Организация защищенного взаимодействия между подсетями центрального офиса и филиала посредством IPsec VPN туннеля (протокол IPv4) с использованием «С-Терра Шлюз КВ»

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 устройств «С-Терра Шлюз КВ» версии 5.0 для обеспечения безопасного взаимодействия между защищаемыми подсетями центрального офиса и филиала. Защищаемые подсети в LAN сегменте и недоверенные подсети в WAN сегменте – IPv4. Обеспечение безопасного взаимодействия достигается путем шифрования и туннелирования трафика с применением отечественных отраслевых стандартов ГОСТ и протокола IPsec.

Партнеры по IPsec, в настоящем документе – криптошлюзы, для аутентификации будут использовать сертификаты в формате x509. В качестве криптопровайдера будет использована криптографическая библиотека, разработанная компанией «С-Терра СиЭсПи».

На криптошлюзах будет настроено межсетевое экранирование с применением статических списков доступа (ACL) и пакетной фильтрации с контролем состояния сессий (stateful, аналог CBAC в Cisco IOS) для протоколов TCP и UDP.

Предварительные требования

Требования к материально-техническому обеспечению

В данном сценарии потребуются несколько USB Flash накопителей. Два накопителя будут использоваться в качестве рабочих ключевых носителей, один накопитель будет использоваться в качестве административного носителя, еще один – для переноса запросов и сертификатов между криптошлюзами и центром выпуска сертификатов.

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 КВ».
      1.1.1. Устройства «С-Терра Шлюз КВ» должны быть в заводском состоянии.
   1.2. Центр выпуска сертификатов.
      1.2.1. Должен быть настроен центр выпуска сертификатов (удостоверяющий центр, далее УЦ) для IPsec. Устройство с именем Certification_authority на схеме.
      1.2.2. Для выпуска цифровых сертификатов допускается использование встроенного в ОС Windows Server 2016 (или новее) удостоверяющего центра совместно с сертифицированным СКЗИ «КриптоПро» CSP 5.0 R2 (или новее).
      1.2.3. Для тестовых целей можно использовать тестовый УЦ от «КриптоПро» (веб-интерфейс: https://testgost2012.cryptopro.ru/certsrv/certrqxt.asp).

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

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

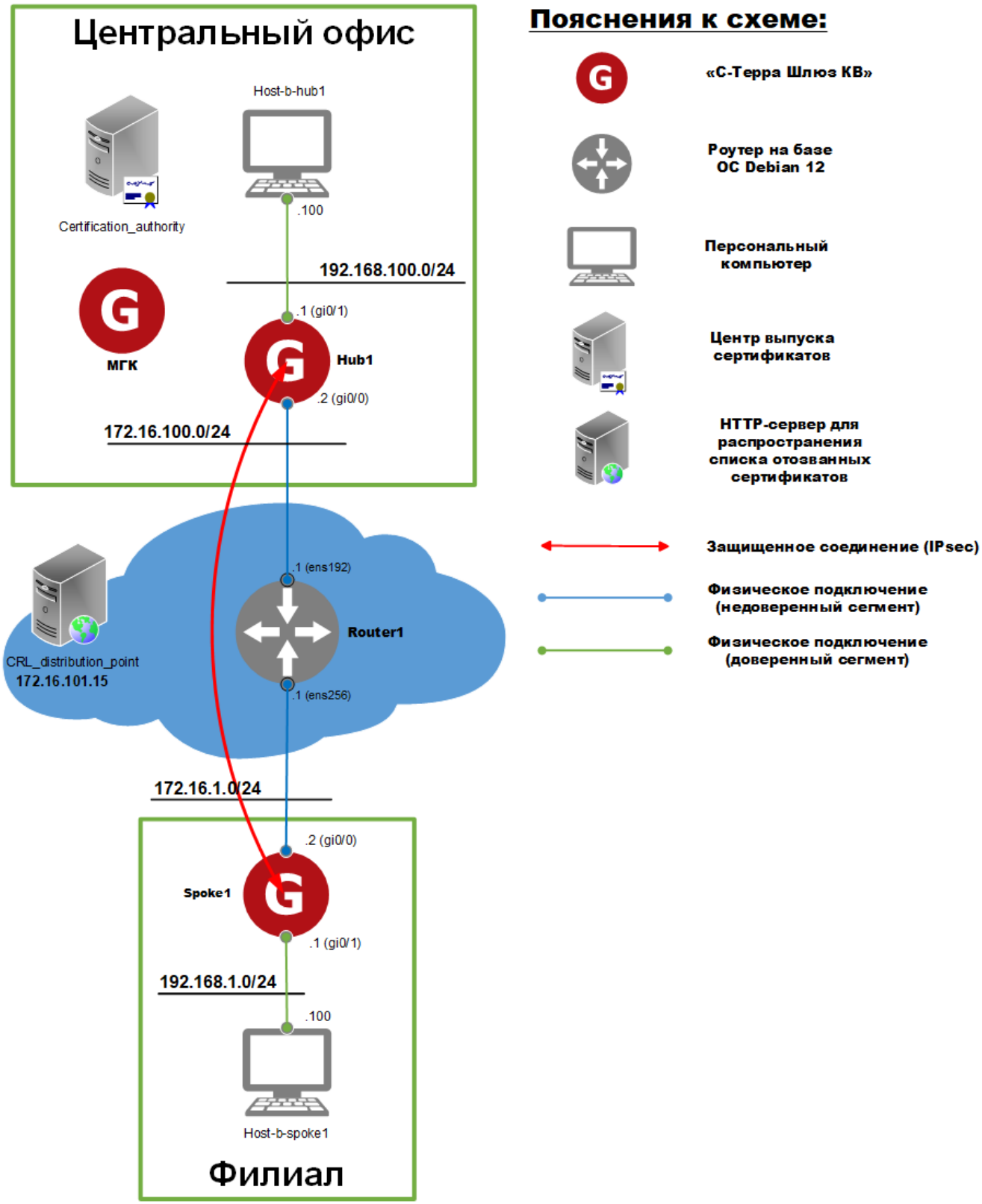


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Устройства «С-Терра Шлюз КВ».

