Резервирование провайдеров при помощи MultiWAN IPv4 (DHCP)

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 «С-Терра Шлюз» версии 5.0, для аутентификации используются 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.

Сценарий предполагает настройку резервирования каналов связи при помощи сервиса `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 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 используются ICMP запросы. Для ICMP запросов сервиса `multiwan` используются отдельные таблицы маршрутизации. Данные таблицы содержат маршруты до проверяемого хоста 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 со сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)». Далее по тексту он будет называться базовым сценарием.

Требования к инфраструктуре

1. Требования к устройствам.

   1.1. «С-Терра Шлюз».

   1.1.1. Устройства «С-Терра Шлюз» должны быть инициализированы. Подробнее на http://doc.s-terra.ru. Для «С-Терра Шлюз» версии 5.0 раздел С-Терра Шлюз -> С-Терра Шлюз 5.0 -> «Инициализация С-Терра Шлюз на вычислительных системах архитектуры Intel x86-64». Для «С-Терра Шлюз» версии 4.3 раздел С-Терра Шлюз -> С-Терра Шлюз 4.3 -> «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) , либо утилиту ca-tool, входящую в состав пакета auxiliary-tools (пакет auxiliary-tools предназначен для установки на «С-Терра Шлюз» 5.0, документация на ca-tool приведена в инструкции «Центр выпуска тестовых ГОСТ сертификатов для IPsec», документацию и пакет можно загрузить на портале https://doc.s-terra.ru).

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

1.3. HTTP сервер для распространения списка отозванных сертификатов.

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

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

2. Требования к сетевому взаимодействию.

2.1. Между устройствами стенда должна быть обеспечена IP связность.

Схема взаимодействия

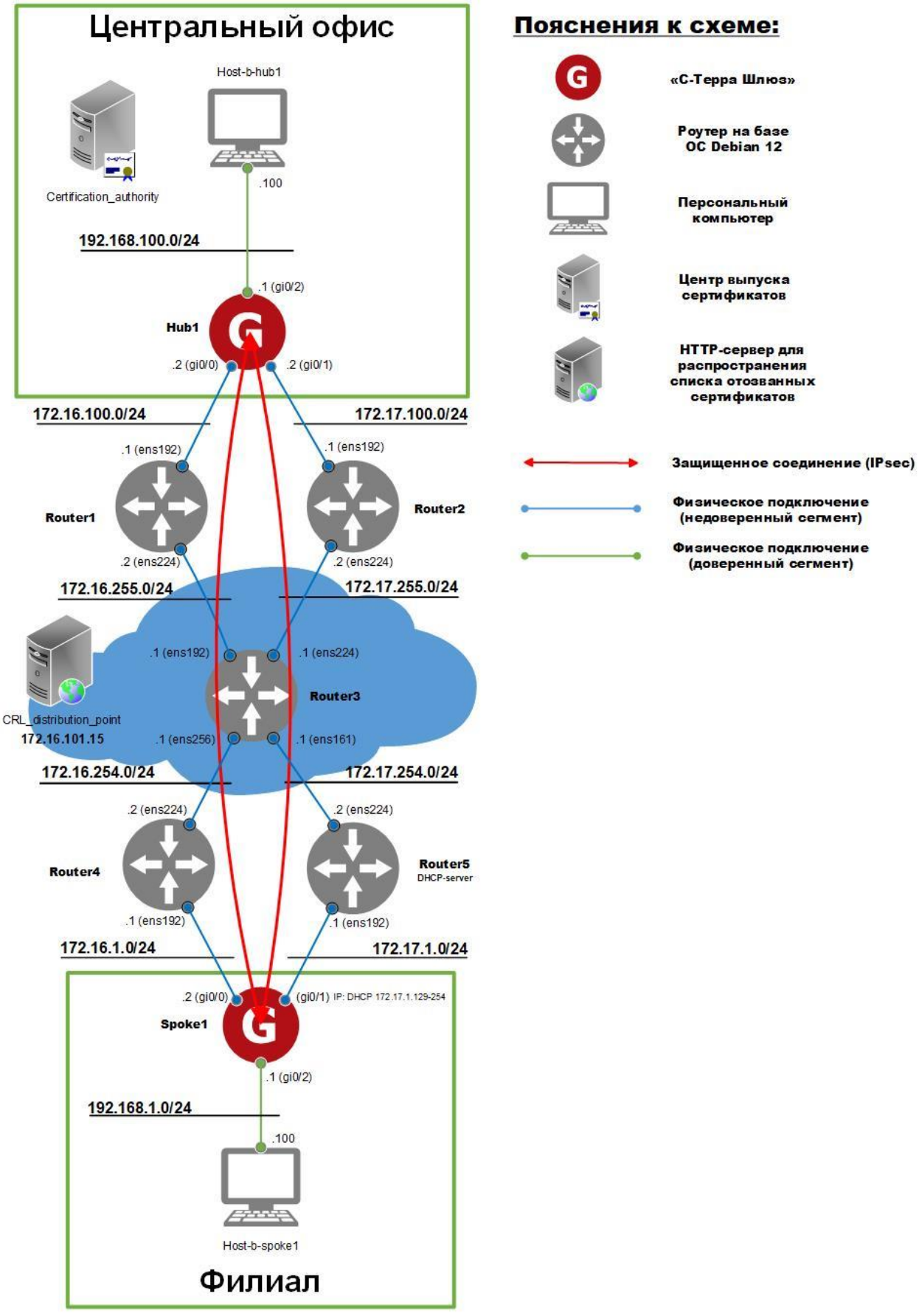


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), криптошлюз «С-Терра Шлюз» (Hub1) версии 5.0 и персональный компьютер (Host-b-hub1).

   1.2. В филиале размещаются: криптошлюз «С-Терра Шлюз» (Spoke1) версии 5.0 и персональный компьютер (Host-b-spoke1).

   1.3. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point) и маршрутизаторы (Router1, Router2, Router3, Router4, Router5).

2. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизаторы Router1, Router2, Router3, Router4, Router5.

   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 Hub1 подключается к сети Интернет с помощью сервиса `multiwan`. Основной маршрут по умолчанию через маршрутизатор Router1. Резервный маршрут по умолчанию через маршрутизатор Router2. На интерфейсы криптошлюза Hub1 назначаются статические адреса.

   2.2. Криптошлюз Spoke1 подключается к сети Интернет с помощью сервиса `multiwan`. Основной маршрут по умолчанию через маршрутизатор Router4. Резервный маршрут по умолчанию через маршрутизатор Router5. На интерфейсе основного канала связи IP адрес настроен статически. На интерфейсе резервного канала связи IP адрес и маршрут по умолчанию будут получены при помощи DHCP клиента.

