Резервирование каналов связи при помощи сервиса MultiWAN (IPv4 через IPv6)

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» 5.0 для обеспечения безопасного взаимодействия между защищаемыми подсетями центрального офиса и филиала. Защищаемые подсети в LAN сегменте – IPv4 и недоверенные подсети в WAN сегменте – IPv6. 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».

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 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 произведет переключение 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. При работе `multiwan`, если в системе присутствуют маршруты, адреса назначения которых совпадают с адресом назначения multiwan маршрута, то такие маршруты 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 используется ICMP запросы. Для ICMP запросов сервиса `multiwan` используются отдельные таблицы маршрутизации. Данные таблицы будут содержать маршруты по умолчанию через адрес следующего перехода и очищаются 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 (протокол IPv4)», со сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (IPv4 через IPv6) » и со сценарием «Резервирование каналов связи при помощи сервиса MultiWAN». Далее по тексту сценарий «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)» будет называться базовым сценарием.

Требования к инфраструктуре

1. Требования к устройствам.

1.1. «С-Терра Шлюз».

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

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

1.2. Центр выпуска сертификатов.

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

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

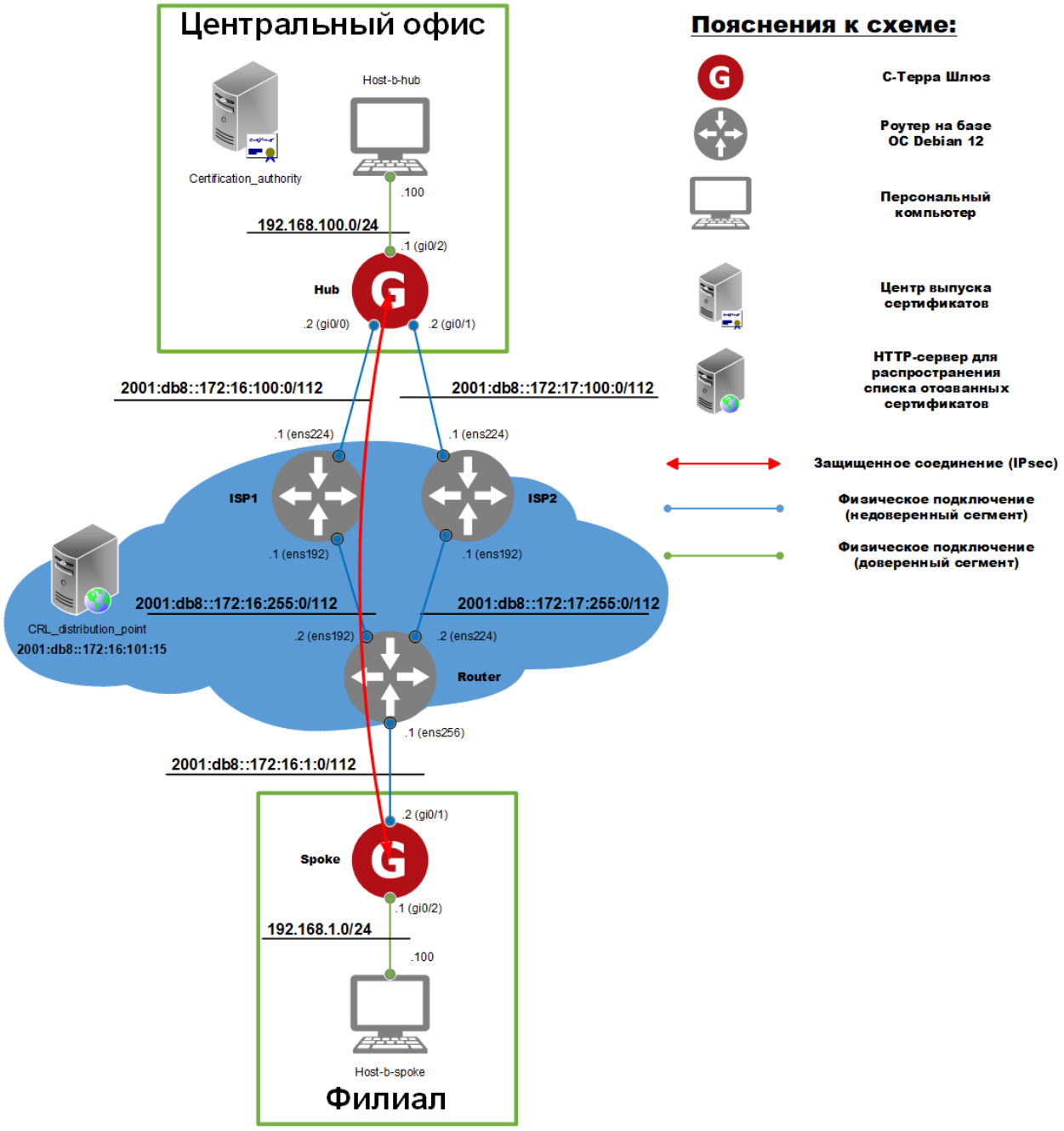


Рисунок 1. Общая схема

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), криптошлюз «С-Терра Шлюз» (Hub) 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. 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).

   2.1. Криптошлюз Hub подключается к сети Интернет с помощью сервиса `multiwan`. Основной маршрут по умолчанию через маршрутизатор ISP1. Резервный маршрут по умолчанию через маршрутизатор ISP2.

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

3. Отказоустойчивость.

   За резервирования каналов связи отвечает сервис `multiwan`. Сервис `multiwan` добавляет IPv6 маршрут по умолчанию через ISP1 или ISP2. Также добавляет IPv4 маршрут по умолчанию через WAN интерфейс в сторону ISP1 или ISP2 для маршрутизации пакетов из IPv4-сети в сеть IPv6. 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 создается своё правило для ICMP запроса. Для каждого канала создаётся своя таблица с маршрутами до проверяемых хостов через ISP1 или через ISP2 с указанием src адреса. Данная таблица будет содержать только маршруты до проверяемых хостов, все остальные маршруты из таблиц для каналов будут удалены.

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:

   - 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:

   - 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 IPv6 трафик:

5.3.1 Пакеты сообщений Neighbor Solicitation (NS) и Neighbor Advertisement (NA) протокола Neighbor Discovery (ND). Разрешение приема сообщений ND-NS необходимо для обнаружения криптошлюза смежным сетевым оборудованием (Router в настоящем сценарии) в случае, если инициатором обмена данными является не сам криптошлюз, а партнер. Прием сообщений ND-NA разрешается для случая, когда сообщение ND-NS посылается криптошлюзом, который инициирует обмен данными с партнером, а смежное сетевое оборудование посылает ND-NA с link-local адреса IPv6.

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

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

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

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 весь IPv4 трафик заблокирован – входящий и исходящий.

5.5. На WAN интерфейсах инспектируются все исходящие локальные и транзитные IPv6 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 – 2001:db8::172:16:255:1/112 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 2001:db8::172:16:100:1/112 на сетевом интерфейсе ens224.
3. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства ISP2

1. Настройте IP адрес – 2001:db8::172:17:255:1/112 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 2001:db8::172:17:100:1/112 на сетевом интерфейсе ens224.
3. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Router

1. Настройте IP адрес – 2001:db8::172:16:255:2/112 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 2001:db8::172:17:255:2/112 на сетевом интерфейсе ens224.
3. Настройте IP адрес – 2001:db8::172:16:1:1/112 на сетевом интерфейсе ens256.
4. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Hub

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. Проверьте включена ли маршрутизация IPv6

```
sterragate(config)# do run sysctl net.ipv6.conf.all.forwarding
net.ipv6.conf.all.forwarding = 1
```

Маршрутизация должна быть включена при инициализации криптошлюза. При необходимости маршрутизация IPv6 может быть включена без перезагрузки криптошлюза следующей командой:

```
sterragate(config)# do run sysctl -w net.ipv6.conf.all.forwarding=1
net.ipv6.conf.all.forwarding = 1
```

Надо учитывать, что результат выполнения команды не сохраняется после перезагрузки криптошлюза. Для включения маршрутизации IPv6 автоматически при загрузке криптошлюза необходимо записать настройку `net.ipv6.conf.all.forwarding=1` в файл `/etc/sysctl.d/02-sterra_internet_settings.conf`

2. Задайте имя устройства:

```
sterragate(config)#hostname Hub
```

3. Включите WAN и LAN интерфейсы и настройте IP адреса на WAN интерфейсах в соответствии со схемой стенда:

```
Hub(config)#interface TenGigabitEthernet0/0
Hub(config-if)# no ip address
Hub(config-if)# ipv6 address 2001:db8::172:16:100:2/112
Hub(config-if)#exit
Hub(config)#interface TenGigabitEthernet0/1
Hub(config-if)# ipv6 address 2001:db8::172:17:100:2/112
Hub(config-if)# no shutdown
Hub(config-if)#exit
Hub(config)#interface TenGigabitEthernet0/2
Hub(config-if)# ip address 192.168.100.1 255.255.255.0
Hub(config-if)# no shutdown
Hub(config-if)#exit
```

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка шифрования

1. Параметры IKE.

