1.56 ms
64 bytes from 192.168.100.10: icmp_seq=3 ttl=63 time=1.00 ms
64 bytes from 192.168.100.10: icmp_seq=4 ttl=63 time=1.07 ms
64 bytes from 192.168.100.10: icmp_seq=5 ttl=63 time=1.29 ms
```

```
--- 192.168.100.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.894/1.161/1.557/0.235 ms
```

```
root@Host-b-hub:~# ping 192.168.100.20 -c 5
```

```
PING 192.168.100.20 (192.168.100.20) 56(84) bytes of data.
64 bytes from 192.168.100.20: icmp_seq=1 ttl=62 time=1.66 ms
64 bytes from 192.168.100.20: icmp_seq=2 ttl=62 time=1.13 ms
64 bytes from 192.168.100.20: icmp_seq=3 ttl=62 time=1.35 ms
64 bytes from 192.168.100.20: icmp_seq=4 ttl=62 time=1.29 ms
64 bytes from 192.168.100.20: icmp_seq=5 ttl=62 time=1.19 ms

--- 192.168.100.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.125/1.324/1.664/0.187 ms
```

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

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.286 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.220 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.225 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.281 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.294 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.220/0.261/0.294/0.031 ms
```

Видно, что с Host-b-hub доступны по ICMP криптошлюзы Hub-n1 и Hub-n2 и адрес маршрута по умолчанию. С Host-b-hub доступен по ICMP криптошлюз Spoke.

3. Проверьте, что с криптошлюзов Hub-n1 и Hub-n2 доступен по ICMP криптошлюз Spoke. Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub-n1#ping 172.16.1.2
```

```
PING 172.16.1.2 (172.16.1.2) 100(128) bytes of data.
108 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=0.748 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=0.931 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=0.887 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=0.803 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=1.21 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.748/0.916/1.213/0.161 ms
```

```
Hub-n2#ping 172.16.1.2
```

```
PING 172.16.1.2 (172.16.1.2) 100(128) bytes of data.
108 bytes from 172.16.1.2: icmp_seq=1 ttl=61 time=1.11 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=61 time=1.02 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=61 time=1.48 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=61 time=1.13 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=61 time=1.02 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.017/1.150/1.475/0.168 ms
```

Видно, что с криптошлюзов Hub-n1 и Hub-n2 доступен по ICMP криптошлюз Spoke.

4. Проверьте наличие маршрутов через основной интерфейс на основной ноде

```
Hub-n1#run ip route show proto multiwan
```

```
default via 172.16.100.1 dev eth0 src 172.16.100.2 metric 700
```

Проверка PKI

1. Проверьте, что на криптошлюзах Hub-n1, Hub-n2 и Spoke СОС импортирован в базу Продукта. Для этого выполните команду `run cert_mgr show` из cisco-like консоли криптошлюзов.

```
Hub-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=Hub-n1
3 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

```
Hub-n2#run cert_mgr show
```

```
Found 2 certificates. Found 1 CRL.
1 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub-n2
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
```

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

   Видно, что на криптошлюзах Hub-n1, Hub-n2 и Spoke СОС импортирован в базу Продукта. Если этого не произошло, то проверьте файл журнала (команда `run grep getcrls_daemon /var/log/cspvpngate.log`) и, при необходимости, перезапустите сервис автоматической загрузки СОС (команда `run systemctl restart getcrls.service`).

2. Проверьте, что на криптошлюзах Hub-n1, Hub-n2 и Spoke статус всех сертификатов Active. Для этого выполните команду `run cert_mgr check` из cisco-like консоли криптошлюзов.

```
Hub-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=Hub-n1
```

```
Hub-n2#run cert_mgr check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub-n2
2 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root
CA
```

```
Spoke#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=Spoke
```

   Видно, что на криптошлюзах Hub-n1, Hub-n2 и Spoke статус всех сертификатов Active. Если статус Inactive, то проверьте загружен ли СОС в базу Продукта и правильность установки даты и времени.