3. Отказоустойчивость.

   За резервирование каналов связи на криптошлюзе Hub1 отвечает сервис `multiwan`. Сервис `multiwan` добавляет маршрут по умолчанию через Router1 или Router2. Выбор зависит от доступности каналов. Если канал через Router1 недоступен, то `multiwan` производит переключение на работу через Router2, если он доступен. Если `multiwan` работает на канале через Router2 и канал через Router1 становится доступен, то `multiwan` переключится на работу через Router1, если включен preempt режим.

   За резервирование каналов связи на криптошлюзе Spoke1 отвечает сервис `multiwan`. Сервис `multiwan` добавляет маршрут по умолчанию через Router4 или Router5. Выбор зависит от доступности каналов. Если канал через Router4 недоступен, то `multiwan` производит переключение на работу через Router5, если он доступен. Если `multiwan` работает на канале через Router5 и канал через Router4 становится доступен, то `multiwan` переключится на работу через Router4, 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 создается своя таблица с маршрутом по умолчанию через Router1 (Router4) или через Router2 (Router5) с указанием src адреса. Данная таблица будет содержать только маршрут по умолчанию, все остальные маршруты из таблиц для каналов будут удалены.

4. Параметры шифрования IPsec.

   Весь IP 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 IPv4 режиме.

   Поскольку криптошлюз сети филиала (Spoke1) получает настройки интерфейса, подключенного к резервному каналу связи, динамически, его адрес может измениться. Для подключения к центральному офису филиалов, у которых может изменяться IP адрес внешнего интерфейса, на криптошлюзе центрального офиса (Hub1) настраиваются динамические крипто-карты. В динамической крипто-карте отсутствует описание партнера по IPsec, поэтому инициировать защищенное соединение

может только трафик из подсети 192.168.1.0/24 в 192.168.100.0/24 (из филиала в центр). Инициация защищенного соединения в направлении из центрального офиса в филиал невозможна.

4.1. Параметры протокола IKE:

4.1.1. 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).

4.1.2. Параметры протокола IKE для совместимости с версией 4.3:

- Аутентификация при помощи цифровых сертификатов, алгоритм подписи – ГОСТ Р 34.10-2012 (ключ 256 бит);
- Алгоритм шифрования – ГОСТ 28147-89 (ключ 256 бит);
- Алгоритм вычисления хеш-функции – ГОСТ Р 34.11-2012 ТК26 (ключ 256 бит);
- Алгоритм выработки общего ключа (аналог алгоритма Диффи-Хеллмана) – VKO_GOSTR3410_2012_256 (ключ 256 бит).

4.1.3. 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).

4.1.4. Параметры протокола ESP для совместимости с версией 4.3:

- Комбинированный алгоритм шифрования и имитозащиты (контроль целостности) – ESP_GOST-4M-IMIT (ключ 256 бит).

5. Параметры межсетевого экранирования.

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

5.1. Для «С-Терра Шлюз» версии 5.0:

5.1.1. На LAN интерфейсах IPv4 трафик не контролируется.

5.1.2. На LAN интерфейсах IPv6 трафик заблокирован.

5.1.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

- Протокол UDP порты 500 и 4500 – для IKE и NAT-Traversal (ESP in UDP).
- Протокол ICMP – для проверки доступности криптошлюза и работы механизма PathMTUd.
- Протокол ESP (номер 50) – для работы IPsec.
- Весь TCP и UDP трафик, для которого есть исходящее соединение (это динамические правила, которые создаются при помощи stateful фильтрации).
- Протокол BOOTP для обмена DHCP пакетами между Spoke1 и Router5.

5.1.4. На WAN интерфейсах IPv6 трафик заблокирован.

5.1.5. 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).

5.2. Для «С-Терра Шлюз» версии 4.3:

5.2.1. Использование stateful фильтрации для версии 4.3 нежелательно, рекомендуется использовать статические правила (ACL).

5.2.2. На LAN и WAN интерфейсах IPv4 трафик не контролируется.

5.2.3. На LAN и WAN интерфейсах IPv6 трафик заблокирован (протокол IPv6 отключен в ядре операционной системы).

Настройка стенда

Настройка устройства Router1

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

Настройка устройства Router2

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

Настройка устройства Router3

1. Настройте IP адрес – 172.16.255.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.17.255.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Настройте IP адрес – 172.16.254.1 и маску – 255.255.255.0 на сетевом интерфейсе ens256.
4. Настройте IP адрес – 172.17.254.1 и маску – 255.255.255.0 на сетевом интерфейсе ens161.
5. Задайте маршрут до сети 172.16.100.0 0.0.0.255 через 172.16.255.2.
6. Задайте маршрут до сети 172.17.100.0 0.0.0.255 через 172.17.255.2.
7. Задайте маршрут до сети 172.16.1.0 0.0.0.255 через 172.16.254.2.
8. Задайте маршрут до сети 172.17.1.0 0.0.0.255 через 172.17.254.2.
9. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Router4

1. Настройте IP адрес – 172.16.254.2 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
2. Настройте IP адрес – 172.16.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
3. Задайте маршрут по умолчанию через 172.16.254.1.
4. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Router5

Настройка сетевых параметров

1. Настройте IP адрес – 172.17.254.2 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
2. Настройте IP адрес – 172.17.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
3. Задайте маршрут по умолчанию через 172.17.254.1.
4. Разрешите прохождение IP трафика (если заблокировано).

Настройка DHCP-сервера

Для данного сценария настраивается только выдача IP адреса и маршрута по умолчанию.

1. Установите на Router5 пакет для настройки DHCP-сервера. Это можно сделать следующей командой (у устройства должен быть доступ к сети Интернет):

```
root@Router5:~# apt-get install isc-dhcp-server
```

2. Для настройки интерфейса, на котором будут раздаваться IP-адреса, отредактируйте файл `/etc/default/isc-dhcp-server`. Имя интерфейса в переменной `INTERFACESv4` должно соответствовать имени интерфейса в нотации linux:

```
INTERFACESv4="ens192"
```

3. Измените файл `/etc/dhcp/dhcpd.conf` следующим образом:

```
server-name Router5;
ddns-update-style none;
log-facility local7;

subnet 172.17.1.0 netmask 255.255.255.0 {
   range 172.17.1.129 172.17.1.254;
   option routers 172.17.1.1;
   option subnet-mask 255.255.255.0;
   default-lease-time 21600;
   max-lease-time 28800;
 }
```

где:

- `server-name <имя>` – имя сервера;
- `log-facility <субъект>` – указание субъекта, формирующего log сообщения. Для local7 логи будут писаться в файл `/var/log/cspvpngate.log`;
- `ddns-update-style none` – отключение поддержки DDNS;
- `subnet <IP-адрес> netmask <маска>` – описание параметров сети, для которой будут раздаваться сетевые настройки;
- `range <диапазон IP-адресов>` – диапазон IP-адресов, который будет раздаваться клиентам;
- `option routers <IP-адрес>` – IP адрес шлюза по умолчанию;
- `option subnet-mask <маска>` – маска сети;
- `default-lease-time <время>` – время аренды IP-адресов (в секундах);
- `max-lease-time <время>` – максимальное время аренды;

4. Запустите сервис `isc-dhcp-server`, а также добавьте его в автозагрузку:

```
root@Router5:~#systemctl start isc-dhcp-server.service
root@Router5:~#systemctl enable isc-dhcp-server.service
```

Настройка устройства Host-b-hub1

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

Настройка устройства Host-b-spoke1

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

Настройка криптошлюза Hub1

Настройка будет происходить локально при помощи консольного подключения.

Настройка может осуществляться и удаленно (по SSH), но исключительно по доверенному каналу связи. Доверенным каналом связи может считаться канал в пределах контролируемой зоны в случае отсутствия в нем нарушителя (в нашем примере это подсеть 192.168.100.0/24 для центрального офиса и подсеть

192.168.1.0/24 для филиала). Доверенным каналом связи также считается канал, защищенный при помощи протокола IPsec (например, после настройки шифрования между подсетями 192.168.1.0/24 и 192.168.100.0/24 разрешается выполнять настройку критошлюза Spoke1 из подсети 192.168.100.0/24 центрального офиса).

Начальные настройки

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

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

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройки PKI (запросы и сертификаты)

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

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

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка паролей доступа к консоли cisco-like

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка сетевых параметров

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

1. Задайте имя устройства:

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

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

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

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

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка шифрования

Количество крипто-карт должно соответствовать количеству внешних интерфейсов (отличие между крипто-картами должно быть только в значении параметра `local-address`) и в них должна присутствовать опция `set dead connection history off`.
В случае, если криптошлюз, использующий `multiwan`, выступает в схеме только в качестве инициатора и получает сетевые параметры по DHCP, допускается отсутствие параметра `local-address` в крипто-картах.
Во всех остальных случаях использование параметра `local-address` **обязательно**.

1. Параметры IKE.

   1.1. Задайте общие параметры IKE:

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

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

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

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

```
Hub1(config)#crypto isakmp policy 2
Hub1(config-isakmp)#encryption gost
Hub1(config-isakmp)#hash gost341112-256-tc26
Hub1(config-isakmp)#authentication gost-sig
Hub1(config-isakmp)#group vko2
Hub1(config-isakmp)#exit
```

   1.4. Создайте профиль настроек ISAKMP под именем `NEW_ALGS` для совместимости с версией 5.0:

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

```
Hub1(config)#crypto isakmp profile OLD_ALGS
Hub1(conf-isa-prof)#match identity dn
Hub1(conf-isa-prof)#set identity mapping
Hub1(conf-isa-prof)#set policy 2
```

2. Параметры IPsec.

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

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

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

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

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

```
Hub1(config)#crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-
imit
Hub1(cfg-crypto-trans)#exit
```

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

```
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE1
Hub1(config-ext-nacl)#permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

2.5. Создайте динамическую крипто-карту (имя `DMAP_VPN1`, раздел 1) для построения туннелей через Router1. В динамической крипто-карте не указывается адрес партнера по IPsec:

```
Hub1(config)#crypto dynamic-map DMAP_VPN1 1
```

2.5.1. Укажите список доступа для защищаемого трафика:

```
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_AND_SPOKE1
```

2.5.2. Укажите при помощи какого набора алгоритмов (IPsec трансформ) нужно защищать трафик:

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

2.5.3. Задайте локальный IP-адрес, который будет использоваться для взаимодействия с партнерами по данной криптографической карте:

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

2.5.4. Прикрепите настройки ISAKMP (протокол IKE):

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

2.5.5. Включите префрагментацию:

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

2.5.6. Обязательно отключите историю удаленных туннелей:

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

2.6. Создайте динамическую крипто-карту для построения туннелей через Router2 (имя `DMAP_VPN2`, раздел 1):

```
Hub1(config)#crypto dynamic-map DMAP_VPN2 1
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_AND_SPOKE1
Hub1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set local-address 172.17.100.2
Hub1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
```

2.7. Опционально: для совместимости с версией 4.3 используйте динамические крипто-карты со следующими параметрами:

```
Hub1(config)#crypto dynamic-map DMAP_VPN1 1
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_AND_SPOKE1
Hub1(config-crypto-map)#set transform-set GOST_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set local-address 172.16.100.2
Hub1(config-crypto-map)#set isakmp-profile OLD_ALGS
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
Hub1(config)#crypto dynamic-map DMAP_VPN2 1
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_AND_SPOKE1
Hub1(config-crypto-map)#set transform-set GOST_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set local-address 172.17.100.2
Hub1(config-crypto-map)#set isakmp-profile OLD_ALGS
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#exit
```

2.8. Создайте статические крипто-карты и привяжите к ним динамические:

```
Hub1(config)#crypto map VPN1 1 ipsec-isakmp dynamic DMAP_VPN1
Hub1(config)#crypto map VPN2 1 ipsec-isakmp dynamic DMAP_VPN2
```

2.9. Прикрепите созданные крипто-карты к внешним интерфейсам `GigabitEthernet0/0` и `GigabitEthernet0/1`:

```
Hub1(config)#interface GigabitEthernet0/0
Hub1(config-if)#crypto map VPN1
Hub1(config-if)#exit
Hub1(config)#interface GigabitEthernet0/1
Hub1(config-if)#crypto map VPN2
Hub1(config-if)#exit
```

2.10. Примените настройки:

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

Подробная настройка с описанием команд приведена в разделе «Настройка шифрования» базового сценария.

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 разделе «Настройка политики обработки COC (CRL)» базового сценария.

Из-за отсутствия маршрута по умолчанию COC не будет 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:

```
Hub1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1(config)#object-group network SELF
Hub1(config-network-group)#remark WAN IP
Hub1(config-network-group)#host 172.16.100.2
Hub1(config-network-group)#host 172.17.100.2
Hub1(config-network-group)#exit
```

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