1.1. Задайте общие параметры IKE:

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

1.2. Создайте политику **1**, описывающую алгоритмы IKE туннеля:

```
Hub(config)#crypto isakmp policy 1
Hub(config-isakmp)# encryption magma
Hub(config-isakmp)# hash stribog-256
```

```
Hub(config-isakmp)# authentication gost-sig
Hub(config-isakmp)# group vko2
Hub(config-isakmp)# exit
```

1.3. Создайте профиль настроек ISAKMP (алгоритм IKE) под именем `NEW_ALGS`, чтобы далее его прикрепить к соответствующему разделу крипто-карты:

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

2. Параметры IPsec.

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

```
Hub(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:

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

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

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

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

```
Hub(config)#crypto map VPN_ISP1 1 ipsec-isakmp
Hub(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Hub(config-crypto-map)#set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
Hub(config-crypto-map)#set local-address ipv6 2001:db8::172:16:100:2
```

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

```
Hub(config-crypto-map)# set peer ipv6 2001:db8::172:16:1:2
Hub(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub(config-crypto-map)#set dead-connection history off
Hub(config-crypto-map)#set pre-fragmentation
Hub(config-crypto-map)#set security-association mss-fix out
Hub(config-crypto-map)#exit
```

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

```
Hub(config)#crypto map VPN_ISP2 1 ipsec-isakmp
Hub(config-crypto-map)#match address IPSEC_ACL_HUB_AND_SPOKE
Hub(config-crypto-map)#set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
Hub(config-crypto-map)#set local-address ipv6 2001:db8::172:17:100:2
```

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

```
Hub(config-crypto-map)#set peer ipv6 2001:db8::172:16:1:2
```

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

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

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

4. Примените введенные настройки:

```
Hub(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 задать правила фильтрации IPv6 трафика 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(config)#object-group v6-network SELF
Hub(config-network-group)#remark WAN IP
Hub(config-network-group)#host 2001:db8::172:16:100:2
Hub(config-network-group)#host 2001:db8::172:17:100:2
Hub(config-network-group)#exit
```

2. Создайте ACL с именем `FW_INPUT_WAN_IPV6` для входящего на WAN интерфейс трафика:

```
Hub(config)#ipv6 access-list FW_INPUT_WAN_IPV6
Hub(config-ipv6-acl)#permit icmp any any nd-ns
Hub(config-ipv6-acl)#permit icmp any any nd-na
Hub(config-ipv6-acl)#permit udp any object-group SELF-IPV6 eq 500
Hub(config-ipv6-acl)#permit udp any object-group SELF-IPV6 eq 4500
Hub(config-ipv6-acl)#permit icmp any object-group SELF-IPV6
```

```
Hub(config-ipv6-acl)#permit esp any object-group SELF-IPV6
Hub(config-ipv6-acl)# exit
```

3. Создайте ACL'и с именами `FIREWALL_OUT` и `FIREWALL_IN` для запрета исходящего и входящего IPv4 трафика на WAN интерфейсе:

```
Hub(config)#ip access-list extended FIREWALL_OUT
Hub(config-ext-nacl)#deny ip any any
Hub(config-ext-nacl)# exit
Hub(config)#ip access-list extended FIREWALL_IN
Hub(config-ext-nacl)#deny ip any any
Hub(config-ext-nacl)# exit
```

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

```
Hub(config)#ipv6 inspect name INSPECT tcp
Hub(config)#ipv6 inspect name INSPECT udp
Hub(config)#ip inspect tcp sessions-max 256000
Hub(config)#ip inspect udp sessions-max 65536
```

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

```
Hub(config)# interface TenGigabitEthernet 0/0
Hub(config-if)#ip access-group FIREWALL_IN in
Hub(config-if)#ip access-group FIREWALL_OUT out
Hub(config-if)#ipv6 access-group FW_INPUT_WAN_IPV6 in
Hub(config-if)#ipv6 inspect INSPECT out
Hub(config-if)#exit
Hub(config)#interface TenGigabitEthernet 0/1
Hub(config-if)#ip access-group FIREWALL_IN in
Hub(config-if)#ip access-group FIREWALL_OUT out
Hub(config-if)#ipv6 access-group FW_INPUT_WAN_IPV6 in
Hub(config-if)#ipv6 inspect INSPECT out
Hub(config-if)#exit
```

6. Примените настройки:

```
Hub(config)#end
```

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка сервиса multiwan

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

1) Конфигурационный файл должен быть формата YAML.
2) Версия формата конфигурационного файла должна быть 1 (параметр `config_format_version`).
3) Приоритеты каналов связи, принадлежащих одной группе, должны отличаться.
4) Каждый канал связи должен содержать название интерфейса, приоритет канала связи, маршруты, адрес следующего перехода и хосты для проверки.
5) Адрес следующего перехода может отсутствовать в секции `ipv4` только если он уже задан в секции `ipv6`. Аналогично адрес следующего перехода может отсутствовать в секции `ipv6` только если он уже задан для секции `ipv4`.

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#exit
administrator@Hub] system
```

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

Для того, чтобы с помощью утилиты `mwan_mgr` корректного отображалось состояние каналов связи, обновите пакет multiwan.

Настройка `multiwan` осуществляется через файл `/etc/multiwan/config.yaml`.

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

При расположении параметра не в своей секции, значение параметра будет проигнорировано.

```
config_format_version: 1
links_params:
  preempt: true
  check_links_reachability_on_start: true
  links:
  - name: ISP1
    iface_name: eth0
    prio: 300
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
      ipv6:
        networks:
        - ::/0
        nexthop: 2001:db8::172:16:100:1
    check:
      hosts_reachability: any
      ipv6:
        hosts:
        - 2001:db8::172:16:101:15
        - 2001:db8::172:16:1:2
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
  - name: ISP2
    iface_name: eth1
    prio: 200
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
      ipv6:
        networks:
        - ::/0
        nexthop: 2001:db8::172:17:100:1
    check:
      hosts_reachability: any
      ipv6:
        hosts:
        - 2001:db8::172:16:101:15
        - 2001:db8::172:16:1:2
```

```
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
```

Параметр `iface_name` должен содержать название интерфейса в нотации linux. Соответствие названий интерфейсов в cisco-like консоли и в нотации linux можно посмотреть в карте интерфейсов:

```
root@Hub:~# 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="*")
```

В секции `routes` создайте секцию `ipv4`, содержащую параметры для IPv4 маршрутов. В нее добавьте параметр `networks` со значением маршрута по умолчанию (0.0.0.0/0). При указании маршрута без указания адреса следующего маршрутизатора, будет добавлен маршрут через интерфейс, указанный в `iface_name`.

Создайте секцию `ipv6`. В ней укажите подсеть, до которой будет добавлен маршрут, и IPv6 адрес следующего маршрутизатора. Формат IP адресов подсети должен соответствовать формату CIDR (Например, 2001:db8::172:16:100:1). Краткая запись для маршрута по умолчанию - ::/0.

В секции `check` задайте параметры проверки доступности каналов связи IPv6.

2. Проверьте правильность составленного конфигурационного файла с помощью команды `/opt/multiwan/multiwan_daemon --check_config`. Для отображения параметров, получаемых из конфигурационного файла, добавьте ключ –-`result`:

```
root@Hub:~# /opt/multiwan/multiwan_daemon --check_config --result
```

```
INFO: Parsed parameters:
config_format_version: 1
daemon_params:
  watch_intervals:
    watch_foreign_routes: 10
    watch_rules: 10
    watch_self_routes: 10
links_params:
...
```

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

Завершение настройки Hub

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
```

```
2024-04-23T17:34:37.220813+03:00 Hub multiwan: INFO: Starting multiwan...
2024-04-23T17:34:48.464978+03:00 Hub multiwan: INFO: NAT translations table has
been cleared.
```

```
2024-04-23T17:34:50.529713+03:00 Hub multiwan: INFO: LSP was reloaded.
2024-04-23T17:34:50.539143+03:00 Hub multiwan: INFO: Used interface "eth0".
Nexthop "2001:db8::172:16:100:1".
```

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

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

```
default dev eth0 metric 700
```

```
root@Hub:~# ip -6 route show proto multiwan
```

```
default via 2001:db8::172:16:100:1 dev eth0 metric 700 pref medium
```

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