Проверка построения защищенного соединения между подсетями Hub-n1 и Spoke

1. Инициируйте построение защищенного соединения между криптошлюзами Hub-n1 и Spoke при помощи ICMP трафика, посылаемого с устройства Host-b-spoke на Host-b-hub (можно и наоборот). Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke.

```
root@Host-b-spoke:~# ping 192.168.100.100 -c 5
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=1 ttl=62 time=403 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=28.5 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=2.14 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=1.65 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=2.14 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.645/87.499/403.100/158.133 ms
```

2. Проверьте, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub-n1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.100.2,500)-(172.16.1.2,500) active 1936 1968

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn
440 440
```

```
Hub-n2#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded
```

```
Spoke#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.1.2,500)-(172.16.100.2,500) active 1968 1936

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn
440 440
```

Видно, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение через основной канал. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка состояния каналов связи

Проверьте с помощью средств `multiwan` состояние каналов связи:

```
Hub-n1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: ISP1
    Interface: eth0
    Priority: 300
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
```

Видно, что первый канал связи, через eth0 с приоритетом 300 доступен. Является активным. Доступность хостом проверяется доступностью любого хоста и заданы IPv4 хосты. Второй канал связи, через eth1 с приоритетом 200 доступен. Является неактивным. Доступность хостом проверяется доступностью любого хоста и заданы IPv4 хосты.

Проверка работоспособности при отказе основного канала связи.

Стоит выделить, что канал связи определяется как выбором ноды, так и выбором маршрутизатора интернет-провайдера.

Считаем, что основной канал связи использует ISP1.

1. Для эмуляции отказа основного канала связи отключим интерфейс с подсетью 172.16.100.0/24 на ISP1. Для этого выполните команду `ifdown` из linux bash консоли устройства ISP1.

```
root@ISP1:~# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 00:50:56:9e:6d:1b brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 172.16.255.1/24 brd 172.16.255.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9e:6d1b/64 scope link
       valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
```

```
    link/ether 00:50:56:9e:fd:17 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 172.16.100.1/24 brd 172.16.100.255 scope global ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9e:fd17/64 scope link
       valid_lft forever preferred_lft forever
```

```
root@ISP1:~# ifdown ens224
```

2. Подождите пока переключится работа `multiwan` через резервный канал. Проверьте наличие маршрутов через резервный интерфейс.

```
Hub-n1#run ip route show proto multiwan
```

```
default via 172.17.100.1 dev eth1 src 172.17.100.2 metric 800
```

3. Пустите трафик с устройства Host-b-spoke на устройство Host-b-hub. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke.

```
root@Host-b-spoke:~# ping 192.168.100.100 -c 30
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=19 ttl=62 time=10029 ms
64 bytes from 192.168.100.100: icmp_seq=20 ttl=62 time=9005 ms
64 bytes from 192.168.100.100: icmp_seq=21 ttl=62 time=7981 ms
64 bytes from 192.168.100.100: icmp_seq=22 ttl=62 time=6957 ms
64 bytes from 192.168.100.100: icmp_seq=23 ttl=62 time=5933 ms
64 bytes from 192.168.100.100: icmp_seq=24 ttl=62 time=4909 ms
64 bytes from 192.168.100.100: icmp_seq=25 ttl=62 time=3885 ms
64 bytes from 192.168.100.100: icmp_seq=26 ttl=62 time=2861 ms
64 bytes from 192.168.100.100: icmp_seq=29 ttl=62 time=1.32 ms
64 bytes from 192.168.100.100: icmp_seq=30 ttl=62 time=1.16 ms

--- 192.168.100.100 ping statistics ---
30 packets transmitted, 10 received, 66.6667% packet loss, time 29632ms
rtt min/avg/max/mdev = 1.159/5156.007/10028.635/3323.677 ms, pipe 10
```

Такое поведение связанно с настройками DPD.

4. Проверьте, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub-n1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 19 (172.17.100.2,500)-(172.16.1.2,500) active 1936 1968

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1  8  (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*)  *  ESP  tunn
880 880
```

```
Spoke#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 15 (172.16.1.2,500)-(172.17.100.2,500) active 1968 1936

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
```

```
1  8  (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*)  *  ESP  tunn
880 880
```

Видно, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение через резервный канал. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

5. Проверьте с помощью средств `multiwan` состояние каналов связи:

```
Hub-n1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: ISP1
    Interface: eth0
    Priority: 300
    State: unreachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
```

Видно, что первый канал связи, через eth0 с приоритетом 300 недоступен и является неактивным. Второй канал связи, через eth1 с приоритетом 200 доступен и является активным.

