Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)

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

Партнеры по IPsec, в настоящем документе – криптошлюзы, для аутентификации будут использовать сертификаты в формате x509. В качестве криптопровайдера будет использована криптографическая библиотека, разработанная компанией «С-Терра СиЭсПи». Криптошлюзы в центральном офисе и филиале № 1 версии 5.0, в филиале № 2 версии 4.3.

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.

В рамках сценария, подсети филиалов между собой безопасно взаимодействовать не могут. Настройка безопасного взаимодействия подсетей филиалов между собой с перешифрованием в центре будет описана в главе «Дополнительные возможности».

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.

Требования к инфраструктуре

1. Требования к устройствам.

   1.1. «С-Терра Шлюз».

      1.1.1. Устройства «С-Терра Шлюз» должны быть инициализированы (подробнее на https://doc.s-terra.ru раздел С-Терра Шлюз -> С-Терра Шлюз <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), 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

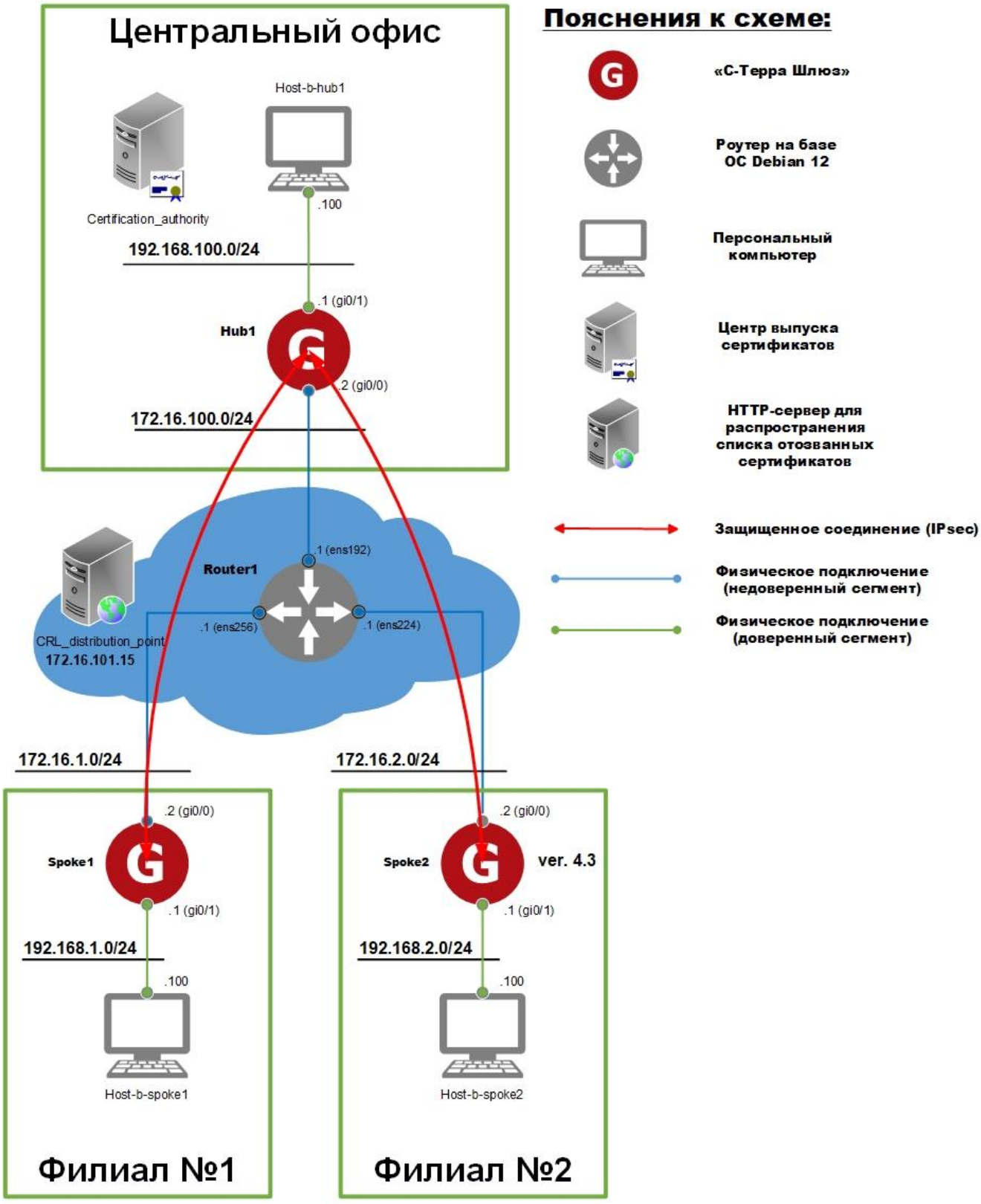


Рисунок 1. Схема взаимодействия

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 № 1 размещаются: криптошлюз «С-Терра Шлюз» (Spoke1) версии 5.0 и персональный компьютер (Host-b-spoke1).

   1.3. В филиале № 2 размещаются: криптошлюз «С-Терра Шлюз» (Spoke2) версии **4.3** и персональный компьютер (Host-b-spoke2).

   1.4. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point), маршрутизатор (Router1).

2. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизатор Router1.

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

   2.1. Криптошлюз Hub1 подключается к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router1).

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

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

3. Параметры шифрования IPsec.

   Весь IPv4 трафик между подсетями центрального офиса (192.168.100.0/24) и филиалов (192.168.1.0/24, 192.168.2.0/24) защищается с использованием алгоритмов ГОСТ и протокола IPsec в туннельном IPv4 режиме.