```
root@Hub:~# ip route get 2001:db8::172:16:101:15
```

```
2001:db8::172:16:101:15 from :: via 2001:db8::172:16:100:1 dev eth0 proto multiwan
src 2001:db8::172:16:100:2 metric 700 pref medium
```

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:~# cert_mgr check
```

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

В Приложении представлены тексты конфигураций для криптошлюза Hub:

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

Настройка криптошлюза Spoke

Настройка криптошлюза Spoke производится по аналогии с Hub без настроек `multiwan`. В статической крипто-карте 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 ipv6 2001:db8::172:16:100:2
Spoke(config-crypto-map)#set peer ipv6 2001:db8::172:17:100:2
Spoke(config-crypto-map)#set pre-fragmentation
Spoke(config-crypto-map)#set dead-connection history off
```

Для успешной работы протокола IPv6 в конфигурацию Spoke нужно добавить два маршрута по умолчанию:

```
Spoke(config)#ip route 0.0.0.0 0.0.0.0 TenGigabitEthernet0/0
Spoke(config)#ipv6 route ::/0 2001:db8::172:16:1:1
```

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

```
Hub#ping 2001:db8::172:16:100:1
```

```
PING 2001:db8::172:16:100:1(2001:db8::172:16:100:1) 100 data bytes
108 bytes from 2001:db8::172:16:100:1: icmp_seq=1 ttl=64 time=1.32 ms
108 bytes from 2001:db8::172:16:100:1: icmp_seq=2 ttl=64 time=0.558 ms
108 bytes from 2001:db8::172:16:100:1: icmp_seq=3 ttl=64 time=0.302 ms
108 bytes from 2001:db8::172:16:100:1: icmp_seq=4 ttl=64 time=0.979 ms
108 bytes from 2001:db8::172:16:100:1: icmp_seq=5 ttl=64 time=0.990 ms

--- 2001:db8::172:16:100:1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4026ms
rtt min/avg/max/mdev = 0.302/0.830/1.322/0.358 ms
```

```
Hub#ping 2001:db8::172:17:100:1
```

```
PING 2001:db8::172:17:100:1(2001:db8::172:17:100:1) 100 data bytes
108 bytes from 2001:db8::172:17:100:1: icmp_seq=1 ttl=64 time=0.352 ms
108 bytes from 2001:db8::172:17:100:1: icmp_seq=2 ttl=64 time=0.486 ms
108 bytes from 2001:db8::172:17:100:1: icmp_seq=3 ttl=64 time=0.495 ms
108 bytes from 2001:db8::172:17:100:1: icmp_seq=4 ttl=64 time=0.560 ms
108 bytes from 2001:db8::172:17:100:1: icmp_seq=5 ttl=64 time=0.516 ms

--- 2001:db8::172:17:100:1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.352/0.481/0.560/0.069 ms
```

```
Spoke#ping 2001:db8::172:16:1:1
```

```
PING 2001:db8::172:16:1:1(2001:db8::172:16:1:1) 100 data bytes
108 bytes from 2001:db8::172:16:1:1: icmp_seq=1 ttl=64 time=0.298 ms
108 bytes from 2001:db8::172:16:1:1: icmp_seq=2 ttl=64 time=0.492 ms
108 bytes from 2001:db8::172:16:1:1: icmp_seq=3 ttl=64 time=0.466 ms
108 bytes from 2001:db8::172:16:1:1: icmp_seq=4 ttl=64 time=0.536 ms
108 bytes from 2001:db8::172:16:1:1: icmp_seq=5 ttl=64 time=0.422 ms

--- 2001:db8::172:16:1:1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4034ms
rtt min/avg/max/mdev = 0.298/0.442/0.536/0.081 ms
```

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

2. Проверьте, что с защищаемых устройств Host-b-hub и Host-b-spoke доступен по ICMP соответствующий криптошлюз, который является 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=64 time=0.211 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.235 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.223 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=0.254 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=64 time=0.233 ms
```