   1.1. Криптошлюз – в данном сценарии так будет называться «С-Терра Шлюз КВ» («С-Терра Шлюз NF КВ», исполнение «ЗМВ»), инициализированный в роли network и использующийся для защиты сетевого трафика (устройства Hub1 и Spoke1 на схеме).

   1.2. МГК (модуль генерации ключей) – в данном сценарии так будет называться «С-Терра Шлюз КВ» («С-Терра МГК КВ», исполнение «МГК»), инициализированный в роли management и использующийся для подготовки ключевых носителей для криптошлюзов (устройство МГК на схеме).

2. Ключевые носители.

   В данном сценарии используются следующие типы ключевых носителей:

   2.1. Шифроблокнот – CD-диск однократной записи, содержащий случайную последовательность. Шифроблокнот используется устройством МГК при подготовке рабочих ключевых носителей.

   2.2. Административный носитель – USB Flash накопитель, использующийся на МГК и содержащий ключевую и конфигурационную информацию.

   2.3. Рабочий ключевой носитель (РКН) – USB Flash накопитель, содержащий доверенный сертификат, контейнер с закрытым ключом и состояние ПДСЧ. Рабочий носитель используются при введении криптошлюза в эксплуатацию и при запуске СКЗИ.

3. Администраторские роли:

   3.1. Администратор безопасности – это администратор устройства МГК.

   3.2. Администратор сети – это администратор криптошлюза.

4. Размещение устройств.

   4.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), устройства «С-Терра Шлюз КВ» (МГК и Hub1) и персональный компьютер (Host-b-hub1).

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

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

5. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизатор Router1.

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

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

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

   5.3. Устройство МГК к сети не подключается.

6. Параметры шифрования IPsec.

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

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

6.1. Параметры протокола IKEv2:

- Аутентификация при помощи цифровых сертификатов, алгоритм подписи – ГОСТ Р 34.10-2012 (ключ 256 бит);
- Алгоритм шифрования – ГОСТ Р 34.12-2015 «Магма» в режиме MGM (ключ 256 бит);
- Алгоритм псевдослучайной функции – ГОСТ Р 34.11-2012 «Стрибог» (ключ 512 бит);
- Алгоритм выработки общего ключа (аналог алгоритма Диффи-Хеллмана) – группа 33 (ГОСТ Р 34.10-2012-256-paramSetA).

6.2. Параметры протокола ESP:

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

7. Параметры межсетевого экранирования.

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

7.1. На LAN и WAN интерфейсах IPv6 трафик заблокирован.

7.2. На LAN интерфейсах IPv4 трафик не контролируется.

7.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

7.3.1. Протокол UDP порты 500 и 4500 – для IKEv2 и NAT-Traversal (ESP in UDP).

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

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

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

7.3.5. На WAN интерфейсах инспектируются все исходящие локальные и транзитные IPv4 соединения для протоколов TCP и UDP. На основе результатов инспектирования осуществляется фильтрация входящего трафика с учетом состояния сессии (stateful).

Настройка стенда

Настройка устройства Router1

1. Настройте IP адрес – 172.16.100.1 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 172.16.1.1 и маску – 255.255.255.0 на сетевом интерфейсе ens256.
3. 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 МГК

Настройка будет происходить локально при помощи консольного подключения. Все действия на устройстве МГК выполняются Администратором безопасности.

Начальные настройки

1. После загрузки ОС войдите в консоль управления под учетной записью администратора. Для этого, после появления сообщения

   ```
   Astra Linux 1.7.6 sterragate ttyS0
   ```

   введите логин, пароль и уровень целостности:

   Пользователь и пароль по умолчанию: `admin, s-terra`. Уровень целостности – 63 (высокий уровень целостности, другие значения не поддерживаются). Обязательно смените пароль для пользователя `admin` при помощи команды `change user password` или в процессе прохождения процедуры инициализации.

   ```
   sterragate login: admin
   Password:
   Integrity level: 63
   ```

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

   ```
   sterragate] terminal terminal-type xterm
   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 (пароль по умолчанию: `s-terra`).

   ```
   sterragate] system
   ```

   ```
   Entering system shell...
   Password:
   ```

   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 (это можно сделать в том числе в процессе прохождения процедуры инициализации):

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

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

5. Выйдите из Linux bash:

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

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

Начальные настройки завершены.

Инициализация и создание административного носителя

К устройству МГК подключите:

- USB Flash накопитель, который будет использоваться в качестве административного носителя;
- CD-ROM с шифроблокнотом.

1. Инициализируйте устройство МГК в роли management. Для этого выполните команду `initialize role_management`:

```
sterragate] initialize role_management
```

```
[init] Snippet "prepare_breach_sensor.sh" start
INFO: Starting breach sensor service initialization.
INFO: Configuration file for CTK-120 hardware platform has been detected.
INFO: Breach sensor service will be enabled.
INFO: Activating breach sensor service.
Created symlink /etc/systemd/system/default.target.wants/breachsensorsvc.service →
/lib/systemd/system/breachsensorsvc.service.
INFO: Breach sensor service activated.
INFO: Breach sensor initialization finished.
[init] Snippet "prepare_breach_sensor.sh" end
[init] Snippet "ssh_warning.sh" start
Not running using an SSH connection. No action needed.
[init] Snippet "ssh_warning.sh" end
[init] Snippet "prepare_force_initialize.sh" start
INFO: --force-reinit option is not provided. No action needed
[init] Snippet "prepare_force_initialize.sh" end
[init] Snippet "license_req.sh" start
INFO: You have to enter license for S-Terra Gate
Available product codes:
GATE
MVPN
```

1.1. Введите лицензионную информацию:

```
Enter product code: GATE
Enter customer code: XXXX
Enter license number: X
Enter license code: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

```
Your license:
  CustomerCode = XXXX
  ProductCode  = GATE
  LicenseNumber= X
  LicenseCode  = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
Is the above data correct? y
INFO: License accepted
[init] Snippet "license_req.sh" end
```