   Инициировать защищенное соединение между криптошлюзами Hub1 и Spoke1 может трафик из подсети филиала (192.168.1.0/24) в подсеть центра (192.168.100.0/24) и наоборот из центра в филиал.

   Инициировать защищенное соединение между криптошлюзами Hub1 и Spoke2 может трафик из подсети филиала (192.168.2.0/24) в подсеть центра (192.168.100.0/24) и наоборот из центра в филиал.

   3.1. Параметры протокола IKE

      3.1.1. 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).

      3.1.2. 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).

   3.2. Параметры протокола ESP

3.2.1. 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).

3.2.2. Параметры протокола ESP для совместимости с версией 4.3:

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

4. Параметры межсетевого экранирования.

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

4.1. Для «С-Терра Шлюз» версии 5.0:

4.1.1. На LAN интерфейсах IPv4 трафик не контролируется.

4.1.2. На LAN интерфейсах IPv6 трафик заблокирован.

4.1.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

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

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

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

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

4.1.4. На WAN интерфейсах IPv6 трафик заблокирован.

4.1.5. 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).

4.2. Для «С-Терра Шлюз» версии 4.3:

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

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

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

Настройка стенда

Настройка устройства Router1

1. Настройте IP адрес – 172.16.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens256.
2. Настройте IP адрес – 172.16.100.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
3. Настройте IP адрес – 172.16.2.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
4. Разрешите прохождение IP трафика (если заблокировано).

Настройка устройства Host-b-hub1

1. Настройте IP адрес – 192.168.100.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.100.1.
3. Разрешите прием и отправку ICMP пакетов (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 устройства Host-b-spoke2

1. Настройте IP адрес – 192.168.2.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.2.1.
3. Разрешите прием и отправку ICMP пакетов (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 № 1 и 192.168.2.0/24 для филиала № 2). Доверенным каналом связи также считается канал, защищенный при помощи протокола IPsec (например, после настройки шифрования между подсетью 192.168.100.0/24 и подсетями 192.168.1.0/24, 192.168.2.0/24 разрешается выполнять настройку критошлюзов Spoke1 и Spoke2 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.

1. Войдите в CLI разграничения доступа. Для этого, после появления сообщения:

   ```
   S-Terra administrative console
   ```

   введите логин и пароль для CLI разграничения доступа:

   Пользователь и пароль по умолчанию: `administrator`, `s-terra`. Обязательно смените пароль для пользователя `administrator` при помощи команды `change user password`.

   ```
   login as: administrator
   ```

   ```
   administrator's password:
   administrator@sterragate]
   ```

   CLI разграничения доступа используется для локальной аутентификации пользователя, прохождения инициализации и переключения в другие консоли криптошлюза.

2. Установите правильный тип терминала (для `putty` тип терминала `xterm`) и требуемую ширину (для удобства работы), например:

   ```
   administrator@sterragate] terminal terminal-type xterm
   administrator@sterragate] terminal width 150
   ```

3. Установите нужную временную зону и правильные дату и время на криптошлюзе, используя консоль linux bash. Для этого выполните следующие команды.

   3.1. Войдите в linux bash.

   ```
   administrator@sterragate] system
   ```

```
Entering system shell...
```

3.2. Установите нужную временную зону:

```
root@sterragate:~# dpkg-reconfigure tzdata
```

3.3. Установите правильное время и дату (формат – месяц/день/год часы:минуты):

```
root@sterragate:~# date -s "07/04/2023 12:32"
```

```
Thu Jul  4 12:32:00 MSK 2023
```

4. Установите надежный пароль для пользователя root (под данным пользователем осуществляется доступ по SSH в linux bash), если не сделали это ранее:

```
root@sterragate:~# passwd
```

```
New password:
Retype new password:
passwd: password updated successfully
```

5. Выйдите из linux bash обратно в CLI разграничения доступа:

```
root@sterragate:~# exit
```

```
logout
Leaving system shell...
administrator@sterragate]
```

Начальные настройки завершены.

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).

Закрытый ключ для сертификата криптошлюза будет сгенерирован при помощи утилиты `cert_mgr` с использованием биологического датчика случайных чисел (БИО ДСЧ). Если на криптошлюзе установлен аппаратный датчик случайных чисел, то для выработки случайных чисел по умолчанию будет использоваться аппаратный датчик. Закрытый ключ может храниться либо на файловой системе устройства, либо на защищенном ключевом носителе (токен). В данном сценарии закрытый ключ будет располагаться в специальном контейнере на файловой системе устройства. Если требуется, чтобы контейнер располагался на токене, то смотрите описание параметра `create` утилиты `cert_mgr` на портале документации http://doc.s-terra.ru.

В момент генерации ключевой пары будет также сгенерирован запрос на локальный сертификат криптошлюза. Данный запрос для его последующей доставки на УЦ будет сохранен на USB Flash накопитель. Сохранение запроса на USB Flash накопитель необходимо для случая, когда вывод криптошлюза происходит на экран монитора, если вывод происходит на консоль и есть возможность выполнить копирование и вставку, то текст запроса может быть выведен на консоль в кодировке BASE64.

Настройка осуществляется в CLI разграничения доступа.

При выполнении сторонних команд в CLI разграничения доступа/консоли cisco-like перед командой нужно указывать ключевое слово `run`.

1. Генерация закрытого ключа и запроса на сертификат криптошлюза.

1.1. Вставьте USB Flash накопитель в свободный USB порт криптошлюза (накопитель будет автоматически примонтирован).