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

```
root@Host-b-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=0.214 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.189 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.210 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.212 ms

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

Видно, что с защищаемых устройств Host-b-hub и Host-b-spoke доступны по ICMP соответствующие криптошлюзы.

3. Проверьте, что с криптошлюза Hub доступен по ICMP криптошлюз Spoke (по WAN интерфейсам). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub#ping 2001:db8::172:16:1:2
```

```
PING 2001:db8::172:16:1:2(2001:db8::172:16:1:2) 100 data bytes
108 bytes from 2001:db8::172:16:1:2: icmp_seq=1 ttl=62 time=0.924 ms
108 bytes from 2001:db8::172:16:1:2: icmp_seq=2 ttl=62 time=1.33 ms
108 bytes from 2001:db8::172:16:1:2: icmp_seq=3 ttl=62 time=1.27 ms
108 bytes from 2001:db8::172:16:1:2: icmp_seq=4 ttl=62 time=1.24 ms
108 bytes from 2001:db8::172:16:1:2: icmp_seq=5 ttl=62 time=1.05 ms

--- 2001:db8::172:16:1:2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.924/1.163/1.329/0.152 ms
```

Видно, что с криптошлюза Hub доступен по ICMP криптошлюз Spoke.

4. Проверьте наличие маршрутов через интерфейс, направленный в сторону основного канала связи.

```
Hub#run ip r show proto multiwan
```

```
default dev eth0 metric 700
```

```
Hub#run ip -6 route show proto multiwan
```

```
default via 2001:db8::172:16:100:1 dev eth0 metric 700 pref medium
```

Проверка PKI

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

```
Hub#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
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
```

```
FoundFound 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 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 и Spoke статус всех сертификатов Active. Для этого выполните команду `run cert_mgr check` из cisco-like консоли криптошлюзов.

```
Hub#run cert_mgr check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub
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 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 и Spoke

1. Инициируйте построение защищенного соединения между криптошлюзами Hub 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=586 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=1.91 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=1.93 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=1.90 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 1.648/118.753/586.379/233.812 ms
```

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

```
Hub#run sa_mgr show
```

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

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (2001:db8::172:16:100:2,500)-(2001:db8::172:16:1:2,500) active 1928 1960

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
2112 2112
```

```
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 (2001:db8::172:16:1:2,500)-(2001:db8::172:16:100:2,500) active 1960 1928

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
5896 5896
```

Видно, что на криптошлюзах Hub 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#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: None
    IPv4 nexthop: None
    IPv6 hosts: "2001:db8::172:16:101:15", "2001:db8::172:16:1:2"
    IPv6 nexthop: "2001:db8::172:16:100:1"
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: None
    IPv4 nexthop: None
    IPv6 hosts: "2001:db8::172:16:101:15", "2001:db8::172:16:1:2"
    IPv6 nexthop: "2001:db8::172:17:100:1"
```

Видно, что первый канал связи, через eth0 с приоритетом 300 доступен. Является активным. Доступность канала проверяется доступностью любого хоста из заданных IPv4 хостов. Второй канал связи, через eth1 с приоритетом 200 доступен. Является неактивным. Доступность канала проверяется доступностью любого хоста из заданных IPv4 хостов.

Проверка работоспособности при отказе основного канала связи.

1. Для эмуляции отказа основного канала связи отключим основной интерфейс.

   ```
   Hub#run ifdown eth0
   ```

2. Подождите пока переключится работа `multiwan` через резервный канал. Проверьте наличие маршрутов через интерфейс, направленный в сторону резервного канала связи.

```
Hub#run ip r show proto multiwan
```

```
default dev eth1 metric 800
```

```
Hub#run ip -6 route show proto multiwan
```

```
default via 2001:db8::172:17:100:1 dev eth1 metric 800 pref medium
```

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 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub#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 11 (2001:db8::172:17:100:2,500)-(2001:db8::172:16:1:2,500) active 1928 1960

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 3 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn
1056 1056
```

```
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 7 (2001:db8::172:16:1:2,500)-(2001:db8::172:17:100:2,500) active 1960 1928

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
1056 1056
```

Видно, что на криптошлюзах Hub 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#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: None
    IPv4 nexthop: None
    IPv6 hosts: "2001:db8::172:16:101:15", "2001:db8::172:16:1:2"
    IPv6 nexthop: "2001:db8::172:16:100:1"
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: None
    IPv4 nexthop: None
    IPv6 hosts: "2001:db8::172:16:101:15", "2001:db8::172:16:1:2"
    IPv6 nexthop: "2001:db8::172:17:100:1"
```

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 основной интерфейс.