1.2. Из таблицы с доступными разделами выберите раздел USB Flash накопителя, который будет использоваться в качестве административного носителя (в данном примере – `/dev/sdb1`) и отформатируйте его:

> Все данные на выбранном разделе будут стерты!

```
[init] Snippet "station_role_management_actions.sh" start
INFO: Checking that the RND disk is mounted at /var/s-terra/media/st_rng
INFO: RND disk is found
INFO: Trying to find an existing or create a new administrative drive
[usb_helper] INFO: Table of available partitions
+----+-----------+---------------------------------+---------+----------+--------+--------+----------+
|    | path      | mountpoint                      | label   | fstype   | type   | size   | fsuse%   |
+====+===========+=================================+=========+==========+========+========+==========+
|  0 | /dev/sdb1 | /media/sdb1-usb-USB_SanDisk_3.2G |         | ntfs     | part   | 14.4G  | 0%       |
+----+-----------+---------------------------------+---------+----------+--------+--------+----------+
[usb_helper] INFO: Pick a partition
[usb_helper] INFO: 0) /dev/sdb1
```

```
[usb_helper] Select one of the options: 0
```

```
[usb_helper] INFO: Unmounting partition /dev/sdb1
[usb_helper] INFO: Preparing to format "/dev/sdb1" partition
[usb_helper] INFO: All data on partition "/dev/sdb1" will be deleted.
```

```
[usb_helper] Press enter to proceed (Ctrl+C to cancel)
```

```
[usb_helper] INFO: Formatting partition "/dev/sdb1" to "vfat" file system with label
"st_mgmt"...
[usb_helper] INFO: Done
```

1.3. Задайте ***пароль данных*** (MGK Password, этим паролем защищается ключевая информация на административном носителе):

```
[usb_helper] Input MGK Password:
[usb_helper] Repeat MGK Password:
```

```
[usb_helper] INFO: Creating identity file on this partition...
[usb_helper] INFO: Succesfully generated identity file with ID "605199" on this
partition
[usb_helper] INFO: Waiting for kernel to update partitions data...
[usb_helper] INFO: Done
[usb_helper] INFO: Calling sync command
[usb_helper] INFO: Done
[usb_helper] INFO: Sending singal for system to re-add partition...
[usb_helper] INFO: Done
[usb_helper] INFO: Waiting until system is aware of new partition...
[usb_helper] INFO: Done
[usb_helper] INFO: Current partitions state:
+----+-----------+--------------+---------+----------+--------+--------+----------+
|    | path      | mountpoint   | label   | fstype   | type   | size   | fsuse%   |
+====+===========+==============+=========+==========+========+========+==========+
|  0 | /dev/sdb1 |              | st_mgmt | vfat     | part   | 14.4G  |          |
+----+-----------+--------------+---------+----------+--------+--------+----------+
[usb_helper] INFO: All operations are completed successfully
INFO: Activating machine checking service for management role
Created                                                                    symlink
/etc/systemd/system/timers.target.wants/machine_check@management.timer              →
/lib/systemd/system/machine_check@.timer.
Created symlink /etc/systemd/system/machine_check_network.service → /dev/null.
INFO: Successfully activated machine checking service for management role
[init] Snippet "station_role_management_actions.sh" end
[init] Snippet "rng_init.sh" start
INFO: Initializing RNG: (press requested keys or Ctrl+C to interrupt)

Already initialized RNG.
INFO: RNG is initialized successfully
[init] Snippet "rng_init.sh" end
```

1.4. Установите надежные пароли для пользователей admin, user и root:

```
[init] Snippet "change_passwd_management.sh" start
```

```
Do you want to change the password for "admin" user? [Yes]
New password:
Retype new password:
```

```
passwd: password updated successfully
```

```
Do you want to change the password for "user" user? [Yes]
New password:
Retype new password:
```

```
passwd: password updated successfully
```

```
Do you want to change the password for "root" user? [Yes]
New password:
Retype new password:
```

```
passwd: password updated successfully
```

```
[init] Snippet "change_passwd_management.sh" end
[init] Snippet "set_initialized_flag_management.sh" start
[init] Snippet "set_initialized_flag_management.sh" end
[init] Snippet "remove_vpndrvr_conf_management.sh" start
INFO: Management interface is eth0
NOTICE: The policy applied to the management interface will be effective only after OS
is rebooted
[init] Snippet "remove_vpndrvr_conf_management.sh" end
INFO [init]: Initialization completed
INFO [init]: Some settings will take effect after OS reboot only.
```

1.5. Процедура инициализации устройства МГК завершена, перезагрузите устройство:

```
sterragate] reboot
```

Создание рабочих ключевых носителей

В данном сценарии потребуются два рабочих ключевых носителя (по одному для каждого из криптошлюзов Hub1 и Spoke1).

К устройству МГК подключите:

- Административный носитель;
- USB Flash накопитель, который будет использоваться в качестве рабочего ключевого носителя.

2. Войдите в консоль управления под учетной записью администратора:

```
Astra Linux 1.7.6 sterragate ttyS0
```

```
sterragate login: admin
Password:
Integrity level: 63
```

```
S-Terra Gate 5.0.26713 (amd64)
S-Terra administrative console

sterragate]
```

3. Выполните команду `login` и введите ***пароль данных*** (MGK Password) для получения доступа к административному носителю:

```
sterragate] login
MGK Password:
```

```
RNG history backup on the management storage is absent. It will be created afte
r logout
User is logged in
sterragate(logged in)]
```

4. Выполните команду `create network storage` для запуска процесса создания рабочего ключевого носителя:

```
sterragate(logged in)] create network storage
```

4.1. Из таблицы с доступными разделами выберите раздел USB Flash накопителя, который будет использоваться в качестве рабочего носителя (в данном примере – `/dev/sdc1`) и отформатируйте его:

Все данные на выбранном разделе будут стерты!

```
[usb_helper] INFO: Table of available partitions
+----+-----------+---------------------------------+---------+----------+--------+--------+----------+
|    | path      | mountpoint                      | label   | fstype   | type   | size   | fsuse%   |
+====+===========+=================================+=========+==========+========+========+==========+
|  0 | /dev/sdb1 | /var/s-terra/media/st_mgmt      | st_mgmt | vfat     | part   | 14.4G  | 0%       |
+----+-----------+---------------------------------+---------+----------+--------+--------+----------+
|  1 | /dev/sdc1 | /media/sdc1-usb-Kingston_DataTra |         | ntfs     | part   | 28.9G  | 0%       |
+----+-----------+---------------------------------+---------+----------+--------+--------+----------+
[usb_helper] INFO: Pick a partition
```

```
[usb_helper] INFO: 0) /dev/sdb1
[usb_helper] INFO: 1) /dev/sdc1
```

```
[usb_helper] Select one of the options: 1
```

```
[usb_helper] INFO: Unmounting partition /dev/sdc1
[usb_helper] INFO: Preparing to format "/dev/sdc1" partition
[usb_helper] INFO: All data on partition "/dev/sdc1" will be deleted.
```

```
[usb_helper] Press enter to proceed (Ctrl+C to cancel)
```

```
[usb_helper] INFO: Formatting partition "/dev/sdc1" to "vfat" file system with label
"st_network"...
[usb_helper] INFO: Done
[usb_helper] INFO: Creating identity file on this partition...
[usb_helper] INFO: Succesfully generated identity file with ID "435883" on this
partition
[usb_helper] INFO: Waiting for kernel to update partitions data...
[usb_helper] INFO: Done
[usb_helper] INFO: Calling sync command
[usb_helper] INFO: Done
[usb_helper] INFO: Sending singal for system to re-add partition...
[usb_helper] INFO: Done
[usb_helper] INFO: Waiting until system is aware of new partition...
[usb_helper] INFO: Done
[usb_helper] INFO: Current partitions state:
+----+-----------+---------------------------+------------+----------+--------+--------+----------+
|    | path      | mountpoint                | label      | fstype   | type   | size   | fsuse%   |
+====+===========+===========================+============+==========+========+========+==========+
|  0 | /dev/sdb1 | /var/s-terra/media/st_mgmt | st_mgmt    | vfat     | part   | 14.4G  | 0%       |
+----+-----------+---------------------------+------------+----------+--------+--------+----------+
|  1 | /dev/sdc1 |                           | st_network | vfat     | part   | 28.9G  |          |
+----+-----------+---------------------------+------------+----------+--------+--------+----------+
[usb_helper] INFO: All operations are completed successfully
```

4.2. Создание рабочего ключевого носителя завершено.

5. Отключите от устройства МГК созданный рабочий носитель, подключите другой USB Flash накопитель и на его основе создайте еще один рабочий носитель с помощью команды `create network storage`.

Создание сессионных данных

К устройству МГК подключите:

- CD-ROM с шифроблокнотом;
- Административный носитель;
- USB Flash накопитель с сертификатом УЦ.

Добавление сертификата УЦ

1. Определите имя (идентификатор) USB Flash накопителя с сертификатом УЦ:

```
sterragate(logged in)] dir media:
```

```
    1  drwx         4096  Fri Jan 16 13:55:31 2026  sdc1-usb-_USB_DISK_3.0_90
```

```
sterragate(logged in)] dir media:sdc1-usb-_USB_DISK_3.0_90/
```

```
    1  -rwx          555  Tue Sep  9 17:38:37 2025  CA.cer
```

В данном примере видно, что файл сертификата (`CA.cer`) лежит в корне USB Flash накопителя с идентификатором `sdc1-usb-_USB_DISK_3.0_90`.

2. Скопируйте сертификат УЦ в каталог доверенных сертификатов на административном носителе:

```
sterragate(logged in)] copy media:sdc1-usb-_USB_DISK_3.0_90/CA.cer trusted-ca sterra-
root-ca
```

```
Trusted CA copied successfully
```

где `sterra-root-ca` – имя, под которым сертификат УЦ будет находиться на административном носителе (в директории с именем `CA`).

3. Посмотреть список доверенных сертификатов на административном носителе можно с помощью команды `show trusted-ca`:

```
sterragate(logged in)] show trusted-ca
```

```
Trusted CA sterra-root-ca:
Certificates/CRLs from file /var/s-terra/media/st_mgmt/CA/sterra-root-ca.ca
Found 1 certificate. No CRLs found.
1 C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

4. USB Flash накопитель с сертификатом УЦ теперь можно отключить.

Создание сессионных данных криптошлюза Hub1

1. Создайте набор данных, предназначенный для криптошлюза Hub1:

```
sterragate(logged in)] create gate Hub1
```

```
sterragate(config-gate)#
```

1.1. Укажите доверенный сертификат:

```
sterragate(config-gate)#trusted-ca sterra-root-ca
```

1.2. Создайте контейнер, который будет использоваться в качестве контейнера с закрытым ключом для локального сертификата на криптошлюзе Hub1 (поддерживаются ключи длиной 256 и 512 бит):

```
sterragate(config-gate)#new-container Hub1_256 key-length 256
```

где `Hub1_256` – имя контейнера.

1.3. Выполните команду `exit` и подтвердите создание набора данных для криптошлюза Hub1:

```
sterragate(config-gate)#exit
```

```
Trusted CAs:
    sterra-root-ca

New containers:
    Hub1_256 key length: 256

% Create gate with entered parameters
    'Yes'  - acknowledge gate creation;
    'No'   - undo gate creation, erase all entered parameters
    Ctrl+C - continue enter parameters