1.2. Определите имя (идентификатор) USB Flash накопителя на криптошлюзе:

```
administrator@sterragate] dir media:
```

```
    1  dr-x         4096  Thu Jul  4 10:40:32 2019  1E181F1C181EF28F
```

Имя (идентификатор) USB Flash накопителя - **1E181F1C181EF28F.**

1.3. Запустите процесс генерации закрытого ключа и запроса на сертификат криптошлюза с сохранением файла запроса на USB Flash накопителе (закрытый ключ остается на криптошлюзе):

```
administrator@sterragate] run cert_mgr create -subj "C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1" -GOST_R341012_256 -fb64 media:1E181F1C181EF28F/hub1.request
```

- ключ `-subj` задает отличительное имя сертификата (Distinguished Name, DN);

Отличительное имя сертификата должно быть уникальным для каждого криптошлюза.

- ключ `-GOST_R341012_256` задает использование алгоритма подписи – ГОСТ Р 34.10-2012 (ключ 256 бит).

На УЦ для поддержки алгоритма ГОСТ Р 34.10-2012 (ключ 256 бит) должно быть установлено СКЗИ «КриптоПро CSP» версии 5.0 или новее.

- ключ `-fb64` задает месторасположение и формат представления запроса на сертификат; будет использован формат представления BASE64 с сохранением файла запроса в корень USB Flash накопителя под именем `hub1.request`. Если данный параметр не указать, то текст запроса будет напечатан на консоль в формате BASE64.

Нажимайте предлагаемые клавиши на клавиатуре для инициализации БИО ДСЧ (если на криптошлюзе установлен аппаратный датчик случайных чисел, то нажимать предлагаемые клавиши не потребуется):

```
Progress: [********* ]
Press key: U
```

После завершения работы ДСЧ файл запроса будет сохранен в корне USB Flash накопителя:

```
administrator@sterragate] dir media:1E181F1C181EF28F/
 1 -rwx 473 Thu Jul  4 10:40:32 2019 hub1.request
```

Настоятельно рекомендуется сохранить контейнер закрытого ключа при помощи утилиты `cont_mgr`. Контейнер может понадобиться в случае восстановления криптошлюза при отказе HDD/SSD диска. Носитель с файлом контейнера нужно хранить в защищенном и недоступном для третьих лиц месте.

1.4. Отмонтируйте и извлеките USB Flash накопитель (посмотреть точки монтирования можно при помощи команды `run mount`):

```
administrator@sterragate] run umount /media/1E181F1C181EF28F
```

2. Выпуск сертификата криптошлюза на УЦ и импортирование сертификатов УЦ и криптошлюза в базу Продукта.

2.1. Доставьте файл запроса на УЦ и выпустите по нему сертификат криптошлюза.

2.2. Скопируйте выпущенный сертификат для криптошлюза под именем `hub1.cer` и сертификат УЦ под именем `ca.cer` в корень USB Flash накопителя.

2.3. Вновь вставьте USB Flash накопитель, содержащий файлы сертификатов, в свободный порт USB на криптошлюзе.

2.4. Убедитесь в наличии сертификатов на USB Flash накопителе:

```
administrator@sterragate] dir media:1E181F1C181EF28F/
 1 -rwx 592 Thu Jul  4 10:40:32 2019 ca.cer
 2 -rwx 804 Thu Jul  4 10:40:32 2019 hub1.cer
```

```
 3 -rwx 473 Thu Jul  4 10:40:32 2019 hub1.request
```

Сохраняйте сертификаты. Они могу понадобиться в случае восстановления криптошлюза при отказе HDD/SSD диска.

2.5. Импортируйте сертификат УЦ в базу Продукта:

```
administrator@sterragate] run cert_mgr import -f media:1E181F1C181EF28F/ca.cer -t
```

- ключ `-f` задает месторасположение файла сертификата;
- ключ `-t` используется для импортирования доверенного (trusted) сертификата УЦ.

2.6. Импортируйте сертификат криптошлюза в базу Продукта:

```
administrator@sterragate] run cert_mgr import -f media:1E181F1C181EF28F/hub1.cer
```

2.7. Убедитесь, что сертификаты успешно импортированы в базу Продукта:

```
administrator@sterragate] run cert_mgr show
```

```
Found 2 certificates. No CRLs found.
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
```

Примечание - Видно, что сертификат УЦ импортирован как **trusted** (доверенный), а сертификат криптошлюза как **local** (local означает, что для данного сертификата есть соответствующий закрытый ключ в ключевом контейнере).

2.8. Выполните проверку статуса сертификатов в базе Продукта:

```
administrator@sterragate] run cert_mgr check
```

```
1 State: Inactive C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
                  Certificate can not be verified.
2 State: Inactive C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
                  Certificate can not be verified.
```

Примечание - Видно, что все сертификаты имеют статус **Inactive** (неактивный) с пометкой: «Certificate can not be verified» (сертификат не может быть проверен). Причиной этому является включенный по умолчанию в консоли cisco-like механизм проверки списка отозванных сертификатов (далее COC или CRL). Так как в базе Продукта COC отсутствует, поэтому проверка не может быть осуществлена. Далее будет описан процесс настройки автоматической загрузки COC и импортирование его в базу Продукта. Загрузка COC осуществляется по протоколу HTTP с заданной периодичностью.

Настройка паролей доступа к консоли cisco-like

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

Пользователь и пароль по умолчанию `cscons`, `csp`. Обязательно смените пароль для пользователя `cscons`, так как под этим пользователем осуществляется доступ по SSH в cisco-like консоль. Также не забудьте сменить `enable` пароль.