6. Включите интерфейс с подсетью 172.16.100.0/24 на ISP1.

```
root@ISP1:~# ifup ens224
```

7. Подождите пока переключится работа `multiwan` через основной канал. Проверьте наличие маршрутов через резервный интерфейс.

```
Hub-n1#run ip route show proto multiwan
```

```
default via 172.16.100.1 dev eth0 src 172.16.100.2 metric 700
```

8. Пустите трафик с устройства Host-b-spoke на устройство Host-b-hub. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke.

```
root@Host-b-spoke:~# ping 192.168.100.100 -c 30
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=20 ttl=62 time=10789 ms
64 bytes from 192.168.100.100: icmp_seq=21 ttl=62 time=9767 ms
64 bytes from 192.168.100.100: icmp_seq=22 ttl=62 time=8743 ms
64 bytes from 192.168.100.100: icmp_seq=23 ttl=62 time=7719 ms
64 bytes from 192.168.100.100: icmp_seq=24 ttl=62 time=6695 ms
64 bytes from 192.168.100.100: icmp_seq=25 ttl=62 time=5671 ms
64 bytes from 192.168.100.100: icmp_seq=26 ttl=62 time=4647 ms
64 bytes from 192.168.100.100: icmp_seq=27 ttl=62 time=3623 ms

--- 192.168.100.100 ping statistics ---
30 packets transmitted, 8 received, 73.3333% packet loss, time 29702ms
rtt min/avg/max/mdev = 3623.275/7206.919/10788.680/2345.757 ms, pipe 11
```

9. Проверьте, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub-n1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 27 (172.16.100.2,500)-(172.16.1.2,500) active 1936 1968

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1  9  (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*)  *  ESP  tunn
704 704
```

```
Spoke#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 17 (172.16.1.2,500)-(172.16.100.2,500) active 1968 1936

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1  9  (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*)  *  ESP  tunn
704 704
```

Видно, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение через основной канал. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

10. Проверьте с помощью средств `multiwan` состояние каналов связи:

```
Hub-n1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: ISP1
    Interface: eth0
    Priority: 300
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
```

Видно, что первый канал связи, через eth0 с приоритетом 300 доступен и снова стал активным. Второй канал связи, через eth1 с приоритетом 200 доступен и является неактивным.

Проверка работоспособности при отказе основной ноды.

Отказ основной ноды может быть вызван как отказом интерфейса, так и отказов интерфейсов коммутаторов, связанных с основной нодой.

1. Убедитесь, что основная нода Hub-n1 находится в состоянии Master, а резервная Hub-n2 в Backup, для этого выполните следующие команды из консоли cisco-like:

```
Hub-n1#show vrrp
```

```
Interface                     VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0         1     Master  IPv4
TenGigabitEthernet0/1         2     Master  IPv4
TenGigabitEthernet0/2         3     Master  IPv4
```

```
Hub-n2#show vrrp
```

```
Interface                     VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0         1     Backup  IPv4
TenGigabitEthernet0/1         2     Backup  IPv4
TenGigabitEthernet0/2         3     Backup  IPv4
```

2. Для эмуляции отказа основной ноды отключим отслеживаемый интерфейс основной ноды или можно отключить сетевые кабели от отслеживаемых интерфейсов:

```
Hub-n1#run ifdown eth0
```

3. Убедитесь, что основная нода перешла в состояние Fault. Для этого посмотрите настройки VRRP:

```
Hub-n1#show vrrp
```

```
Interface                     VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0         1     Fault   IPv4
TenGigabitEthernet0/1         2     Fault   IPv4
TenGigabitEthernet0/2         3     Fault   IPv4
```

Если вы отключили кабели вместо отключения интерфейса, интерфейсы перейдут в состояние Backup

```
Hub-n1#show vrrp
```

```
Interface                     VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0         1     Backup  IPv4
TenGigabitEthernet0/1         2     Backup  IPv4
TenGigabitEthernet0/2         3     Backup  IPv4
```

4. Убедитесь, что резервная нода перешла в состояние Master:

```
Hub-n2#show vrrp
```

```
Interface                     VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0         1     Master  IPv4
TenGigabitEthernet0/1         2     Master  IPv4
TenGigabitEthernet0/2         3     Master  IPv4
```

5. Убедитесь, что `multiwan` маршрутов на основной Hub-n1 ноде нет, но появились на резервной Hub-n2 ноде:

```
Hub-n1#run ip route show proto multiwan
```

```
Hub-n1#
```

```
Hub-n2#run ip route show proto multiwan
```

```
default via 172.16.100.1 dev eth0 src 172.16.100.2 metric 700
```

5. Проверьте, что с криптошлюзов Hub-n1 и Hub-n2 доступен по ICMP криптошлюз Spoke. Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub-n1#ping 172.16.1.2
```

```
PING 172.16.1.2 (172.16.1.2) 100(128) bytes of data.
108 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=0.637 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=0.764 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=0.695 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=0.912 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=0.958 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4019ms
rtt min/avg/max/mdev = 0.637/0.793/0.958/0.123 ms
```

```
Hub-n2#ping 172.16.1.2
```

```
PING 172.16.1.2 (172.16.1.2) 100(128) bytes of data.
108 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=1.20 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=1.09 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=1.23 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=1.28 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=1.08 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.084/1.174/1.278/0.077 ms
```

Видно, что с криптошлюзов Hub-n1 и Hub-n2 доступен по ICMP криптошлюз Spoke.

6. Пустите трафик с устройства Host-b-spoke на устройство Host-b-hub. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke.

```
root@Host-b-spoke:~# ping 192.168.100.100 -c 5
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=1 ttl=62 time=1015 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=23.1 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=2.46 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=1.60 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=1.92 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4020ms
rtt min/avg/max/mdev = 1.600/208.759/1014.718/403.062 ms
```

7. Проверьте, что на криптошлюзах Hub-n2 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub-n2#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.100.2,500)-(172.16.1.2,500) active 2312 2428

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 2 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn
440 440
```

```
Spoke#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 4 (172.16.1.2,500)-(172.16.100.2,500) active 2428 2312

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 3 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn
440 440
```

Видно, что на криптошлюзах Hub-n2 и Spoke установлено защищенное IPsec соединение через основной канал. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

8. Проверьте с помощью средств `multiwan` состояние каналов связи:

```
Hub-n1#run mwan_mgr show --links_status
```

```
Link: 1
    Name: ISP1
    Interface: eth0
    Priority: 300
    State: unreachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: unreachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
```

```
Hub-n2#run mwan_mgr show --links_status
```

```
Link: 1
    Name: ISP1
    Interface: eth0
    Priority: 300
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.1.2"
    IPv6 hosts: None
```

Видно, что первый канал связи, через eth0 с приоритетом 300 доступен и является активным. Второй канал связи, через eth1 с приоритетом 200 тоже доступен.

9. Выполните проверку работоспособности при отказе основного канала связи для Hub-n2. Поведение для Hub-n2 должно быть аналогичным поведению Hub-n1.

10. Восстановите основную ноду включив внешний интерфейс основной ноды или подключив сетевые кабели от интерфейсов `TenGigabitEthernet0/0` и `TenGigabitEthernet0/0`:

```
Hub-n1#run ifup eth0
```

11. Убедитесь, что основная нода перешла в состояние Master. Для этого посмотрите настройки VRRP:

```
Hub-n1#show vrrp
```

```
Interface                      VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0          1     Master  IPv4
TenGigabitEthernet0/1          2     Master  IPv4
TenGigabitEthernet0/2          3     Master  IPv4
```

12. Убедитесь, что резервная нода перешла в состояние Backup:

```
Hub-n2#show vrrp
```

```
Interface                      VRID  State   AddrFamily
-----------------------------------------------------
TenGigabitEthernet0/0          1     Backup  IPv4
TenGigabitEthernet0/1          2     Backup  IPv4
TenGigabitEthernet0/2          3     Backup  IPv4
```

13. Убедитесь, что `multiwan` маршрутов на резервной Hub-n2 ноде нет, но появились на основной Hub-n1 ноде:

```
Hub-n1#run ip route show proto multiwan
```

```
default via 172.16.100.1 dev eth0 src 172.16.100.2 metric 700
```

```
Hub-n2#run ip route show proto multiwan
```

```
Hub-n2#
```

14. Проверьте, что с криптошлюзов Hub-n1 и Hub-n2 доступен по ICMP криптошлюз Spoke. Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub-n1#ping 172.16.1.2
```

```
PING 172.16.1.2 (172.16.1.2) 100(128) bytes of data.
108 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=0.637 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=0.764 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=0.695 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=0.912 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=0.958 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4019ms
rtt min/avg/max/mdev = 0.637/0.793/0.958/0.123 ms
```

```
Hub-n2#ping 172.16.1.2
```

```
PING 172.16.1.2 (172.16.1.2) 100(128) bytes of data.
108 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=1.20 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=1.09 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=1.23 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=1.28 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=1.08 ms
```

```
--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.084/1.174/1.278/0.077 ms
```

Видно, что с криптошлюзов Hub-n1 и Hub-n2 доступен по ICMP криптошлюз Spoke.