    (you can simply enter first letter) ? Yes
```

При этом на административном носителе в директории `Gates` создается новая директория с именем `Hub1`, в которую сохраняются указанные данные. Посмотреть сохраненные данные для криптошлюза Hub1 на административном носителе можно с помощью команды `show gate Hub1`, внести изменения в существующий набор данных для Hub1 можно с помощью команды `modify gate Hub1`.

Создание сессионных данных криптошлюза Spoke1

Аналогичным образом, как это было сделано для Hub1, создайте набор данных для криптошлюза Spoke1.

Запись сессионных данных на рабочие ключевые носители

К устройству МГК подключите:

- CD-ROM с шифроблокнотом;
- Административный носитель;
- Рабочий ключевой носитель.

1. Выполните команду `pack Hub1` для записи сессионных данных криптошлюза Hub1 на рабочий носитель (опционально: в команде можно указать id, который задает имя директории на рабочем носителе. Если id не задавать, то имя по умолчанию – 0000):

```
sterragate(logged in)] pack Hub1 id Hub1
```

2. Задайте ***транспортный пароль*** (Transport Password, этим паролем защищается ключевая информация на рабочем носителе):

```
New transport password:
Re-type new password:
```

```
File compression...

containers\
  Hub1_256.s50
certs\
  trusted\
    sterra-root-ca.ca
pskeys\

...Done
```

3. Отключите рабочий носитель с данными для криптошлюза Hub1, подключите другой рабочий носитель и аналогичным образом запишите на него данные для криптошлюза Spoke1.

Настройка устройства МГК завершена.

Настройка криптошлюза Hub1

Настройка будет происходить локально при помощи консольного подключения. Все действия на криптошлюзе Hub1 выполняются Администратором сети.

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. После загрузки ОС войдите в консоль управления под учетной записью администратора. Для этого, после появления сообщения

   ```
   Astra Linux 1.7.6 sterragate ttyS0
   ```

   введите логин, пароль и уровень целостности:

   Пользователь и пароль по умолчанию: `admin, s-terra`. Уровень целостности – 63 (высокий уровень целостности, другие значения не поддерживаются). Обязательно смените пароль для пользователя `admin` при помощи команды `change user password` или в процессе прохождения процедуры инициализации.

   ```
   sterragate login: admin
   Password:
   Integrity level: 63
   ```

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

   ```
   sterragate] terminal terminal-type xterm
   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 (пароль по умолчанию: `s-terra`).

   ```
   sterragate] system
   ```

   ```
   Entering system shell...
   Password:
   ```

   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 (это можно сделать в том числе в процессе прохождения процедуры инициализации):

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

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

5. Выйдите из linux bash:

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

   ```
   logout
   Leaving system shell...
   ```

```
sterragate]
```

Начальные настройки завершены.

Инициализация

1. Подключите рабочий ключевой носитель с сессионными данными для криптошлюза Hub1.
2. Инициализируйте криптошлюз Hub1 в роли network. Для этого выполните команду `initialize role_network`:

```
sterragate] initialize role_network
```

```
[init] Snippet "prepare_breach_sensor.sh" start
INFO: Starting breach sensor service initialization.
INFO: Configuration file for CTK-120 hardware platform has been detected.
INFO: Breach sensor service will be enabled.
INFO: Activating breach sensor service.
Created symlink /etc/systemd/system/default.target.wants/breachsensorsvc.service →
/lib/systemd/system/breachsensorsvc.service.
INFO: Breach sensor service activated.
INFO: Breach sensor initialization finished.
[init] Snippet "prepare_breach_sensor.sh" end
[init] Snippet "ssh_warning.sh" start
Not running using an SSH connection. No action needed.
[init] Snippet "ssh_warning.sh" end
[init] Snippet "prepare_force_initialize.sh" start
INFO: --force-reinit option is not provided. No action needed
[init] Snippet "prepare_force_initialize.sh" end
[init] Snippet "license_req.sh" start
INFO: You have to enter license for S-Terra Gate
Available product codes:
GATE
MVPN
```

2.1. Введите лицензионную информацию:

```
Enter product code: GATE
Enter customer code: XXXX
Enter license number: X
Enter license code: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

```
Your license:
  CustomerCode = XXXX
  ProductCode  = GATE
  LicenseNumber= X
  LicenseCode  = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

```
Is the above data correct? Y
```

```
INFO: License accepted
[init] Snippet "license_req.sh" end
[init] Snippet "ssh_action.sh" start
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
INFO: SSH server was successfully enabled and started
[init] Snippet "ssh_action.sh" end
[init] Snippet "station_role_network_actions.sh" start
INFO: Checking that the special device is mounted at /var/s-terra/media/st_network
Info [mount-helper]: Successfully mounted device with label "st_network"
INFO: Successfully found device mounted at /var/s-terra/media/st_network
INFO: Possible gate-id:
Hub1
```

2.2. Укажите id (значение, которое было задано при записи сессионных данных на рабочий носитель на устройстве МГК):

```
Enter gate-id [Hub1]:
```

```
INFO: Activating machine checking service for network role
Created symlink /etc/systemd/system/timers.target.wants/machine_check@network.timer →
/lib/systemd/system/machine_check@.timer.
Created symlink /etc/systemd/system/machine_check_management.service → /dev/null.
INFO: Successfully activated machine checking service for network role
[init] Snippet "station_role_network_actions.sh" end
[init] Snippet "set_internet.sh" start
```

2.3. На вопрос о включении маршрутизации IPv6 трафика ответьте отрицательно:

```
Would you like to enable IPv6 forwarding? [No] No
```

```
INFO: IP forwarding status:
INFO: IPv4: enabled
INFO: IPv6: disabled.
[init] Snippet "set_internet.sh" end
[init] Snippet "run_reset_settings.sh" start
INFO: Reset product settings was completed successfully
[init] Snippet "run_reset_settings.sh" end
[init] Snippet "netifcfg_enumap.sh" start
INFO: Correct network interface mapping is found. No action required.
[init] Snippet "netifcfg_enumap.sh" end
[init] Snippet "check_policy.sh" start
NOTICE:
Network traffic is blocked, except on management interface (if using one).
To unblock network traffic, please setup the network security policy.

[init] Snippet "check_policy.sh" end
[init] Snippet "change_passwd_network.sh" start
```

2.4. Установите надежные пароли для пользователей admin, user и root:

```
Do you want to change the password for "admin" user? [Yes]
```