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

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

2. Смените пароль для пользователя `cscons` и на `enable`:

```
sterragate#configure terminal
```

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

```
sterragate(config)#username cscons secret 0 ПАРОЛЬ
sterragate(config)#enable secret 0 ПАРОЛЬ
```

Настройка сетевых параметров

Изменение состояния интерфейсов и назначение IP адресов применяются сразу после ввода соответствующих команд. **Политика безопасности применяется после выхода из режима конфигурирования (сетевые маршруты относятся к политике безопасности).**

1. Задайте имя устройства:

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

2. Включите внешний `GigabitEthernet0/0` и внутренний `GigabitEthernet0/1` интерфейсы. В зависимости от параметров сетевого интерфейса его имя может отличаться от `GigabitEthernetN/M` и иметь имена, например `FastEthernetN/M`, `TenGigabitEthernetN/M`:

```
Hub1(config)#interface range GigabitEthernet 0/0 - 1
Hub1(config-if-range)#no shutdown
Hub1(config-if-range)#exit
```

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

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

```
GigabitEthernet0/0 is up, line protocol is up
  Hardware address is 0050.569e.b06d
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.d3c3
  MTU 1500 bytes
...
```

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

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

```
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 192.168.100.1 255.255.255.0
Hub1(config-if)#exit
```

5. Задайте маршрут по умолчанию через устройство Router1:

```
Hub1(config)#ip route 0.0.0.0 0.0.0.0 172.16.100.1
Hub1(config)#exit
```

6. Проверьте доступность устройства Router1:

```
Hub1#ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 100(128) bytes of data.
108 bytes from 172.16.100.1: icmp_seq=1 ttl=64 time=1.13 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.237 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.192 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.200 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.299 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4078ms
rtt min/avg/max/mdev = 0.192/0.411/1.131/0.362 ms
```

Настройка шифрования

1. Параметры IKE.

1.1. Укажите в качестве типа идентификатора (identity), используемого в рамках протокола IKE, отличительное имя (Distinguished Name, DN) x509 сертификата (глобальный параметр):

```
Hub1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Hub1(config)#crypto isakmp identity dn
```

По умолчанию отличительное имя будет взято из сертификата устройства, например, для Hub1 это «C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1».

1.2. Настройте параметры DPD (dead peer detection), срабатывающего по отсутствию входящего трафика в IPsec туннеле (в версии 5.0 можно настроить периодическую отправку пакетов DPD):

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

Примечание - Если в течение 3 секунд отсутствует входящий трафик в IPsec туннеле, **но присутствует исходящий,** то с интервалом в 2 секунды посылается 5 keepalive пакетов в рамках IKE туннеля, чтобы удостовериться в работоспособности туннеля. Если партнер не отвечает на keepalive пакеты, то соответствующий IKE туннель и связанные с ним IPsec туннели уничтожаются. В случае наличия исходящего защищаемого трафика происходит попытка создания новых IKE/IPsec туннелей.

1.3. Включите фрагментацию IKE пакетов (IKE пакеты будут фрагментироваться таким образом, чтобы их размер не превышал 576 байт):

```
Hub1(config)#crypto isakmp fragmentation
```

1.4. Включите случайный разброс времени жизни IKE и IPsec SA, чтобы снизить нагрузку на шлюз (позволяет избежать единовременное массовое пересоздание SA при обновлении ключей):

```
Hub1(config)#crypto isakmp security-association lifetime delta 50
```

1.5. Увеличьте допустимое количество одновременно инициируемых IKE сессий (не путать с общим количеством IKE сессий) для всех партнёров (значение по умолчанию 30):

```
Hub1(config)#crypto isakmp initiator-sessions-max 100
```

1.6. Увеличьте допустимое количество одновременных IKE обменов, проводимых шлюзом со всеми партнерами в качестве ответчика (не путать с общим количеством IKE сессий; значение по умолчанию 20):

```
Hub1(config)#crypto isakmp responder-sessions-max 100
```

1.7. Создайте политику **1**, описывающую алгоритмы 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
```

1.8. Создайте политику **2**, описывающую алгоритмы 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
```

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

```
Hub1(config)#crypto isakmp profile NEW_ALGS
```

1.9.1. Задайте, чтобы в качестве типа идентификатора (identity) партнера использовалось отличительное имя из сертификата (Distinguished Name, DN):

```
Hub1(conf-isa-prof)#match identity dn
```

1.9.2. Задайте, чтобы при поиске сертификата партнера использовался идентификатор (identity) из его сертификата (Distinguished Name, DN):

```
Hub1(conf-isa-prof)#set identity mapping
```

1.9.3. Задайте использование политики **1**, описывающей алгоритмы IKE туннеля для совместимости с версией 5.0:

```
Hub1(conf-isa-prof)#set policy 1
Hub1(conf-isa-prof)#exit
```

1.10. Создайте профиль настроек ISAKMP (алгоритм IKE) под именем `OLD_ALGS` для совместимости с версией 4.3, 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
```

1.10.1. Задайте использование политики **2**, описывающей алгоритмы IKE туннеля для совместимости с версией 4.3:

```
Hub1(conf-isa-prof)#set policy 2
Hub1(conf-isa-prof)#exit
```

2. Параметры IPsec.

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

```
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_GOST_ENCRYPT_AND_INTEGRITY esp-magma-
mgm
Hub1(cfg-crypto-trans)#exit
```

2.3. Создайте IPsec трансформ, задающий комбинированный алгоритм шифрования и имитозащиты на базе ГОСТ 28147-89 для трафика (для версий 4.3):