15. Пустите трафик с устройства Host-b-spoke на устройство Host-b-hub. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke.

```
root@Host-b-spoke:~# ping 192.168.100.100 -c 5
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=1 ttl=62 time=211 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=23.9 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=1.95 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=2.19 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=1.67 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.674/48.131/210.999/81.875 ms
```

16. Проверьте, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub-n1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.100.2,500)-(172.16.1.2,500) active 1936 1968

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn
440 440
```

```
Spoke#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 6 (172.16.1.2,500)-(172.16.100.2,500) active 1968 1936

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 5 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn
440 440
```

Видно, что на криптошлюзах Hub-n1 и Spoke установлено защищенное IPsec соединение через основной канал. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Приложение

Конфигурация криптошлюза Hub-n1

1. Конфигурация cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 secret 5 $6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOW
Zlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA1aTRi20ElR4wtMXTl0
aaa new-model
!
!
fhrp version vrrp v3
hostname Hub-n1
enable secret 5 PC9d7N5HlAyLrzuA3qRJvQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp policy 2
 encr gost
 hash gost341112-256-tc26
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto isakmp profile OLD_ALGS
 match identity dn
 set identity mapping
 set policy 2
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
!
```

```
object-group network SELF
 description IP on WAN interfaces
 host 172.16.100.10
 host 172.17.100.10
 host 172.16.100.2
 host 172.17.100.2
!
ip access-list extended IPSEC_ACL_HUB_AND_SPOKE
 permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
!
ip access-list extended FIREWALL_IN
 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 vrrp any host 224.0.0.18
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN_ISP1 1 ipsec-isakmp
 match address IPSEC_ACL_HUB_AND_SPOKE
 set transform-set MAGMA_GOST_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_ISP2 1 ipsec-isakmp
 match address IPSEC_ACL_HUB_AND_SPOKE
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.17.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 ip address 172.16.100.10 255.255.255.0
 ip access-group FIREWALL_IN in
 crypto map VPN_ISP1
 ip inspect INSPECT out
 vrrp 1 address-family ipv4
  timers advertise 3
  timers garp 5
  no track interface primary
  address 172.16.100.2 255.255.255.0
  exit-vrrp
!
interface TenGigabitEthernet0/1
 ip address 172.17.100.10 255.255.255.0
 ip access-group FIREWALL_IN in
 crypto map VPN_ISP2
```

```
 ip inspect INSPECT out
 vrrp 2 address-family ipv4
  timers advertise 3
  timers garp 5
  no track interface primary
  address 172.17.100.2 255.255.255.0
  exit-vrrp
!
interface TenGigabitEthernet0/2
 ip address 192.168.100.10 255.255.255.0
 vrrp 3 address-family ipv4
  timers advertise 3
  timers garp 5
  track interface TenGigabitEthernet0/0 weight -40
  track interface TenGigabitEthernet0/1 weight -40
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface TenGigabitEthernet0/3
 no ip address
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 192.168.100.20 1001
!
!
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
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
E34EE9640D905032C1F1E3B4491257D5

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Thu Feb  8 09:16:53 2024

GlobalParameters(
    Title                          = "This LSP was automatically generated by CSP
Converter at Thu Feb  8 09:16:53 2024 (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 = 1001
        )
```

```
)

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
)

IKETransform crypto:isakmp:policy:2
(
    CipherAlg   = "G2814789CPRO1-K256-CBC-65534"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

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

FilterChain FilterChain:FIREWALL_IN (
    Filters = Filter (
        DestinationIP = 172.16.100.10
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.10
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
```

```
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.10
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.10
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.10
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.10
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
```

```
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.10
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.10
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.10
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.10
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
```

```
    ))

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

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

IKERule IKERule:VPN_ISP1:1
(
```

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

IKERule IKERule:VPN_ISP2:1
(
    IKEPeerIPFilter = 172.16.1.2
    IKELocalIPFilter = 172.17.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 20
)

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

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

FilterChain IPsecPolicy:VPN_ISP1 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
```

```
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP1:1 >
        LogEventID = "IPsec:Protect:VPN_ISP1:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