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

```
Do you want to change the password for "user" user? [Yes]
```

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

```
Do you want to change the password for "root" user? [Yes]
```

```
New password:
Retype new password:
passwd: password updated successfully
[init] Snippet "change_passwd_network.sh" end
[init] Snippet "set_initialized_flag_network.sh" start
[init] Snippet "set_initialized_flag_network.sh" end
[init] Snippet "remove_vpndrvr_conf_network.sh" start
INFO: Management interface is GigabitEthernet0/0 (eth0)
NOTICE: The policy applied to the management interface will be effective only after OS
is rebooted
[init] Snippet "remove_vpndrvr_conf_network.sh" end
INFO [init]: Initialization completed
INFO [init]: Some settings will take effect after OS reboot only.
```

2.5. Процедура инициализации криптошлюза Hub1 завершена, перезагрузите устройство:

```
sterragate] reboot
```

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 в соответствии с правилами пользования СКЗИ.

Запрос на сертификат криптошлюза будет сгенерирован при помощи утилиты `cert_mgr`, при этом будет использован созданный ранее контейнер с закрытым ключом.

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 консоли управления.

При выполнении сторонних команд в консоли управления и в консоли cisco-like перед командой нужно указывать ключевое слово `run`.

1. Войдите в консоль управления под учетной записью администратора:

```
sterragate login: admin
Password:
Integrity level: 63
```

2. Выполните команду `deploy` с указанием id криптошлюза (в данном примере – Hub1) и введите ***транспортный пароль*** (Transport Password) (при этом выполняется загрузка данных с рабочего носителя в оперативную память и запуск VPN подсистемы криптошлюза):

```
sterragate] deploy Hub1
Transport password:
Your configuration was deployed successfully
```

3. Убедитесь, что контейнер с закрытым ключом был успешно импортирован. Для этого выполните команду `run cont_mgr show`:

```
sterragate] run cont_mgr show
system::ram_s50://Hub1_256
```

Видно, что на криптошлюзе присутствует контейнер с именем `Hub1_256`, префикс `ram_s50` указывает на то, что это контейнер в формате s50, хранящийся в оперативной памяти криптошлюза.

4. Убедитесь, что доверенный сертификат УЦ был успешно импортирован в базу криптошлюза. Для этого выполните команду `run cert_mgr show`:

```
sterragate] run cert_mgr show
Found 1 certificate. No CRLs found.
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=S-Terra CSP Test Root CA
```

Видно, что на криптошлюзе присутствует доверенный (trusted) сертификат УЦ.

5. Генерация запроса на сертификат криптошлюза.

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

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

```
sterragate] dir media:
1  drwx          4096  Mon Jan 19 11:52:41 2026  sdc1-usb-_USB_DISK_3.0_90
```

Имя (идентификатор) USB Flash накопителя - **sdc1-usb-_USB_DISK_3.0_90**.

5.3. Запустите процесс генерации запроса на сертификат криптошлюза с сохранением файла запроса на USB Flash накопителе:

```
administrator@sterragate] run cert_mgr create -subj "C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1" -use_exist_cont -kc ram_s50://Hub1_256 -fb64 media:sdc1-usb-_USB_DISK_3.0_90/Hub1.request
```

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

Отличительное имя сертификата должно быть уникальным для каждого криптошлюза.

- ключ `-use_exist_cont` указывает, что будет использован существующий контейнер;
- ключ `-kc` содержит имя контейнера.
- 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.

Файл запроса будет сохранен в корне USB Flash накопителя:

```
sterragate] dir media:sdc1-usb-_USB_DISK_3.0_90/
```

```
    1  -rwx           530  Mon Jan 19 17:31:08 2026  Hub1.request
```

5.4. Для безопасного извлечения USB Flash накопителя перейдите в Linux bash и отмонтируйте его с помощью команды `umount` (посмотреть точки монтирования можно при помощи команды `mount`):

```
sterragate] system
```

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

```
Password:
root@sterragate:~# umount /media/sdc1-usb-_USB_DISK_3.0_90
```

5.5. Выйдите из Linux bash:

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

```
logout
Leaving system shell...
```

6. Выпуск сертификата криптошлюза на УЦ и импортирование сертификата в базу Продукта.

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

6.2. Скопируйте выпущенный сертификат для криптошлюза под именем `Hub1.cer` в корень USB Flash накопителя.

6.3. Подключите USB Flash накопитель с файлом сертификата в свободный порт USB на криптошлюзе.

6.4. Убедитесь в наличии сертификата на USB Flash накопителе:

```
sterragate] dir media:sdc1-usb-_USB_DISK_3.0_90/
```

```
    1  -rwx           612  Mon Jan 19 15:33:40 2026  Hub1.cer
    2  -rwx           530  Mon Jan 19 17:31:08 2026  Hub1.request
```

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

6.5. Импортируйте сертификат в базу криптошлюза:

```
sterragate] run cert_mgr import -f media:sdc1-usb-_USB_DISK_3.0_90/Hub1.cer
```

6.6. Убедитесь, что сертификат успешно импортирован:

```
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
```

Видно, что сертификат криптошлюза импортирован как **local** (local означает, что для данного сертификата есть соответствующий закрытый ключ в ключевом контейнере).

6.7. Выполните проверку статуса сертификатов:

```
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 криптошлюза COC отсутствует, поэтому проверка не может быть осуществлена. Далее будет описан процесс настройки автоматической загрузки COC и импортирование его в базу криптошлюза. Загрузка COC осуществляется по протоколу HTTP с заданной периодичностью.

Настройка паролей доступа к консоли cisco-like

1. Войдите в cisco-like консоль из консоли управления:

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.