```
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 IPv4 трафика, который нужно защищать между центральным офисом и филиалом №1:

```
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 список доступа (ACL) для IPv4 трафика, который нужно защищать между центральным офисом и филиалом №2:

```
Hub1(config)#ip access-list extended IPSEC_ACl_HUB1_AND_SPOKE2
Hub1(config-ext-nacl)#permit ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

2.6. Создайте крипто-карту (имя `VPN`, раздел 1) для построения IPsec туннеля между центральным офисом и филиалом №1:

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

2.6.1. Укажите список доступа для защищаемого трафика:

```
Hub1(config-crypto-map)# match address IPSEC_ACl_HUB1_AND_SPOKE1
```

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

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

2.6.3. Укажите IPv4 адрес партнера по IPsec, в данном сценарии это внешний IPv4 адрес устройства Spoke1 (если криптошлюз находится за устройством, выполняющим NAT, то указывается его внешний адрес):

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

2.6.4. Прикрепите настройки ISAKMP (протокол IKE):

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

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

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

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

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

2.6.7. Включите автоматическую и интеллектуальную корректировку значения MSS (величина подбирается в зависимости от алгоритма шифрования и MTU трассы) для исходящего TCP трафика, чтобы его пакеты имели оптимальный размер и не фрагментировались криптошлюзом (данная настройка должна быть задача с двух сторон):

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

2.7. Создайте раздел 2 крипто-карты `VPN` для построения IPsec туннеля между центральным офисом и филиалом №2. Добавьте в него настройки для построения IPsec туннеля между центральным офисом и филиалом №2:

```
Hub1(config)#crypto map VPN 2 ipsec-isakmp
Hub1(config-crypto-map)# match address IPSEC_ACl_HUB1_AND_SPOKE2
Hub1(config-crypto-map)# set transform-set GOST_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)# set isakmp-profile OLD_ALGS
Hub1(config-crypto-map)# set security-association mss-fix out
Hub1(config-crypto-map)# set peer 172.16.2.2
Hub1(config-crypto-map)# set pre-fragmentation
Hub1(config-crypto-map)# set dead-connection history off
Hub1(config-crypto-map)#exit
```

2.8. Прикрепите созданную крипто-карту `VPN` к внешнему WAN интерфейсу `GigabitEthernet0/0`:

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

3. Примените настройки:

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

Настройка политики обработки COC (CRL)

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

1. Включите проверку COC:

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

Примечание - Если по обоснованным причинам использование COC невозможно, то выключите проверку COC и **не включайте автоматическую загрузку COC** (команда `crl download …` ниже):

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

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

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

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

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

4. Примените настройки:

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

5. Проверьте загружен ли COC в базу Продукта:

```
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
```

Примечания:

1. Если COC не загрузился, то проверьте файл журнала, например:

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

2. Чтобы не ждать следующего периода загрузки COC можно перезапустить сервис (getcrls) загрузки COC вручную:

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

3. Начиная с версии 5.0 COC можно удалить из базы Продукта:

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

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

6. Выполните проверку статуса сертификатов в базе Продукта:

```
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
```

Настройка межсетевого экрана

Начиная с версии 5.0 были существенно доработаны функции (улучшен функционал, повышены стабильность и производительность) межсетевого экранирования, осуществляющие фильтрацию с контролем состояния сессий (stateful для протоколов TCP и UDP). В связи с этим, с целью повышения безопасности, мы настоятельно рекомендуем использовать данный функционал.

В «С-Терра Шлюз» 5.0 межсетевой экран с контролем состояния сессий (stateful) реализован аналогично функционалу Context-Based Access Control (CBAC) в Cisco IOS, но имеет ряд преимуществ: помимо транзитного трафика также контролируется трафик, предназначенный для самого криптошлюза, то есть идущий в TCP/IP стек ОС.

Настройка межсетевого экрана осуществляются 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. Подробное описание работы данной команды приведено в сценарии «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv6)».

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

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

Примечание - Данный ACL разрешает входящие пакеты протоколов IKE, ESP, ICMP, а остальные блокирует только для случая, когда криптошлюз является ответчиком **впервые**. Если криптошлюз является инициатором или ответчиком в рамках ранее созданной сессии, то ответные пакеты будут разрешены, так как согласно алгоритму работы stateful фильтрации динамические разрешающие правила более приоритетны чем статические. Stateful фильтрация будет настроена далее в данном разделе. Динамические правила stateful фильтрации в cisco-like консоли НЕ отображаются, их можно посмотреть при помощи утилиты `fwconn_mgr` с параметром `show`.

3. Создайте правила инспектирования (stateful) всего TCP и UDP трафика и задайте необходимое максимальное количество сессий (в версии 5.0 данная команда устанавливает максимальное количество сессий для IPv4 и IPv6):

```
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
```

Максимальное количество сессий настраивается в зависимости от условий, потребностей и производительности аппаратной платформы. Суммарное максимальное рекомендуемое значение для количества сессий – 10 000 000 (для Gate 7000+). Также может потребоваться увеличить значение fw_storage_hash_bits в drv_mgr, более 22 ставить не рекомендуется.