```
Hub1(config)#ip access-list extended FW_INPUT_WAN
Hub1(config-ext-nacl)#permit udp any object-group SELF eq isakmp
Hub1(config-ext-nacl)#permit udp any object-group SELF eq non500-isakmp
Hub1(config-ext-nacl)#permit icmp any object-group SELF
Hub1(config-ext-nacl)#permit esp any object-group SELF
Hub1(config-ext-nacl)#exit
```

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

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

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

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

5. Примените настройки:

```
Hub1(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:~#
```

Настройка `multiwan` осуществляются из linux bash консоли 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:
  links:
  - name: Router1
    iface_name: eth0
    prio: 300
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: 172.16.100.1
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.16.255.1
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
  - name: Router2
    iface_name: eth1
    prio: 200
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: 172.17.100.1
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.17.255.1
      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 название интерфейса в cisco-like консоли и в нотации linux можно посмотреть в карте интерфейсов:

```
root@Hub1:~# cat /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
```

```
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="GigabitEthernet0/3" pattern="eth3")
interface (name="default" pattern="*")
```

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@Hub1:~# /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:
...
```

При блокировке icmp трафика между криптошлюзом центрального офиса Hub1 и проверочными хостами основного канала связи, `multiwan` посчитает основной канал связи неработоспособным и переключится на работу через резервный канал связи. Соответственно, запросы на установление соединения будут приходить через основной канал связи (ISAKMP и ESP трафик), ответные пакеты - через резервный канал связи. Это приведет к тому, что на Hub1 не удастся создать общую политику безопасности (IPsec SA), исходящие с Hub1 пакеты будут отбрасываться с сообщением `failed to build SA bundle` (вывод команды `klogview -f drop`). В результате защищенное соединение построено не будет.

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

В Приложении представлены тексты конфигураций для криптошлюза Hub1:

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

Настройка криптошлюза Spoke1

Настройка криптошлюза Spoke1 происходит аналогично настройке Hub1, за исключением настройки статических крипто-карт, сетевых настроек, настройки `multiwan` и настройки межсетевого экрана.

Настройка шифрования

На Spoke1 используются только статические крипто-карты. Для каждого WAN интерфейса необходимо создать крипто-карту. В каждой должно быть указано два партнера по IPsec соединению. Также поскольку интерфейс, к которому прикрепляется крипто-карта VPN2, получает адрес при помощи DHCP, в ней не должен быть указан `local-address`.

В случае, если криптошлюз, использующий `multiwan`, выступает в схеме только в качестве инициатора и получает сетевые параметры по DHCP, допускается отсутствие параметра `local-address` в крипокартах.
Во всех остальных случаях использование параметра `local-address` **обязательно**.

Количество крипто-карт должно соответствовать количеству внешних интерфейсов (отличие между крипто-картами должно быть только в значении параметра `local-address`), и в них должна присутствовать опция `set dead connection history off`.

```
Spoke1(config)#crypto map VPN1 1 ipsec-isakmp Spoke1(config-crypto-map)#match address
IPSEC_ACL_HUB1_AND_SPOKE1
Spoke1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Spoke1(config-crypto-map)#set isakmp-profile NEW_ALGS
Spoke1(config-crypto-map)#set security-association mss-fix out
Spoke1(config-crypto-map)#set local-address 172.16.1.2
Spoke1(config-crypto-map)#set peer 172.16.100.2
Spoke1(config-crypto-map)#set peer 172.17.100.2
Spoke1(config-crypto-map)#set pre-fragmentation
Spoke1(config-crypto-map)#set dead-connection history off
Spoke1(config)#crypto map VPN2 1 ipsec-isakmp
Spoke1(config-crypto-map)#match address IPSEC_ACL_HUB1_AND_SPOKE1
Spoke1(config-crypto-map)#set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
Spoke1(config-crypto-map)#set isakmp-profile NEW_ALGS
Spoke1(config-crypto-map)#set security-association mss-fix out
Spoke1(config-crypto-map)#set peer 172.16.100.2
Spoke1(config-crypto-map)#set peer 172.17.100.2
Spoke1(config-crypto-map)#set pre-fragmentation
Spoke1(config-crypto-map)#set dead-connection history off
```

Подробная настройка с описанием команд приведена в разделе «Настройка шифрования» базового сценария.

Настройка DHCP-клиента

На криптошлюзе Spoke1 настройка DHCP-клиента будет производиться с использованием утилиты `stdhcpclient`.

Утилита `stdhcpclient` предназначена для управления получением сетевых параметров по DHCP на указанном сетевом интерфейсе. Она позволяет использовать индивидуальные конфигурационные файлы DHCP клиента для конкретных интерфейсов.

1. Выключите сетевой интерфейс, на котором планируется использование индивидуальных параметров DHCP:

   ```
   root@Spoke1:~# ifdown eth1
   ```

2. Уберите из секции начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###`, которая находится в файле `/etc/network/interfaces`, все строки, соответствующие интерфейсу eth1.

3. Добавьте настройки в файл `/etc/network/interfaces`:

   ```
   root@Spoke1:~# vim.tiny /etc/network/interfaces
   ```

```
...
###netifcfg-begin###
...
###netifcfg-end###

auto eth1
iface eth1 inet manual
post-up stdhcpclient start -i $IFACE
pre-down stdhcpclient stop -i $IFACE
mtu 1500
```

Примечание – если конфигурационный файл для интерфейса `eth2` не будет создан, утилита `stdhcpclient` завершит работу с ошибкой. Этот файл обязателен для корректной работы утилиты.

4. Создайте конфигурационный файл `/etc/dhcp/dhclient-eth1.conf` на основе шаблона `/opt/stmodems/docs/dhclient-usb0.conf.example`:

```
# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
#       man page for more information about the syntax of this file
#       and a more comprehensive list of the parameters understood by
#       dhclient.
#
# Normally, if the DHCP server provides reasonable information and does
#       not leave anything out (like the domain name, for example), then
#       few changes must be made to this file, if any.
#

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;

send host-name = gethostname();
request subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, domain-search, host-name,
        dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
        netbios-name-servers, netbios-scope, interface-mtu,
        rfc3442-classless-static-routes, ntp-servers;
```

5. Включите сетевой интерфейс:

```
root@Spoke1:~# ifup eth1
```

6. Посмотрите выданный адрес:

```
root@Spoke1:~# ip address show dev eth1
```

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 10000
    link/ether 00:50:56:9e:99:fd brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname GigabitEthernet0_1
    inet 172.17.1.129/24 brd 172.17.1.255 scope global dynamic eth1
       valid_lft 21055sec preferred_lft 21055sec
    inet6 fe80::d9e5:f178:9282:b348/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Настройка сервиса multiwan