```
sterragate] configure
```

```
Entering cs_console...
Password:
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 000c.29eb.5a96
  Internet address is 192.168.4.1/24
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 000c.29eb.5aa0
  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=0.349 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.305 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.296 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.394 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.254 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 102ms
rtt min/avg/max/mdev = 0.254/0.319/0.394/0.051 ms
```

Настройка шифрования

1. Параметры IKEv2.

   1.1. Создайте набор параметров аутентификации с использованием сертификатов в рамках протокола IKEv2 под именем `IKEv2_AUTH`:

```
Hub1#configure terminal
```

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

```
Hub1(config)#crypto ikev2 authentication sig IKEv2_AUTH
```

   1.2. Укажите в качестве типа идентификатора (identity) отличительное имя (Distinguished Name, DN) x509 сертификата:

```
Hub1(conf-ikev2-auth)#identity dn
Hub1(conf-ikev2-auth)#exit
```

   1.3. Создайте набор алгоритмов под номером 1 для защиты IKEv2 туннеля:

```
Hub1(config)#crypto ikev2 proposal 1
Hub1(config-ikev2-proposal)#encryption magma
Hub1(config-ikev2-proposal)#group gost3410-12-256
Hub1(config-ikev2-proposal)#prf streebog512
```

   1.4. Создайте политику IKEv2 под номером 1 и добавьте в нее созданный выше набор алгоритмов:

```
Hub1(config)#crypto ikev2 policy 1
Hub1(config-ikev2-policy)#proposal 1
Hub1(config-ikev2-policy)#exit
```

1.5. Создайте профиль настроек IKEv2 под именем IKEv2_ALGS:

```
Hub1(config)#crypto ikev2 profile IKEv2_ALGS
```

1.5.1. Укажите использование политики 1:

```
Hub1(config-ikev2-profile)#set policy 1
```

1.5.2. Укажите созданный ранее набор параметров аутентификации `IKEv2_AUTH` для локальной аутентификации и аутентификации партнера:

```
Hub1(config-ikev2-profile)#authentication local IKEv2_AUTH
Hub1(config-ikev2-profile)#authentication remote IKEv2_AUTH
```

1.5.3. Настройте параметры DPD (dead peer detection), срабатывающего по отсутствию входящего трафика в IPsec или в IKEv2 туннеле:

```
Hub1(config-ikev2-profile)#dpd 3 on-demand
Hub1(config-ikev2-profile)#exit
```

Примечание: если в туннеле в течение 3 секунд отсутствует входящий трафик, но присутствует исходящий, то будет инициирована DPD сессия. Если партнер не отвечает на DPD сессию, то выполняется попытка перестроения IKEv2 и соответствующего IPsec туннеля.

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 набор алгоритмов с именем IKEv2_IPSEC_ALGS для защиты IPsec туннелей (в данном сценарии используется комбинированный алгоритм шифрования и имитозащиты ГОСТ 34.12-2015 Магма в режиме MGM):

```
Hub1(config)#crypto ikev2 ipsec-proposal IKEv2_IPSEC_ALGS
Hub1(ipsec-proposal)#esp encryption magma-mgm
Hub1(ipsec-proposal)#exit
```

2.3. Создайте список доступа (ACL) для IPv4 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
```

2.4. Создайте крипто-карту (имя `VPN`, раздел 1) для построения IPsec туннеля между центральным офисом и филиалом:

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

2.4.1. Укажите список доступа для защищаемого трафика:

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

2.4.2. Укажите набор алгоритмов для защиты IPsec туннелей:

```
Hub1(config-crypto-map)#set ikev2-ipsec-proposal IKEv2_IPSEC_ALGS
```

2.4.3. Прикрепите IKEv2 профиль к криптокарте:

```
Hub1(config-crypto-map)#set ikev2-profile IKEv2_ALGS
```

2.4.4. 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
```

2.4.5. 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
```

2.4.6. Включите туннельный режим IPsec:

```
Hub1(config-crypto-map)#set mode tunnel
```

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

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

2.5. 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 СОС не загрузился, то проверьте файл журнала, например:

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

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

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

3. Начиная с версии 5.0 СОС 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 КВ» 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 IKEv2, 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
```

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
```

Опционально: сохранение контрольных сумм конфигурации

После завершения процесса конфигурации есть возможность сохранить контрольную сумму настроек криптошлюза. Для этого перейдите в консоль управления и выполните команду `record checksum Hub1`:

```
Hub1] record checksum Hub1
```

При этом на рабочем ключевом носителе создается файл с контрольными суммами текущих настроек, включая следующие:

- Cisco-like и LSP конфигурация;
- Локальные сертификаты;
- Настройки протоколирования;
- Настройки VPN драйвера.

После этого при выполнении команды `deploy` в консоли управления перед запуском VPN сервиса будет выполняться проверка контрольных сумм, в случае обнаружения несоответствия будет выведено предупреждение и VPN сервис запущен не будет. Для обновления контрольных сумм нужно будет повторно выполнить команду `record checksum Hub1` в консоли управления криптошлюза.

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. Все действия на криптошлюзе Spoke1 выполняются Администратором сети.

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст консоли cisco-like;
- текст LSP.

Проверка работоспособности стенда

Проверку работоспособности нужно выполнять после завершения настройки всех устройств стенда.

Проверка IP связности

1. Проверьте, что с криптошлюзов 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
```

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

2. Проверьте, что с защищаемых устройств Host-b-hub1 и Host-b-spoke1 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
```

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

3. Проверьте, что с криптошлюза Hub1 доступен по ICMP криптошлюз Spoke1 (по внешнему интерфейсу). Для этого выполните команду `ping` из cisco-like консоли криптошлюза Hub1.

```
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
```

Видно, что с криптошлюза Hub1 доступен по ICMP криптошлюз Spoke1.