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

```
Hub1(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)#end
```

5. Опционально: если необходимо включить аудит построения и завершения сессий (информация будет в `/var/log/cspvpngate.log`), то это можно сделать командой:

Включение аудита снижает производительность stateful фильтрации и существенно расходует дисковое пространство. Используйте аудит в случае, когда это оправдано.

```
Hub1(config)#ip inspect audit-trail
```

Пример работы аудита:

```
2024-01-11T14:11:01.961041+03:00 Hub1 vpnlogsvc: 00290014 TCP session created:
initiator 172.16.100.2:47778 -> responder 172.16.100.1:22
2024-01-11T14:11:32.757597+03:00 Hub1 vpnlogsvc: 00290001 TCP session closed: initiator
...
172.16.100.2:47778 sent 4592 bytes -> responder 172.16.100.1:22 sent 5800 bytes
```

```
2024-01-11T14:11:36.833870+03:00 Hub1 vpnlogsvc: 00290014 UDP session created:
initiator 192.168.100.100:40060 -> responder 10.0.0.16:53
2024-01-11T14:11:36.833921+03:00 Hub1 vpnlogsvc: 0029000f UDP session closed: initiator
192.168.100.100:40060 sent 47 bytes -> responder 10.0.0.16:53 sent 0 bytes
```

Настройка криптошлюза Hub1 завершена.

В Приложении представлены тексты конфигураций для криптошлюза Hub1:

- текст консоли cisco-like;
- текст LSP.

Настройка криптошлюза Spoke1

Настройка криптошлюза Spoke1 происходит аналогично настройке Hub1, за исключением настроек IKE/IPsec для совместимости с версией 4.3.

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст консоли cisco-like;
- текст LSP.

Настройка криптошлюза Spoke2

Настройка криптошлюза Spoke2 происходит аналогично настройке Spoke1, за исключением настройки межсетевого экрана, ISAKMP профайлов.

В Приложении представлены тексты конфигураций для криптошлюза Spoke2:

- текст консоли cisco-like;
- текст LSP.

Дополнительные возможности

Перешифрование в центре

Для настройки безопасного взаимодействия подсетей филиалов между собой с перешифрованием в центре требуется добавить трафик между филиалами в списки доступа.

1. Измените на криптошлюзе центрального офиса список доступа (ACL) для филиала №1. Добавьте IPv4 трафик от филиала №2 к филиалу №1, который нужно защищать:

```
Hub1(config)#ip access-list extended IPSEC_ACl_HUB1_AND_SPOKE1
Hub1(config-ext-nacl)#permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

2. Измените на криптошлюзе центрального офиса список доступа (ACL) для филиала №2. Добавьте IPv4 трафик от филиала №1 к филиалу №2, который нужно защищать:

```
Hub1(config)#ip access-list extended IPSEC_ACl_HUB1_AND_SPOKE2
Hub1(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

3. Измените на криптошлюзе филиала №1 список доступа (ACL). Добавьте IPv4 трафик от филиала №2 к филиалу №1, который нужно защищать:

```
Spoke1(config)#ip access-list extended IPSEC_ACl_HUB1_AND_SPOKE1
Spoke1(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Spoke1(config-ext-nacl)#exit
```

4. Измените на криптошлюзе филиала №2 список доступа (ACL). Добавьте IPv4 трафик от филиала №1 к филиалу №2, который нужно защищать:

```
Spoke2(config)#ip access-list extended IPSEC_ACl_HUB1_AND_SPOKE2
Spoke2(config-ext-nacl)#permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
Spoke2(config-ext-nacl)#exit
```

Проверка работоспособности стенда

Проверку работоспособности нужно выполнять после завершения настройки всех устройств стенда.

Проверка IP связности

1. Проверьте, что с криптошлюзов по ICMP доступен шлюз по умолчанию (Router1). Для этого выполните команду `ping` из cisco-like консоли соответствующего криптошлюза.

```
Hub1#ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 100(128) bytes of data.
108 bytes from 172.16.100.1: icmp_seq=1 ttl=64 time=0.918 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.250 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.306 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.225 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.238 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4081ms
rtt min/avg/max/mdev = 0.225/0.387/0.918/0.267 ms
```

```
Spoke1#ping 172.16.1.1
```

```
PING 172.16.1.1 (172.16.1.1) 100(128) bytes of data.
108 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.771 ms
108 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.282 ms
108 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.276 ms
108 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.268 ms
108 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.244 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4096ms
rtt min/avg/max/mdev = 0.244/0.368/0.771/0.202 ms
```

```
Spoke2#ping 172.16.2.1
```

```
PING 172.16.2.1 (172.16.2.1) 100(128) bytes of data.
108 bytes from 172.16.2.1: icmp_seq=1 ttl=64 time=1.42 ms
108 bytes from 172.16.2.1: icmp_seq=2 ttl=64 time=0.444 ms
108 bytes from 172.16.2.1: icmp_seq=3 ttl=64 time=0.376 ms
108 bytes from 172.16.2.1: icmp_seq=4 ttl=64 time=0.306 ms
108 bytes from 172.16.2.1: icmp_seq=5 ttl=64 time=0.313 ms

--- 172.16.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4070ms
rtt min/avg/max/mdev = 0.306/0.571/1.420/0.428 ms
```

Видно, что устройство Router1 доступно по ICMP со всех криптошлюзов.

2. Проверьте, что с защищаемых устройств Host-b-hub1, Host-b-spoke1, Host-b-spoke2 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=0.469 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.398 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.411 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=2.47 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=64 time=0.428 ms

--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.398/0.834/2.467/0.816 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=2.81 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.854 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.444 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.390 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=2.29 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.390/1.357/2.808/1.000 ms
```

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

```
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.296 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.326 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.320 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.289 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=0.345 ms

--- 192.168.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.289/0.315/0.345/0.020 ms
```

Видно, что со всех защищаемых устройств соответствующие криптошлюзы доступны по ICMP.

3. Проверьте, что с криптошлюза Hub1 доступны по ICMP криптошлюзы Spoke1 и Spoke2 (по внешним интерфейсам). Для этого выполните команду `ping` из cisco-like консоли соответствующего криптошлюза.