```
Hub#run ifup eth0
```

7. Подождите пока переключится работа `multiwan` через основной канал. Проверьте наличие маршрутов через интерфейс, направленный в сторону основного канала связи.

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

```
default dev eth0 metric 700
```

```
Hub#run ip -6 route show proto multiwan
```

```
default via 2001:db8::172:16:100:1 dev eth0 metric 700 pref medium
```

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 и Spoke установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub#run sa_mgr show
```

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

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 19 (2001:db8::172:16:100:2,500)-(2001:db8::172:16:1:2,500) active 1904 1936

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 5 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn
55616 55440
```

```
Spoke#run sa_mgr show
```

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

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 9 (2001:db8::172:16:1:2,500)-(2001:db8::172:16:100:2,500) active 1936 1904

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
65120 65296
```

Видно, что на криптошлюзах Hub 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#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: None
    IPv4 nexthop: None
    IPv6 hosts: "2001:db8::172:16:101:15", "2001:db8::172:16:1:2"
    IPv6 nexthop: "2001:db8::172:16:100:1"
Link: 2
    Name: ISP2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: None
    IPv4 nexthop: None
    IPv6 hosts: "2001:db8::172:16:101:15", "2001:db8::172:16:1:2"
    IPv6 nexthop: "2001:db8::172:17:100:1"
```

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.

Приложение

Конфигурация криптошлюза Hub

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username          cscons          privilege          15          secret          5
$6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOWZlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA
1aTRi20ElR4wtMXTl0
aaa new-model
!
!
hostname Hub
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 v6-network SELF
 description IP on WAN interface
 host 2001:db8::172:16:100:2
 host 2001:db8::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_OUT
 deny   ip any any
!
ip access-list extended FIREWALL_IN
 deny   ip any any
!
ipv6 default filtering deny
!
ipv6 access-list FW_INPUT_WAN_IPV6
 permit icmp any any nd-ns
```

```
 permit icmp any any nd-na
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ipv6 inspect name INSPECT tcp
ipv6 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 ipv6 2001:db8::172:16:100:2
 set peer ipv6 2001:db8::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 ipv6 2001:db8::172:17:100:2
 set peer ipv6 2001:db8::172:16:1:2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
 no ip address
 ipv6 address 2001:db8::172:16:100:2/112
 ip access-group FIREWALL_IN in
 ip access-group FIREWALL_OUT out
 ipv6 access-group FW_INPUT_WAN_IPV6 in
 crypto map VPN_ISP1
 ipv6 inspect INSPECT out
!
interface TenGigabitEthernet0/1
 no ip address
 ipv6 address 2001:db8::172:17:100:2/112
 ip access-group FIREWALL_IN in
 ip access-group FIREWALL_OUT out
 ipv6 access-group FW_INPUT_WAN_IPV6 in
 crypto map VPN_ISP2
 ipv6 inspect INSPECT out
!
interface TenGigabitEthernet0/2
 ip address 192.168.100.1 255.255.255.0
!
interface TenGigabitEthernet0/3
 no ip address
 shutdown
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://[2001:db8::172:16:101:15]/certcrl.crl
```

```
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 2B6E3351FD6EB2AD48200203CB5BA141
3082024C308201FBA00302010202102B6E3351FD6EB2AD48200203CB5BA14130
...
E34EE9640D905032C1F1E3B4491257D5

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Thu May 16 10:25:53 2024

GlobalParameters(
    Title                          = "This LSP was automatically generated by CSP
Converter at Thu May 16 10:25:53 2024 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

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

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

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

FilterChain FilterChain:FIREWALL_IN:FW_INPUT_WAN_IPV6 (
    Filters = Filter (
```

```
        DestinationIP = 0.0.0.0/0
        Action = DROP
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        ProtocolID = 58
        ICMPType = 135
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = ::/0
        ProtocolID = 58
        ICMPType = 136
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:100:2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:17:100:2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:100:2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:17:100:2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:100:2
        ProtocolID = 58
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:17:100:2
        ProtocolID = 58
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
```

```
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:100:2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:17:100:2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

IKERule IKERule:VPN_ISP1:1
(
    IKEPeerIPFilter = 2001:db8::172:16:1:2
    IKELocalIPFilter = 2001:db8::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 = 2001:db8::172:16:1:2
    IKELocalIPFilter = 2001:db8::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 = IPv6TunnelEntry(
        LocalIPAddress = 2001:db8::172:16:100:2
        PeerAddress = 2001:db8::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 = IPv6TunnelEntry(
        LocalIPAddress = 2001:db8::172:17:100:2
        PeerAddress = 2001:db8::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:FW_INPUT_WAN_IPV6
    OutputFilter = FilterChain:FIREWALL_OUT
    IPsecPolicy = IPsecPolicy:VPN_ISP1
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/1"
    InputFilter = FilterChain:FIREWALL_IN:FW_INPUT_WAN_IPV6
    OutputFilter = FilterChain:FIREWALL_OUT
    IPsecPolicy = IPsecPolicy:VPN_ISP2
)
```

