Организация защищенного взаимодействия между подсетями при одновременном использовании нескольких УЦ

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 рекомендации по использованию нескольких удостоверяющих центров при настройке криптошлюзов «С-Терра Шлюз» для обеспечения безопасного взаимодействия между защищаемыми подсетями центрального офиса и филиала. Дополнительно данный сценарий описывает настройку безопасного взаимодействия между защищаемыми подсетями центрального офиса и удаленным VPN клиентом (подробная настройка «С-Терра Клиент» не описывается). 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 и VPN клиент, для аутентификации будут использовать сертификаты в формате x509. В рамках данного сценария для аутентификации партнеры будут использовать сертификаты, выпущенные на разных удостоверяющих центрах. Рассматриваются следующие варианты:

- Оба удостоверяющих центра являются корневыми;
- Один из удостоверяющих центров является корневым, а второй – подчинённым.

В качестве криптопровайдера будет использована криптографическая библиотека, разработанная компанией «С-Терра СиЭсПи». Криптошлюзы в центральном офисе и филиале версии 5.0, VPN клиент – «С-Терра Клиент» версии 5.0.

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

Для переноса запросов и сертификатов между криптошлюзами и центром выпуска сертификатов, а также установочных пакетов «С-Терра Клиент» требуется USB Flash накопитель.

Требования к квалификации администратора

Администратор должен обладать обширными знаниями в области сетевой информационной безопасности, иметь опыт работы с аналогичным оборудованием/программным обеспечением, знать и понимать следующие технологии и протоколы: PKI, IPsec, NAT, Firewall, routing, switching.

Администратор должен быть ознакомлен со сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)». Далее по тексту он будет называться базовым сценарием.

Требования к инфраструктуре

1. Требования к устройствам.

   1.1. «С-Терра Шлюз».

      1.1.1 Устройства «С-Терра Шлюз» должны быть инициализированы (подробнее на 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. Устройство Client1.

      1.2.1 На компьютере пользователя (устройство Client1 на схеме) должна быть установлена ОС Windows (поддерживаемые версии 8, 8.1, 10, 11).

   1.3. Центр выпуска сертификатов.

      1.3.1 Должны быть настроены центры выпуска сертификатов (удостоверяющие центры, далее УЦ) для IPsec. Устройства с именами Certification_authority_X на схеме.

      1.3.2 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).

      1.3.3 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.