```
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=63 time=0.802 ms
108 bytes from 172.16.1.2: icmp_seq=2 ttl=63 time=0.416 ms
108 bytes from 172.16.1.2: icmp_seq=3 ttl=63 time=0.449 ms
108 bytes from 172.16.1.2: icmp_seq=4 ttl=63 time=0.393 ms
108 bytes from 172.16.1.2: icmp_seq=5 ttl=63 time=0.485 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.393/0.509/0.802/0.149 ms
```

```
Hub1#ping 172.16.2.2
```

```
PING 172.16.2.2 (172.16.2.2) 100(128) bytes of data.
108 bytes from 172.16.2.2: icmp_seq=1 ttl=63 time=1.23 ms
108 bytes from 172.16.2.2: icmp_seq=2 ttl=63 time=0.614 ms
108 bytes from 172.16.2.2: icmp_seq=3 ttl=63 time=0.692 ms
108 bytes from 172.16.2.2: icmp_seq=4 ttl=63 time=0.562 ms
108 bytes from 172.16.2.2: icmp_seq=5 ttl=63 time=3.23 ms

--- 172.16.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4055ms
rtt min/avg/max/mdev = 0.562/1.264/3.227/1.009 ms
```

Видно, что с криптошлюза Hub1 доступны по ICMP криптошлюзы Spoke1 и Spoke2.

Проверка PKI

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

```
Hub1#run cert_mgr show
```

```
Found 2 certificates. Found 1 CRL.
```

```
1 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
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
```

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

```
Spoke2#run cert_mgr show
```

```
Found 2 certificates. Found 1 CRL.
1 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Spoke2
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
```

Видно, что на криптошлюзах 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 статус всех сертификатов Active. Для этого выполните команду `run cert_mgr check` из cisco-like консоли соответствующего криптошлюза.

```
Hub1#run cert_mgr check
```

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

```
Spoke1#run cert_mgr check
```

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

```
Spoke2#run cert_mgr check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Spoke2
2 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

Видно, что на криптошлюзах статус всех сертификатов Active. Если статус Inactive, то проверьте:

- загружен ли СОС в базу Продукта;
- правильны и одинаковы ли дата и время на всех криптошлюзах.

Проверка построения защищенного соединения между подсетями Hub1 и Spoke1, Hub1 и Spoke2

1. Инициируйте построение защищенного соединения между криптошлюзами Hub1 и Spoke1 при помощи ICMP трафика, посылаемого с устройства Host-b-spoke1 на Host-b-hub1 (можно и наоборот). 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=884 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=0.721 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=0.880 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=0.819 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=0.781 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4068ms
rtt min/avg/max/mdev = 0.721/177.445/884.027/353.291 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 1 (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 1 (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 1 (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 1 (192.168.1.0-192.168.1.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn 440 440
```

3. Инициируйте построение защищенного соединения между криптошлюзами Hub1 и Spoke2 при помощи ICMP трафика, посылаемого с устройства Host-b-spoke2 на Host-b-hub1 (можно и наоборот). Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke2.

```
root@Host-b-spoke2:~# 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=884 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=0.721 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=0.880 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=0.819 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=0.781 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4068ms
rtt min/avg/max/mdev = 0.721/177.445/884.027/353.291 ms
```

4. Проверьте, что на криптошлюзах Hub1 и Spoke2 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 1 (172.16.100.2,500)-(172.16.1.2,500) active 1928 1960
2 2 (172.16.100.2,500)-(172.16.2.2,500) active 1916 1908

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 440 440
2 2 (192.168.100.0-192.168.100.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 440 440
```

```
Spoke2#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.2.2,500)-(172.16.100.2,500) active 1908 1916

IPsec connections:
```

```
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.2.0-192.168.2.255,*)-(192.168.100.0-192.168.100.255,*) * ESP tunn 440 440
```

Видно, что между криптошлюзами Hub1, Spoke1 и Hub1, Spoke2 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 работы межсетевого экрана

Для проверки работы межсетевого экрана выполните последовательно следующие команды из cisco-like консоли криптошлюзов и проанализируйте их вывод.

```
Hub1#run fwconn_mgr show
Hub1#run klogview -f FW,drop
```

Проверка построения защищенного соединения между подсетями Spoke1 и Spoke2

Проверку построения защищенного соединения между подсетями Spoke1 и Spoke2 следует проводить при настройке перешифрования в центре.