FilterChain IPsecPolicy:VPN_ISP2 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP2:1 >
        LogEventID = "IPsec:Protect:VPN_ISP2:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/0"
    InputFilter = FilterChain:FIREWALL_IN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN_ISP1
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/1"
    InputFilter = FilterChain:FIREWALL_IN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN_ISP2
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

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

```
config_format_version: 1
links_params:
  links:
  - name: ISP1
    check:
        hosts_reachability: any
        ipv4:
            hosts:
            - 172.16.101.15
            - 172.16.1.2
```

```
            source_address: 172.16.100.2
        method:
            icmp:
                max_rtt: 100
                interval: 3
                restore_interval: 1
                restore_count: 10
                failure_count: 10
                tos: 184
                watch_rules: true
    iface_name: eth0
    prio: 300
    routes:
        ipv4:
            networks:
            - 0.0.0.0/0
            nexthop: 172.16.100.1
            source_address: 172.16.100.2
        watch_foreign_routes: false
        watch_self_routes: true
  - name: ISP2
    check:
        hosts_reachability: any
        ipv4:
            hosts:
            - 172.16.101.15
            - 172.16.1.2
            source_address: 172.17.100.2
        method:
            icmp:
                max_rtt: 100
                interval: 3
                restore_interval: 1
                restore_count: 10
                failure_count: 10
                tos: 184
                watch_rules: true
    iface_name: eth1
    prio: 200
    routes:
        ipv4:
            networks:
            - 0.0.0.0/0
            nexthop: 172.17.100.1
            source_address: 172.17.100.2
        watch_foreign_routes: false
        watch_self_routes: true
```

Конфигурация криптошлюза Hub-n2

1. Конфигурация cisco-like:

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

```
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 secret 5 $6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOW
Zlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA1aTRi20ElR4wtMXTl0
aaa new-model
!
!
fhrp version vrrp v3
hostname Hub-n2
enable secret 5 PC9d7N5HlAyLrzuA3qRJvQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp policy 2
 encr gost
 hash gost341112-256-tc26
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto isakmp profile OLD_ALGS
 match identity dn
 set identity mapping
 set policy 2
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
!
object-group network SELF
 description IP on WAN interfaces
 host 172.16.100.20
 host 172.17.100.20
 host 172.16.100.2
 host 172.17.100.2
!
ip access-list extended IPSEC_ACL_HUB_AND_SPOKE
 permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
!
ip access-list extended FIREWALL_IN
 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 vrrp any host 224.0.0.18
```

```
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN_ISP1 1 ipsec-isakmp
 match address IPSEC_ACL_HUB_AND_SPOKE
 set transform-set MAGMA_GOST_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_ISP2 1 ipsec-isakmp
 match address IPSEC_ACL_HUB_AND_SPOKE
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.17.100.2
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 ip address 172.16.100.20 255.255.255.0
 ip access-group FIREWALL_IN in
 crypto map VPN_ISP1
 ip inspect INSPECT out
 vrrp 1 address-family ipv4
  timers advertise 3
  timers garp 5
  priority 50
  no track interface primary
  address 172.16.100.2 255.255.255.0
  exit-vrrp
!
interface TenGigabitEthernet0/1
 ip address 172.17.100.20 255.255.255.0
 ip access-group FIREWALL_IN in
 crypto map VPN_ISP2
 ip inspect INSPECT out
 vrrp 2 address-family ipv4
  timers advertise 3
  timers garp 5
  priority 50
  no track interface primary
  address 172.17.100.2 255.255.255.0
  exit-vrrp
!
interface TenGigabitEthernet0/2
 ip address 192.168.100.20 255.255.255.0
 vrrp 3 address-family ipv4
  timers advertise 3
  timers garp 5
  priority 50
```

```
  address 192.168.100.1 255.255.255.0
  exit-vrrp
!
interface TenGigabitEthernet0/3
 no ip address
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 192.168.100.10 1001
!
!
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
3082024C308201FBA003020102102B6E3351FD6EB2AD48200203CB5BA14130
...
E34EE9640D905032C1F1E3B4491257D5

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Thu Feb  8 09:21:58 2024

GlobalParameters(
    Title                         = "This LSP was automatically generated by CSP
Converter at Thu Feb  8 09:21:58 2024 (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 = 1001
        )

)

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
)

IKETransform crypto:isakmp:policy:2
(
    CipherAlg   = "G2814789CPRO1-K256-CBC-65534"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

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

FilterChain FilterChain:FIREWALL_IN (
    Filters = Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
```

```
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 50
```

```
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

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

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

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

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

IKERule IKERule:VPN_ISP2:1
(
    IKEPeerIPFilter = 172.16.1.2
    IKELocalIPFilter = 172.17.100.2
```