   1.4. HTTP сервер для распространения списка отозванных сертификатов.

      1.4.1 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

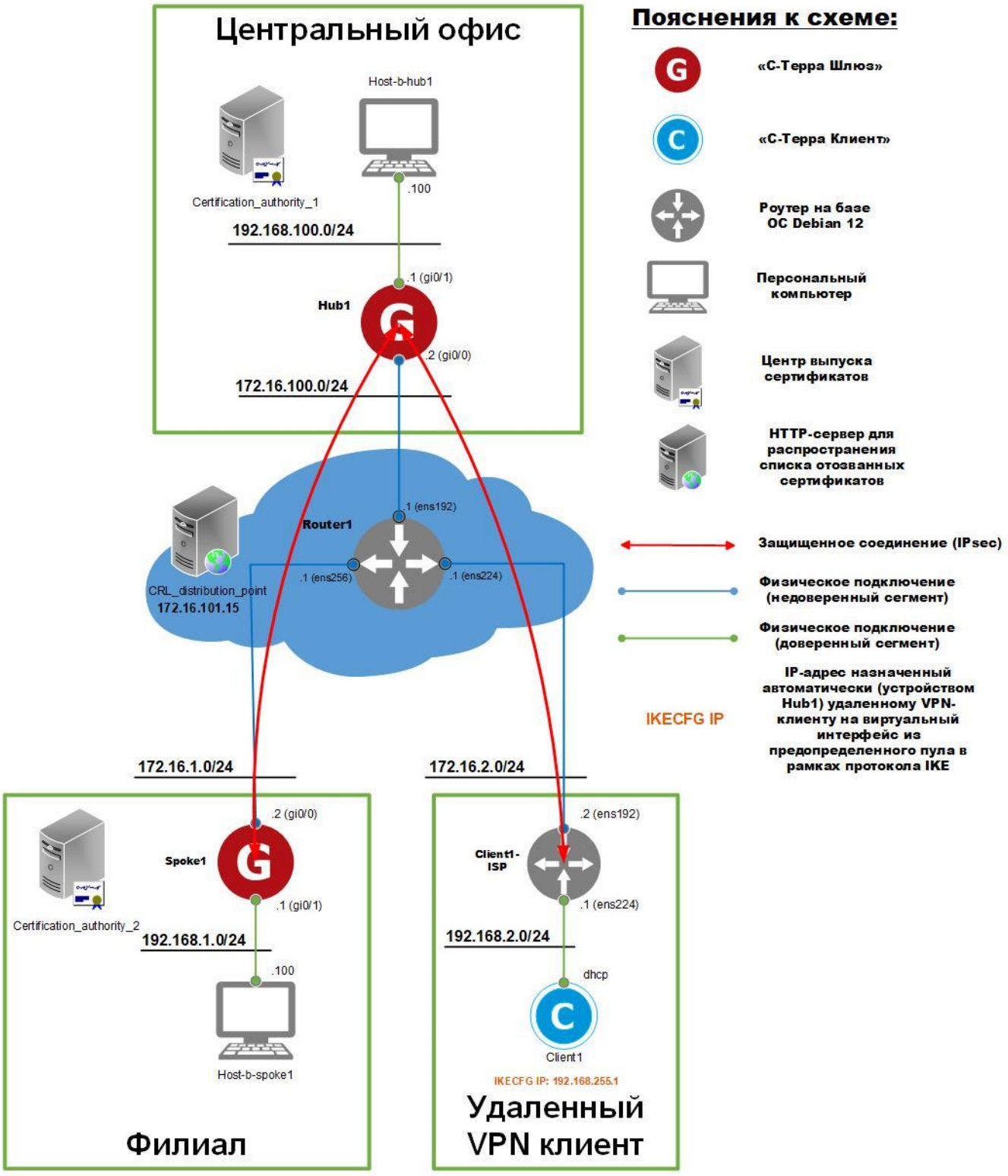


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: (корневой либо подчинённый) центр выпуска сертификатов (Certification_authority_1), криптошлюз «С-Терра Шлюз» (Hub1) версии 5.0 и персональный компьютер (Host-b-hub1).

   1.2. В филиале размещаются: (корневой либо подчинённый) центр выпуска сертификатов (Certification_authority_2), криптошлюз «С-Терра Шлюз» (Spoke1) версии 5.0 и персональный компьютер (Host-b-spoke1).

   1.3. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point), маршрутизаторы (Router1, Client1-ISP) и компьютер пользователя (Client1).

2. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизаторы Router1 и Client1-ISP.

   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. Устройство Client1 получает адрес по протоколу DHCP и подключается к сети Интернет с помощью статической маршрутизации. Роль шлюза по умолчанию играет устройство и Client1-ISP.

3. Параметры шифрования IPsec.

   Весь IPv4 трафик между подсетями центрального офиса (192.168.100.0/24), филиалом (192.168.1.0/24) и удаленным VPN клиентом 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 криптошлюзом центрального офиса и удаленным VPN клиентом может только трафик от Client1 в подсеть центрального офиса.

   Для аутентификации криптошлюз Hub1 использует сертификат, выданный Certification_authority_1, а Spoke1 и Client1 – Certification_authority_2.

   Рассматриваются следующие варианты использования УЦ:

   - Оба удостоверяющих центра (Certification_authority_1 и Certification_authority_2) являются корневыми;
   - УЦ Certification_authority_1 является корневым, а Certification_authority_2 – подчинённым по отношению к Certification_authority_1;
   - УЦ Certification_authority_2 является корневым, а Certification_authority_1 – подчинённым по отношению к Certification_authority_2.

   3.1. Параметры протокола IKE:

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

3.2. Параметры протокола ESP:

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

4. Параметры межсетевого экранирования.

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

   4.1. На LAN интерфейсах IPv4 трафик не контролируется.

   4.2. На LAN интерфейсах IPv6 трафик заблокирован.

   4.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

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

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

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

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

   4.4. На WAN интерфейсах IPv6 трафик заблокирован.