1. Инициируйте построение защищенного соединения между криптошлюзами Spoke1 и Spoke2 при помощи ICMP трафика, посылаемого с устройства Host-b-spoke1 на Host-b-spoke2 (можно и наоборот). Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke1.

```
root@Host-b-spoke1:~# ping 192.168.2.100 -c 5
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_seq=1 ttl=61 time=101 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=61 time=16.0 ms
64 bytes from 192.168.2.100: icmp_seq=3 ttl=61 time=14.2 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=61 time=2.93 ms
64 bytes from 192.168.2.100: icmp_seq=5 ttl=61 time=2.63 ms

--- 192.168.2.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.633/27.430/101.306/37.353 ms
```

2. Проверьте, что на криптошлюзе Hub1 установлено защищенное IPsec соединение и изменилось количество байт, переданных по защищенному IPsec соединению. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюза.

```
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 3 (172.16.100.2,500)-(172.16.1.2,500) active 1936 1968
2 4 (172.16.100.2,500)-(172.16.2.2,500) active 2068 1936

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 3 (192.168.2.0-192.168.2.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 880 880
2 4 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 880 880
```

Видно, что между криптошлюзами Hub1, Spoke1 и Hub1, Spoke2 установлено защищенное IPsec соединение и количество полученных и отправленных пакетов изменилось на счётчиках обоих IPsec соединений. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости, увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Приложение

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
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Hub1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp 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 GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
```

```
 remark WAN IP
 host 172.16.100.2
!
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 IPSEC_ACL_HUB1_AND_SPOKE2
 permit ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN 1 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.1.2
 set pre-fragmentation
 set dead-connection history off
crypto map VPN 2 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_SPOKE2
 set transform-set GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile OLD_ALGS
 set security-association mss-fix out
 set peer 172.16.2.2
 set pre-fragmentation
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.100.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.100.1 255.255.255.0
!
interface GigabitEthernet0/2
 no ip address
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 172.16.100.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
```

```
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Fri Aug 16 09:36:01 2024

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Fri Aug 16 09:36:01 2024 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 172.16.100.1
        )

)

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
)

IKETransform crypto:isakmp:policy:2
(
    CipherAlg   = "G2814789CPRO1-K256-CBC-65534"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
```

```
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

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

ESPProposal GOST_ENCRYPT_AND_INTEGRITY:ESP
(
    Transform* = ESPTransform
    (
        CipherAlg*          = "G2814789CPRO2-K288-CNTMAC-253"
        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 = 172.16.100.2
        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 = 172.16.100.2
        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
)

IdentityEntry Remote:OLD_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:OLD_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
)

AuthMethodGOSTSign GOST:Sign:OLD_ALGS
```

```
(
    RemoteID             = Remote:OLD_ALGS
    LocalID              = Local:Cert:OLD_ALGS
    SendRequestMode      = ALWAYS
    SendCertMode         = ALWAYS
)

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

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

IPsecAction IPsecAction:VPN:2
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.2.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( GOST_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:2
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
```

```
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        Label = "VPN:2:IPSEC_ACL_HUB1_AND_SPOKE2"
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.2.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2 >
        LogEventID = "IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_AND_SPOKE2"
    )
)

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

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

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

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
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Spoke1
```

```
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.1.2
!
ip access-list extended IPSEC_ACL_HUB1_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
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN 1 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_SPOKE1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set security-association mss-fix out
 set peer 172.16.100.2
 set 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 VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/2
 no ip address
```

```
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Fri Aug 16 09:33:40 2024

GlobalParameters(
    Title                       = "This LSP was automatically generated by CSP Converter
at Fri Aug 16 09:33:40 2024 (user: cscons)"
    Version                       = LSP_5_0
    CRLHandlingMode               = ENABLE
    PreserveIPsecSA               = FALSE
)

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

)

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

IKETransform crypto:isakmp:policy:1
(
    CipherAlg   = "GR341215M-K256-CFB-65526"
```

```
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

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

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

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

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
```

```
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_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_HUB1_AND_SPOKE1"
    )
)

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

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

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

Конфигурации криптошлюза Spoke2

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes 4294967295
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
```

```
username          cscons          privilege          15          secret          5
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Spoke2
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
crypto isakmp policy 1
 encr gost
 hash gost341112-256-tc26
 authentication gost-sig
 group vko2
!
crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
!
ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE2
 permit ip 192.168.2.0 0.0.0.255 192.168.100.0 0.0.0.255
!
!
crypto map VPN 1 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_SPOKE2
 set transform-set GOST_ENCRYPT_AND_INTEGRITY
 set peer 172.16.100.2
 set dead-connection history off
!
interface GigabitEthernet0/0
 ip address 172.16.2.2 255.255.255.0
 crypto map VPN
!
interface GigabitEthernet0/1
 ip address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/2
 no ip address
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 172.16.2.1
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

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

```
#   Conversion Date/Time:   Fri Aug 16 09:38:05 2024

GlobalParameters(
    Title                     = "This LSP was automatically generated by CSP Converter
at Fri Aug 16 09:38:05 2024 (user: cscons)"
    Version                     = LSP_4_3
    CRLHandlingMode             = ENABLE
    PreserveIPsecSA             = FALSE
)

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

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
)

IKETransform crypto:isakmp:policy:1
(
    CipherAlg   = "G2814789CPRO1-K256-CBC-65534"
    HashAlg     = "GR341112_256TC26-65128"
    GroupID     = VKO2_1B
    RestrictAuthenticationTo = GOST_SIGN
    LifetimeSeconds = 86400
)

ESPProposal GOST_ENCRYPT_AND_INTEGRITY:ESP
(
    Transform* = ESPTransform
    (
        CipherAlg*          = "G2814789CPRO2-K288-CNTMAC-253"
        LifetimeSeconds     = 3600
        LifetimeKilobytes   = 4294967295
    )
)

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

AuthMethodGOSTSign GOST:Sign
(
    LocalID        =  IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA )
    SendRequestMode    =  ALWAYS
    SendCertMode       =  ALWAYS
)
```

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

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
        DFHandling=COPY
        Assemble=TRUE
    )
    ContainedProposals = ( GOST_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    IKERule = IKERule:VPN:1
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.100.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE2"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    IPsecPolicy = IPsecPolicy:VPN
)
```