1. Для корректной работы multiwan с интерфейсом, получающим сетевые настройки по DHCP (eth1), задайте в качестве значения параметра `nexthop` значение `DHCP`:

```
nexthop: DHCP
```

2. Файл конфигурации `/etc/multiwan/config.yaml` должен иметь следующий вид:

```
config_format_version: 1
links_params:
  links:
  - name: Router4
    iface_name: eth0
    prio: 300
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: 172.16.1.1
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.16.254.1
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
  - name: Router5
    iface_name: eth1
    prio: 200
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: DHCP
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.17.254.1
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
```

В файл `/tmp/multiwan_dhcp-hook.debug` записываются сообщения о работе DHCP hook. Чтобы избежать его заполнения, на криптошлюзе Spoke1 отредактируйте DHCP hook файлы `/etc/dhcp/dhclient-exit-hooks.d/98-multiwan` и `/etc/dhcp/dhclient-enter-hooks.d/01-multiwan`. Задайте переменной DEBUG значение no - `DEBUG="no"`.

Настройка межсетевого экрана

1. Создайте сетевой `object-group` с именем `SELF` для хранения в нем WAN адресов криптошлюза. Поскольку интерфейс резервного канала получает настройки динамически, в качестве объекта в

`object-group` с именем `SELF` следует добавить не адрес интерфейса, а подсеть, из которой будут назначаться адреса (определяется настройками DHCP сервера):

```
Spoke1(config)#object-group network SELF
Spoke1(config-network-group)#remark WAN IP
Spoke1(config-network-group)#172.17.1.128 255.255.255.128
Spoke1(config-network-group)#host 172.16.1.2
Spoke1(config-network-group)#exit
```

2. Создайте ACL с именем `FW_INPUT_WAN` для входящего на WAN интерфейс трафика. Для корректной работы DHCP следует добавить правило, разрешающее прохождение DHCP пакетов:

```
Spoke1(config)#ip access-list extended FW_INPUT_WAN
Spoke1(config-ext-nacl)#permit udp any object-group SELF eq isakmp
Spoke1(config-ext-nacl)#permit udp any object-group SELF eq non500-isakmp
Spoke1(config-ext-nacl)#permit icmp any object-group SELF
Spoke1(config-ext-nacl)#permit esp any object-group SELF
Spoke1(config-ext-nacl)#permit udp host 172.17.1.1 object-group SELF eq bootpc
Spoke1(config-ext-nacl)#exit
```

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст консоли cisco-like;
- текст LSP;
- содержимое `/etc/multiwan/config.yaml`.
- содержание файла `/etc/network/interfaces`.

Проверка работоспособности стенда

Проверку работоспособности нужно выполнять после завершения настройки всех устройств стенда.

Проверка связности

1. Проверьте, что с криптошлюза Hub1 по ICMP доступны маршрутизаторы провайдеров (Router1 и Router2), и с криптошлюза Spoke1 по ICMP доступны маршрутизаторы провайдеров (Router4 и Router5). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub1#ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 56(84) bytes of data.
64 bytes from 172.16.100.1: icmp_seq=1 ttl=64 time=0.312 ms
64 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.696 ms
64 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.492 ms
64 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.338 ms
64 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.456 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4070ms
rtt min/avg/max/mdev = 0.312/0.458/0.696/0.136 ms
```

```
Hub1#ping 172.17.100.1
```

```
PING 172.17.100.1 (172.17.100.1) 56(84) bytes of data.
64 bytes from 172.17.100.1: icmp_seq=1 ttl=64 time=0.408 ms
64 bytes from 172.17.100.1: icmp_seq=2 ttl=64 time=0.361 ms
64 bytes from 172.17.100.1: icmp_seq=3 ttl=64 time=0.372 ms
64 bytes from 172.17.100.1: icmp_seq=4 ttl=64 time=0.357 ms
64 bytes from 172.17.100.1: icmp_seq=5 ttl=64 time=0.522 ms

--- 172.17.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4078ms
rtt min/avg/max/mdev = 0.357/0.404/0.522/0.061 ms
```

```
Spoke1#ping 172.16.1.1
```

```
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.286 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.473 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.536 ms
64 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.479 ms
64 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.510 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.286/0.456/0.536/0.088 ms
```

```
Spoke1#ping 172.17.1.1
```

```
PING 172.17.1.1 (172.17.1.1) 56(84) bytes of data.
64 bytes from 172.17.1.1: icmp_seq=1 ttl=64 time=0.376 ms
64 bytes from 172.17.1.1: icmp_seq=2 ttl=64 time=0.478 ms
64 bytes from 172.17.1.1: icmp_seq=3 ttl=64 time=0.596 ms
64 bytes from 172.17.1.1: icmp_seq=4 ttl=64 time=0.461 ms
64 bytes from 172.17.1.1: icmp_seq=5 ttl=64 time=0.647 ms

--- 172.17.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4052ms
rtt min/avg/max/mdev = 0.376/0.511/0.647/0.097 ms
```

Вывод команды свидетельствует о том, что устройства Router1 и Router2 доступны по ICMP с Hub1, а устройства Router4 и Router5 доступны по ICMP со Spoke1.

2. Проверьте, что с защищаемых устройств Host-b-spoke1 и Host-b-hub1 доступен по ICMP соответствующий криптошлюз, который является шлюзом по умолчанию. Для этого выполните команду `ping` из linux bash консоли защищаемых устройств.

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

```
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.386 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.512 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.385 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=0.380 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=64 time=0.537 ms

--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.380/0.440/0.537/0.069 ms
```

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

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.557 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.439 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.481 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.483 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.486 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.439/0.489/0.557/0.038 ms
```

Видно, что с защищаемых устройств Host-b-spoke1 и Host-b-hub1 соответствующие криптошлюзы доступны по ICMP.

3. На криптошлюзе Spoke1 проверьте с помощью команды `show interface`, какой адрес был выдан интерфейсу, направленному в сторону резервного канала связи:

```
Spoke1#show interfaces gigabitEthernet 0/1
```

```
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.2899
  Internet address is 172.17.1.129/24
  MTU 1500 bytes
```

4. Проверьте, что с криптошлюза Hub1 доступен по ICMP криптошлюз Spoke1 (по внешним интерфейсам). Для этого выполните команду `ping` из cisco-like консоли криптошлюзов.

```
Hub1#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=0.709 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=61 time=0.779 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=61 time=0.981 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=61 time=0.852 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=61 time=0.820 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4027ms
rtt min/avg/max/mdev = 0.709/0.828/0.981/0.090 ms
```

```
Hub1#ping 172.17.1.129
```

```
PING 172.17.1.129 (172.17.1.129) 100(128) bytes of data.
108 bytes from 172.17.1.129: icmp_seq=1 ttl=61 time=1.02 ms
108 bytes from 172.17.1.129: icmp_seq=2 ttl=61 time=0.886 ms
108 bytes from 172.17.1.129: icmp_seq=3 ttl=61 time=1.06 ms
108 bytes from 172.17.1.129: icmp_seq=4 ttl=61 time=0.898 ms
108 bytes from 172.17.1.129: icmp_seq=5 ttl=61 time=0.908 ms
```

```
--- 172.17.1.129 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 0.886/0.953/1.061/0.070 ms
```

Видно, что с криптошлюза Hub1 доступны по ICMP оба интерфейса Spoke1.