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

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

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

FilterChain IPsecPolicy:VPN_ISP1 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP1:1 >
        LogEventID = "IPsec:Protect:VPN_ISP1:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

FilterChain IPsecPolicy:VPN_ISP2 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
```

```
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP2:1 >
        LogEventID = "IPsec:Protect:VPN_ISP2:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/0"
    InputFilter = FilterChain:FIREWALL_IN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN_ISP1
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/1"
    InputFilter = FilterChain:FIREWALL_IN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN_ISP2
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

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

```
config_format_version: 1
links_params:
  links:
  - name: ISP1
    check:
        hosts_reachability: any
        ipv4:
            hosts:
            - 172.16.101.15
            - 172.16.1.2
            source_address: 172.16.100.2
        method:
            icmp:
                max_rtt: 100
                interval: 3
                restore_interval: 1
                restore_count: 10
                failure_count: 10
                tos: 184
                watch_rules: true
    iface_name: eth0
    prio: 300
    routes:
        ipv4:
            networks:
```

```
            - 0.0.0.0/0
            nexthop: 172.16.100.1
            source_address: 172.16.100.2
        watch_foreign_routes: false
        watch_self_routes: true
  - name: ISP2
    check:
        hosts_reachability: any
        ipv4:
            hosts:
            - 172.16.101.15
            - 172.16.1.2
            source_address: 172.17.100.2
        method:
            icmp:
                max_rtt: 100
                interval: 3
                restore_interval: 1
                restore_count: 10
                failure_count: 10
                tos: 184
                watch_rules: true
    iface_name: eth1
    prio: 200
    routes:
        ipv4:
            networks:
            - 0.0.0.0/0
            nexthop: 172.17.100.1
            source_address: 172.17.100.2
        watch_foreign_routes: false
        watch_self_routes: true
```

Конфигурация криптошлюза Spoke

1. Конфигурация cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 secret 5 $6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOW
Zlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA1aTRi20ElR4wtMXTl0
aaa new-model
!
!
hostname Spoke
enable secret 5 PC9d7N5HlAyLrzuA3qRJvQ==
!
!
!
!
```

```
!
!
!
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_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 description IP on WAN interfaces
 host 172.16.1.2
!
ip access-list extended IPSEC_ACL_HUB_AND_SPOKE
 permit ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
!
ip access-list extended FIREWALL_IN
 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_HUB_AND_SPOKE
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 set peer 172.17.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 ip address 172.16.1.2 255.255.255.0
 ip access-group FIREWALL_IN in
 crypto map VPN
 ip inspect INSPECT out
!
interface TenGigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
!
interface TenGigabitEthernet0/2
 no ip address
 shutdown
!
!
```

```
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
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Thu Feb  8 09:21:58 2024

GlobalParameters(
    Title                          = "This LSP was automatically generated by CSP
Converter at Thu Feb  8 09:21:58 2024 (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 = 1001
        )

)

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
)

IKETransform crypto:isakmp:policy:2
(
    CipherAlg   = "G2814789CPRO1-K256-CBC-65534"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

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

FilterChain FilterChain:FIREWALL_IN (
    Filters = Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
```

```
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
```

```
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.20
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.20
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 172.17.100.2
        ProtocolID = 112
        Action = PASS
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

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

```
    ))

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

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

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

IKERule IKERule:VPN_ISP2:1
(
    IKEPeerIPFilter = 172.16.1.2
    IKELocalIPFilter = 172.17.100.2
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 20
)
```

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

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

FilterChain IPsecPolicy:VPN_ISP1 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP1:1 >
        LogEventID = "IPsec:Protect:VPN_ISP1:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

FilterChain IPsecPolicy:VPN_ISP2 (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN_ISP2:1 >
        LogEventID = "IPsec:Protect:VPN_ISP2:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

NetworkInterface (
```

```
    LogicalName = "TenGigabitEthernet0/0"
    InputFilter = FilterChain:FIREWALL_IN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN_ISP1
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/1"
    InputFilter = FilterChain:FIREWALL_IN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN_ISP2
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/2"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/3"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```