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
```

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
```

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

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

IKEv2 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 1423 1395

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 528 528
```

```
Spoke1#run sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 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 1395 1423

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 528 528
```

Видно, что между криптошлюзами 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 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
```

Приложение

Конфигурации криптошлюза Hub1

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity address
crypto ipsec security-association lifetime kilobytes unlimited
username          cscons          privilege          15          secret          5
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Hub1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
logging console critical
!
!
crypto ikev2 proposal 1
 encryption magma
 group gost3410-12-256
 prf streebog512
!
crypto ikev2 policy 1
 proposal 1
!
crypto ikev2 authentication sig IKEv2_AUTH
 identity dn
 set identity-mapping auto
!
crypto ikev2 profile IKEv2_ALGS
 authentication remote IKEv2_AUTH
 authentication local IKEv2_AUTH
 set policy 1
 dpd 3 on-demand
!
crypto ikev2 ipsec-proposal IKEv2_IPSEC_ALGS
 esp encryption 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 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 ikev2-ipsec-proposal IKEv2_IPSEC_ALGS
 set ikev2-profile IKEv2_ALGS
 set security-association mss-fix out
 set mode tunnel
 set peer 172.16.1.2
 set pre-fragmentation
!
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
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Mon Jan 19 16:46:52 2026

GlobalParameters(
    Title                       = "This LSP was automatically generated by CSP Converter
at Mon Jan 19 16:46:52 2026 (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
)

IKEParameters2 (
    SessionTimeout = 20
    ResponseTimeMax = 10
    SendRetries = 5
    RetryTimeBase = 1
    RetryTimeMax = 10
    CookieChallenge = 10
    NATKeepaliveInterval = 20
    NATKeepaliveRetain = 300
    FragmentSize = 576
    FragmentSize6 = 1280
)

IKEProposal2 n1 (
    EncryptionAlg = ENCR_MAGMA_MGM_KTREE
    PrfAlg        = PRF_HMAC_STREEBOG_512
    KeyAlg        = GOST3410_2012_256
)

IPsecProposal2 IKEv2_IPSEC_ALGS (
    ESPEncryptionAlg = ENCR_MAGMA_MGM_KTREE
)

IdentityEntry id:local:IKEv2_AUTH (
    DistinguishedName = USER_SPECIFIC_DATA
)

IKEAuthSignature2 local:IKEv2_AUTH (
    Identity  = id:local:IKEv2_AUTH
    Credential  = CertDescription()
    SendCertMode  = ALWAYS
)

IdentityEntry id:remote:IKEv2_AUTH (
    DistinguishedName = USER_SPECIFIC_DATA
```

```
)

IKEAuthSignature2 remote:IKEv2_AUTH (
    Identity  = id:remote:IKEv2_AUTH
    Credential  = CertDescription()
    SendRequestMode  = ALWAYS
)

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

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

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

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

IKEParameters(
    FragmentSize = 0
)

IKERule2 IKERule2:VPN:1
(
    IKEPeerIPFilter = 172.16.1.2
    Proposals = n1
    LocalAuth  = local:IKEv2_AUTH
    RemoteAuth  = remote:IKEv2_AUTH
    DPDMode             = ONDEMAND
    DPDIdleDuration     = 3
    Priority            = 10
)

IPsecAction2 IPsecAction2:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.1.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = IKEv2_IPSEC_ALGS
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule2:VPN:1
)

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

```
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction2: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
)
```

Конфигурации криптошлюза Spoke1

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity address
crypto ipsec security-association lifetime kilobytes unlimited
username          cscons          privilege          15          secret          5
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Spoke1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
logging console critical
!
!
crypto ikev2 proposal 1
 encryption magma
 group gost3410-12-256
 prf streebog512
!
crypto ikev2 policy 1
 proposal 1
!
crypto ikev2 authentication sig IKEv2_AUTH
```

```
 identity dn
 set identity-mapping auto
!
crypto ikev2 profile IKEv2_ALGS
 authentication remote IKEv2_AUTH
 authentication local IKEv2_AUTH
 set policy 1
 dpd 3 on-demand
!
crypto ikev2 ipsec-proposal IKEv2_IPSEC_ALGS
 esp encryption 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 ikev2-ipsec-proposal IKEv2_IPSEC_ALGS
 set ikev2-profile IKEv2_ALGS
 set security-association mss-fix out
 set mode tunnel
 set peer 172.16.100.2
 set pre-fragmentation
!
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
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Mon Jan 19 16:47:38 2026

GlobalParameters(
    Title                       = "This LSP was automatically generated by CSP Converter
at Mon Jan 19 16:47:38 2026 (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
)

IKEParameters2 (
    SessionTimeout = 20
    ResponseTimeMax = 10
    SendRetries = 5
    RetryTimeBase = 1
    RetryTimeMax = 10
    CookieChallenge = 10
    NATKeepaliveInterval = 20
    NATKeepaliveRetain = 300
    FragmentSize = 576
    FragmentSize6 = 1280
)
```

```
IKEProposal2 n1 (
    EncryptionAlg = ENCR_MAGMA_MGM_KTREE
    PrfAlg        = PRF_HMAC_STREEBOG_512
    KeyAlg        = GOST3410_2012_256
)

IPsecProposal2 IKEv2_IPSEC_ALGS (
    ESPEncryptionAlg = ENCR_MAGMA_MGM_KTREE
)

IdentityEntry id:local:IKEv2_AUTH (
    DistinguishedName = USER_SPECIFIC_DATA
)

IKEAuthSignature2 local:IKEv2_AUTH (
    Identity  = id:local:IKEv2_AUTH
    Credential  = CertDescription()
    SendCertMode  = ALWAYS
)

IdentityEntry id:remote:IKEv2_AUTH (
    DistinguishedName = USER_SPECIFIC_DATA
)

IKEAuthSignature2 remote:IKEv2_AUTH (
    Identity  = id:remote:IKEv2_AUTH
    Credential  = CertDescription()
    SendRequestMode  = ALWAYS
)

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

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

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

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

IKEParameters(
    FragmentSize = 0
)

IKERule2 IKERule2:VPN:1
(
    IKEPeerIPFilter = 172.16.100.2
    Proposals = n1
    LocalAuth  = local:IKEv2_AUTH
```

```
    RemoteAuth  = remote:IKEv2_AUTH
    DPDMode             = ONDEMAND
    DPDIdleDuration     = 3
    Priority            = 10
)

IPsecAction2 IPsecAction2:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = IKEv2_IPSEC_ALGS
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule2: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 = IPsecAction2: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
)
```