5. Проверьте на криптошлюзах Spoke1 и Hub1 наличие маршрутов через интерфейс, направленный в сторону основного канала связи.

```
Hub1#run ip route show proto multiwan
```

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

```
Spoke1#run ip route show proto multiwan
```

```
default via 172.16.1.1 dev eth0 metric 700
```

Проверка PKI

1. Проверьте, что на криптошлюзах Hub1 и Spoke1 СОС импортирован в базу Продукта. Для этого выполните команду `run cert_mgr show` из cisco-like консоли криптошлюзов.

```
Hub1#run cert_mgr show
```

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

```
Spoke1#run cert_mgr show
```

```
Found 3 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=Spoke1
3 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

Видно, что на криптошлюзах Hub1 и Spoke1 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 Hub1 и Spoke1 статус всех сертификатов Active. Для этого выполните команду `run cert_mgr check` из cisco-like консоли криптошлюзов.

```
Hub1#run cert_mgr check
```

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

```
Spoke1#run cert_mgr check
```

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

Видно, что на криптошлюзах Hub1 и Spoke1 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 Hub1 и Spoke1

1. Инициируйте построение защищенного соединения между криптошлюзами Hub1 и Spoke1 при помощи ICMP трафика, посылаемого с устройства Host-b-spoke1 на Host-b-hub1. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke1.

```
root@Host-b-spoke1:~# 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=25.1 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=2.66 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=2.59 ms
```

```
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=3.89 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=2.76 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4039ms
rtt min/avg/max/mdev = 2.592/7.404/25.123/8.872 ms
```

2. Проверьте, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 5 (172.16.100.2,500)-(172.16.1.2,500) active 2672 2900

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 7 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 440 440
```

```
Spoke1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 8 (172.16.1.2,500)-(172.16.100.2,500) active 2900 2672

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 7 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn 440 440
```

Видно, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка состояния каналов связи

1. Проверьте с помощью средств `multiwan` состояние каналов связи на криптошлюзе Hub1:

```
Hub1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: Router1
    Interface: eth0
    Priority: 300
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.255.1"
    IPv4 nexthop: "172.16.100.1"
    IPv6 hosts: None
    IPv6 nexthop: None
Link: 2
    Name: Router2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.17.255.1"
    IPv4 nexthop: "172.17.100.1"
    IPv6 hosts: None
```

```
    IPv6 nexthop: None
```

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

2. Проверьте с помощью средств `multiwan` состояние каналов связи на криптошлюзе Spoke1:

```
Spoke1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: Router4
    Interface: eth0
    Priority: 300
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.254.1"
    IPv4 nexthop: "172.16.1.1"
    IPv6 hosts: None
    IPv6 nexthop: None
Link: 2
    Name: Router5
    Interface: eth1
    Priority: 200
    State: reachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.17.254.1"
    IPv4 nexthop: "172.17.1.1" (DHCP)
    IPv6 hosts: None
    IPv6 nexthop: None
```

Видно, что первый канал связи, через eth0 с приоритетом 300 доступен. Является активным. Доступность канала проверяется доступностью любого хоста из числа IPv4 хостов (секция `check`). Второй канал связи, через eth1 с приоритетом 200 доступен. Является неактивным. Доступность канала проверяется доступностью любого хоста из числа IPv4 хостов (секция `check`). Nexthop-адрес получен по DHCP.

Проверка отказоустойчивости

Проверка работоспособности при отказе основного канала связи на стороне Spoke1

1. Проверьте отказоустойчивость построенного соединения на стороне Spoke1. Для этого смоделируйте отказ основного провайдера Spoke1 (Router4). Это можно сделать отключением основного провайдера Router4 или отключением интерфейса eth0 на Spoke1.
2. Подождите пока на Spoke1 переключится работа `multiwan` через резервный канал. Проверьте наличие маршрутов через резервный интерфейс.

```
Spoke1#run ip route show proto multiwan
```

```
default via 172.17.1.1 dev eth1 metric 800
```

3. Вновь постройте защищенное соединение. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke1.

```
root@Host-b-spoke1:~# ping 192.168.100.100
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=6 ttl=62 time=5636 ms
64 bytes from 192.168.100.100: icmp_seq=7 ttl=62 time=4610 ms
64 bytes from 192.168.100.100: icmp_seq=8 ttl=62 time=3584 ms
```

```
64 bytes from 192.168.100.100: icmp_seq=12 ttl=62 time=1.98 ms
64 bytes from 192.168.100.100: icmp_seq=13 ttl=62 time=7.12 ms
64 bytes from 192.168.100.100: icmp_seq=14 ttl=62 time=2.24 ms
64 bytes from 192.168.100.100: icmp_seq=15 ttl=62 time=1.93 ms
64 bytes from 192.168.100.100: icmp_seq=16 ttl=62 time=2.31 ms
64 bytes from 192.168.100.100: icmp_seq=17 ttl=62 time=2.98 ms
64 bytes from 192.168.100.100: icmp_seq=18 ttl=62 time=2.84 ms
^C
--- 192.168.100.100 ping statistics ---
18 packets transmitted, 15 received, 16.6667% packet loss, time 17318ms
rtt min/avg/max/mdev = 1.931/3827.047/10764.770/3967.023 ms, pipe 11м
```

4. Проверьте, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение через резервный провайдер Router5. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 8 (172.16.100.2,500)-(172.17.1.129,500) active 1928 1960

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 1320
1320
```

```
Spoke1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 52 (172.17.1.129,500)-(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 10 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn 5632
5632
```

Видно, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение через резервного провайдера Spoke1 – Router5.

5. Проверьте с помощью средств `multiwan` состояние каналов связи на криптошлюзе Spoke1:

```
Spoke1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: Router4
    Interface: eth0
    Priority: 300
    State: unreachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.254.1"
    IPv4 nexthop: "172.16.1.1"
    IPv6 hosts: None
    IPv6 nexthop: None
Link: 2
    Name: Router5
    Interface: eth1
    Priority: 200
    State: reachable
    Active: True
```

```
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.17.254.1"
    IPv4 nexthop: "172.17.1.1" (DHCP)
    IPv6 hosts: None
    IPv6 nexthop: 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.