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

```
config_format_version: 1
links_params:
  preempt: true
  check_links_reachability_on_start: true
  links:
  - name: ISP1
    iface_name: eth0
    prio: 300
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
```

```
      ipv6:
        networks:
        - ::/0
        nexthop: 2001:db8::172:16:100:1
    check:
      hosts_reachability: any
      ipv6:
        hosts:
        - 2001:db8::172:16:101:15
        - 2001:db8::172:16:1:2
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
  - name: ISP2
    iface_name: eth1
    prio: 200
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
      ipv6:
        networks:
        - ::/0
        nexthop: 2001:db8::172:17:100:1
    check:
      hosts_reachability: any
      ipv6:
        hosts:
        - 2001:db8::172:16:101:15
        - 2001:db8::172:16:1:2
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
```

Конфигурация криптошлюза Spoke

1. Консоль cisco-like

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

```
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username          cscons          privilege          15          secret          5
$6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOWZlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA
1aTRi20ElR4wtMXTl0
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
!
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_OUT
 deny   ip any any
!
ip access-list extended FIREWALL_IN
 deny   ip any any
!
ipv6 default filtering deny
!
ipv6 access-list FW_INPUT_WAN_IPV6
 permit icmp any any nd-ns
 permit icmp any any nd-na
 permit udp any host 2001:db8::172:16:1:2 eq isakmp
 permit udp any host 2001:db8::172:16:1:2 eq non500-isakmp
 permit icmp any host 2001:db8::172:16:1:2
 permit esp any host 2001:db8::172:16:1:2
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ipv6 inspect name INSPECT tcp
ipv6 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 ipv6 2001:db8::172:16:100:2
 set peer ipv6 2001:db8::172:17:100:2
 set pre-fragmentation
 set dead-connection history off
!
interface TenGigabitEthernet0/0
```

```
 no ip address
 ipv6 address 2001:db8::172:16:1:2/112
 ip access-group FIREWALL_IN in
 ip access-group FIREWALL_OUT out
 ipv6 access-group FW_INPUT_WAN_IPV6 in
 crypto map VPN
 ipv6 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 TenGigabitEthernet0/0
!
ipv6 route ::/0 2001:db8::172:16:1:1
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://[2001:db8::172:16:101:15]/certcrl.crl
 crl     download     time     60crypto     pki     certificate     chain     s-
terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Thu May 16 10:52:30 2024

GlobalParameters(
    Title                         = "This LSP was automatically generated by CSP
Converter at Thu May 16 10:52:30 2024 (user: root)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            NetworkInterface = "TenGigabitEthernet0/0"
        )
    IPv6Routes =
        Route(
            Destination = ::/0
            Gateway = 2001:db8::172:16:1:1
        )

)
```

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

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

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

FilterChain FilterChain:FIREWALL_IN:FW_INPUT_WAN_IPV6 (
    Filters = Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
        LogEventID = "FIREWALL_IN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        ProtocolID = 58
        ICMPType = 135
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = ::/0
        ProtocolID = 58
        ICMPType = 136
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:1:2
```

```
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:1:2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:1:2
        ProtocolID = 58
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = 2001:db8::172:16:1:2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN_IPV6"
    ),
    Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)
```

```
IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

IKERule IKERule:VPN:1
(
    IKEPeerIPFilter = 2001:db8::172:16:100:2, 2001:db8::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            = 10
)

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = IPv6TunnelEntry(
        PeerAddress = 2001:db8::172:16:100:2
        Prefragmentation = ENABLE
    ),
    IPv6TunnelEntry(
        PeerAddress = 2001:db8::172:17:100:2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_GOST_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:1
)

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

```
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB_AND_SPOKE"
    )
)

NetworkInterface (
    LogicalName = "TenGigabitEthernet0/0"
    InputFilter = FilterChain:FIREWALL_IN:FW_INPUT_WAN_IPV6
    OutputFilter = FilterChain:FIREWALL_OUT
    IPsecPolicy = IPsecPolicy:VPN
)
```