Проверка работоспособности при отказе основного канала связи на стороне Hub1

1. Проверьте отказоустойчивость построенного соединения на стороне Hub1. Для этого смоделируйте отказ основного провайдера Hub1 (Router1). Это можно сделать отключением основного провайдера Router1 или отключением интерфейса eth0 на Hub1.
2. Подождите пока на Hub1 переключится работа `multiwan` через резервный канал. Проверьте наличие маршрутов через резервный интерфейс.

```
Hub1#run ip route show proto multiwan
```

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

3. Вновь постройте защищенное соединение. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke1.

```
root@Host-b-spoke1:~# ping 192.168.100.100
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=39 ttl=62 time=2695 ms
64 bytes from 192.168.100.100: icmp_seq=40 ttl=62 time=1671 ms
64 bytes from 192.168.100.100: icmp_seq=41 ttl=62 time=647 ms
64 bytes from 192.168.100.100: icmp_seq=42 ttl=62 time=3.12 ms
64 bytes from 192.168.100.100: icmp_seq=43 ttl=62 time=3.90 ms
64 bytes from 192.168.100.100: icmp_seq=44 ttl=62 time=3.77 ms
64 bytes from 192.168.100.100: icmp_seq=45 ttl=62 time=3.74 ms
64 bytes from 192.168.100.100: icmp_seq=46 ttl=62 time=4.33 ms
64 bytes from 192.168.100.100: icmp_seq=47 ttl=62 time=2.90 ms
64 bytes from 192.168.100.100: icmp_seq=48 ttl=62 time=3.35 ms
64 bytes from 192.168.100.100: icmp_seq=49 ttl=62 time=3.38 ms
^C
--- 192.168.100.100 ping statistics ---
49 packets transmitted, 16 received, 67.3469% packet loss, time 48952ms
rtt min/avg/max/mdev = 2.904/2117.059/7814.035/2686.746 ms, pipe 8
```

4. Проверьте, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение через резервный провайдер Router5. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

```
Hub1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 9 (172.17.100.2,500)-(172.17.1.129,500) 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 3784
3784
```

```
Spoke1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
```

```
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 55 (172.17.1.129,500)-(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 11 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn 968
968
```

Видно, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение через резервного провайдера Spoke1 – Router5 и резервного провайдера Hub1 – Router2.

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

```
Hub1#run /opt/multiwan/mwan_mgr show --links_status
```

```
Link: 1
    Name: Router1
    Interface: eth0
    Priority: 300
    State: unreachable
    Active: False
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.16.255.1"
    IPv4 nexthop: "172.16.100.1"
    IPv6 hosts: None
    IPv6 nexthop: None
Link: 2
    Name: Router2
    Interface: eth1
    Priority: 200
    State: reachable
    Active: True
    Hosts reachability: any
    IPv4 hosts: "172.16.101.15", "172.17.255.1"
    IPv4 nexthop: "172.17.100.1"
    IPv6 hosts: None
    IPv6 nexthop: 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.

Проверка работоспособности при восстановлении основных каналов связи

1. Для проверки обратного переключения с резервных каналов связи на основные смоделируйте восстановление основных провайдеров Router1 и Router4. Включите провайдеры, если они были выключены, или включите основные интерфейсы шлюзов Spoke1 и Hub1.

2. Подождите пока на криптошлюзах произойдет переключение работы `multiwan` через основной канал. Проверьте на каждом криптошлюзе наличие маршрута через основной канал связи.

```
Hub1#run ip route show proto multiwan
```

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

```
Spoke1#run ip route show proto multiwan
```

```
default via 172.16.1.1 dev eth0 metric 700
```

3. Вновь постройте защищенное соединение. Для этого выполните команду ping из linux bash консоли защищаемого устройства Host-b-spoke1.

```
root@ Host-b-spoke1:~# ping 192.168.100.100
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=11 Destination Net Unreachable
From 192.168.1.1 icmp_seq=12 Destination Net Unreachable
```

```
64 bytes from 192.168.100.100: icmp_seq=13 ttl=62 time=997 ms
64 bytes from 192.168.100.100: icmp_seq=14 ttl=62 time=1.04 ms
64 bytes from 192.168.100.100: icmp_seq=15 ttl=62 time=3.85 ms
64 bytes from 192.168.100.100: icmp_seq=16 ttl=62 time=3.36 ms
64 bytes from 192.168.100.100: icmp_seq=17 ttl=62 time=2.30 ms
64 bytes from 192.168.100.100: icmp_seq=18 ttl=62 time=4.07 ms
64 bytes from 192.168.100.100: icmp_seq=19 ttl=62 time=3.90 ms
64 bytes from 192.168.100.100: icmp_seq=20 ttl=62 time=3.48 ms
64 bytes from 192.168.100.100: icmp_seq=21 ttl=62 time=3.25 ms
^C
--- 192.168.100.100 ping statistics ---
21 packets transmitted, 9 received, +2 errors, 57.1429% packet loss, time 20241ms
rtt min/avg/max/mdev = 1.043/113.580/996.969/312.326 ms
```

4. Проверьте, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение через основные провайдеры Router1 и Router4. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюзов.

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

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

Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 29 (172.16.100.2,500)-(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  23  (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*)  *  ESP  tunn
792 792
```

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

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 143 (172.16.1.2,500)-(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  (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn 792 792
```

Видно, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение через основного провайдера Spoke1 – Router4 и основного провайдера Hub1 – Router1.

Приложение

Конфигурации криптошлюза Hub1

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 secret 5 $y$j9T$TuRUHDE2W7KORtooiCU43/$7O07G8ISCUH
qPFUHYldKUfV5GQSh3eKTmcPq4zTHWE5
aaa new-model
!
!
hostname Hub1
enable secret 5 JLI9qp52IVCYdKnFL5Xk/A==!
!
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_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
!
object-group network SELF
 remark WAN IP
 host 172.16.100.2
 host 172.17.100.2
!
ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE1
 permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
!
```

```
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto dynamic-map DMAP_VPN1 1
 match address IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set local-address 172.16.100.2
 set pre-fragmentation
 set dead-connection history off
!
crypto dynamic-map DMAP_VPN2 1
 match address IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set local-address 172.17.100.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN1 1 ipsec-isakmp dynamic DMAP_VPN1
!
crypto map VPN2 1 ipsec-isakmp dynamic DMAP_VPN2
!
interface GigabitEthernet0/0
 ip address 172.16.100.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN1
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 172.17.100.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN2
 ip inspect INSPECT out
!
interface GigabitEthernet0/2
 ip address 192.168.100.1 255.255.255.0
!
interface GigabitEthernet0/3
 no ip address
 shutdown
!
!
!
!
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
. . .
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Fri Mar 28 13:39:35 2025

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Fri Mar 28 13:39:35 2025 (user: cscons)"
    Version                       = LSP_5_0
    CRLHandlingMode               = ENABLE
    PreserveIPsecSA               = FALSE
)

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

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

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

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

```
    )
)

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

```
        Action = DROP
    ))

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

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

IKERule IKERule:VPN1:1:DMAP_VPN1:1
(
```

```
    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            = 100
)

IKERule IKERule:VPN2:1:DMAP_VPN2:1
(
    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            = 110
)

IPsecAction IPsecAction:VPN1:1:DMAP_VPN1:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN1:1:DMAP_VPN1:1
)

IPsecAction IPsecAction:VPN2:1:DMAP_VPN2:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.17.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN2:1:DMAP_VPN2:1
)

FilterChain IPsecPolicy:VPN1 (
    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:VPN1:1:DMAP_VPN1:1 >
        LogEventID = "IPsec:Protect:VPN1:1:DMAP_VPN1:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    )
)
```

```
FilterChain IPsecPolicy:VPN2 (
    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:VPN2:1:DMAP_VPN2:1 >
        LogEventID = "IPsec:Protect:VPN2:1:DMAP_VPN2:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN1
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN2
)

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

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

3. Конфигурация `multiwan`:

```
config_format_version: 1
links_params:
  links:
  - name: Router1
    iface_name: eth0
    prio: 300
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: 172.16.100.1
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.16.255.1
      method:
        icmp:
```

```
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
  - name: Router2
    iface_name: eth1
    prio: 200
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: 172.17.100.1
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.17.255.1
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
```

Конфигурации криптошлюза Spoke1

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 secret 5 $y$j9T$TuRUHDE2W7KORtooiCU43/$7O07G8ISCUH¶
qPFUHYldKUfV5GQSh3eKTmcPq4zTHWE5
aaa new-model
!
!
hostname Spoke1
enable secret 5 JLI9qp52IVCYdKnFL5Xk/A==
!
!
!
!
!
!
!
```

```
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.1.2
 172.17.1.128 255.255.255.128
!
ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE1
 permit ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
 permit udp host 172.17.1.1 object-group SELF eq bootpc
!
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 VPN1 1 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set local-address 172.16.1.2
 set peer 172.16.100.2
 set peer 172.17.100.2
 set pre-fragmentation
 set dead-connection history off
!
crypto map VPN2 1 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 set peer 172.17.100.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.1.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN1
 ip inspect INSPECT out
```

```
!
interface GigabitEthernet0/1
 ip address 172.17.1.129 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN2
 ip inspect INSPECT out
!
interface GigabitEthernet0/2
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/3
 no ip address
 shutdown
!
!
!
!
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
. . .
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Tue May  6 16:43:40 2025

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Tue May  6 16:43:40 2025 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

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

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

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

FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.1.128/25
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.1.128/25
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.1.128/25
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.1.2
        ProtocolID = 50
```

```
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.17.1.128/25
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 172.17.1.1
        DestinationIP = 172.16.1.2
        ProtocolID = 17
        DestinationPort = 68
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        SourceIP = 172.17.1.1
        DestinationIP = 172.17.1.128/25
        ProtocolID = 17
        DestinationPort = 68
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    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 = 17
        Action = PASS
        ExtendedAction = inspect_udp<>
        LogEventID = "Inspect:INSPECT:udp"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = PASS
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

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

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

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

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

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

IKERule IKERule:VPN1:1
(
    IKEPeerIPFilter = 172.16.100.2, 172.17.100.2
    IKELocalIPFilter = 172.16.1.2
    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:VPN2:1
(
    IKEPeerIPFilter = 172.16.100.2, 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:VPN1:1
(
    TunnelingParameters = TunnelEntry(
        LocalIPAddress = 172.16.1.2
        PeerAddress = 172.16.100.2
        Prefragmentation = ENABLE
    ),
    TunnelEntry(
```

```
        LocalIPAddress = 172.16.1.2
        PeerAddress = 172.17.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN1:1
)

IPsecAction IPsecAction:VPN2:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
        Prefragmentation = ENABLE
    ),
    TunnelEntry(
        PeerAddress = 172.17.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN2:1
)

FilterChain IPsecPolicy:VPN1 (
    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:VPN1:1 >
        LogEventID = "IPsec:Protect:VPN1:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    )
)

FilterChain IPsecPolicy:VPN2 (
    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:VPN2:1 >
        LogEventID = "IPsec:Protect:VPN2:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
```

```
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN1
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN2
)

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

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

3. Конфигурация `multiwan`:

```
config_format_version: 1
links_params:
  links:
  - name: Router4
    iface_name: eth0
    prio: 300
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: 172.16.1.1
    check:
      hosts_reachability: any
      ipv4:
        hosts:
        - 172.16.101.15
        - 172.16.254.1
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
  - name: Router5
    iface_name: eth1
    prio: 200
    routes:
      ipv4:
        networks:
        - 0.0.0.0/0
        nexthop: DHCP
    check:
      hosts_reachability: any
      ipv4:
```

```
        hosts:
        - 172.16.101.15
        - 172.17.254.1
      method:
        icmp:
          max_rtt: 100
          interval: 3
          restore_interval: 1
          restore_count: 10
          failure_count: 10
          tos: 184
          data_size: 56
```

4. Содержание файла `/etc/network/interfaces`:

```
############################################################
# CAUTION: lines under special marker: ###netifcfg-*###
# contains autogenerated information. You can add/modify
# lines outside of those markers
############################################################

# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

# loopback configuration
auto lo
iface lo inet loopback

###netifcfg-begin###

auto eth0
iface eth0 inet static
mtu 1500
address 172.16.1.2
netmask 255.255.255.0
broadcast 172.16.1.255


auto eth2
iface eth2 inet static
mtu 1500
address 192.168.1.1
netmask 255.255.255.0
broadcast 192.168.1.255

auto eth3
iface eth3 inet manual
mtu 1500
###netifcfg-end###

auto eth1
iface eth1 inet manual
post-up stdhcpclient start -i $IFACE
pre-down stdhcpclient stop -i $IFACE
mtu 1500
```