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

5. IKECFG адрес.

   Компьютеру пользователя на специальный виртуальный сетевой интерфейс будет назначен IP-адрес посредством IKECFG из пула 192.168.255.0/24 (данный пул задается на криптошлюзе Hub1). Все взаимодействия между компьютером пользователя и защищаемой подсетью центрального офиса будут осуществляться от этого IP-адреса, а не от IP-адреса, который был выдан по DHCP от маршрутизатора Client-ISP устройству Client1. Данная настройка используется для того, чтобы избежать совпадения IP-адресов у удаленных клиентов в случае отсутствия возможности осуществлять контроль над адресным пространством компьютеров удаленных клиентов.

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 Client1-ISP

1. Настройте IP адрес – 172.16.2.2 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 192.168.2.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Настройте на интерфейсе ens224 DHCP сервер, раздающий доступные IP адреса из подсети этого интерфейса (192.168.2.2 – 192.168.2.254) и требуемые сетевые настройки.
4. Настройте NAT (SNAT либо маскарадинг), транслирующий все адреса сети 192.168.2.0/24 в адрес внешнего интерфейса ens192.
5. Задайте маршрут по умолчанию через 172.16.2.1.
6. Разрешите прохождение IP трафика (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. Доверенным каналом связи также считается канал, защищенный при помощи протокола IPsec.

Начальные настройки

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

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

Подробная настройка с описанием команд приведена в разделе «Начальные настройки» базового сценария.

Настройки PKI (запросы и сертификаты)

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

Сгенерируйте запрос на сертификат криптошлюза, доставьте его на УЦ Certification_authority_1 и выпустите там. Затем доставьте его вместе с сертификатами УЦ (Certification_authority_1, Certification_authority_2) на криптошлюз и импортируйте их в базу Продукта. При импортировании сертификатов УЦ и криптошлюза в базу Продукта выполните следующие действия:

1. Импортируйте сертификат УЦ в базу Продукта:

   1.1. Импорт сертификата УЦ Certification_authority_1:

   1.1.1 Если УЦ Certification_authority_1 является корневым, то импортируйте его как `trusted`:

```
administrator@sterragate] run cert_mgr import -f media:1E181F1C181EF28F/ca1.cer -t
```

   1.1.2 Если УЦ Certification_authority_1 является подчинённым по отношению к Certification_authority_2 то импортируйте сертификат УЦ Certification_authority_1 как `remote`:

```
administrator@sterragate] run cert_mgr import -f media:1E181F1C181EF28F/ca1.cer
```

   Примечание – Команда выше устанавливает сертификат подчинённого УЦ Certification_authority_1 как `remote`. Это необходимо для того, чтобы при проверке сертификатов, выпущенных Certification_authority_1, проверялась вся цепочка целиком, вплоть до корневого сертификата УЦ Certification_authority_2. Также это необходимо для

корректной отправки всей цепочки сертификатов партнерам, которые ее запросили в рамках протокола IKE.

1.2. Если УЦ Certification_authority_1 является подчинённым по отношению к Certification_authority_2, настройте логику отправки локального сертификата партнеру на криптошлюзе так, чтобы при аутентификации он отправлял свой сертификат, подписанный подчинённым УЦ, вместе с цепочкой сертификатов УЦ при запросе от партнера:

1.2.1 Разрешите редактирование файла настроек:

```
administrator@sterragate] run chmod +w /opt/VPNagent/etc/cs_conv.ini
```

1.2.2 Поменяйте значение параметра `send_cert` из раздела `[auth_cert]` на `AUTO`.

1.2.3 Уберите в файле настроек права на редактирование:

```
administrator@sterragate] run chmod -w /opt/VPNagent/etc/cs_conv.ini
```

1.2.4 Пересчитайте контрольную сумму файла:

```
administrator@sterragate] run integr_mgr calc -f /opt/VPNagent/etc/cs_conv.ini
```

1.2.5 Чтобы обновленная логика отправки локального сертификата партнеру вступила в силу, обновите LSP. Для этого зайдите в cisco-like консоль, перейдите в режим конфигурирования любого интерфейса и выйдите:

```
sterragate(config)#configure terminal
sterragate(config)#interface GigabitEthernet 0/0
sterragate(config)#end
sterragate#exit
```

1.2.6 Убедитесь, что логика отправки локального сертификата партнеру была изменена. Для этого выполните команду `lsp_mgr show` и удостоверьтесь, что значение `SendCertMode` равно `AUTO`.

1.3. Импорт сертификата УЦ Certification_authority_2:

1.3.1 Если УЦ Certification_authority_2 является корневым, то импортируйте его как `trusted`:

```
administrator@sterragate] run cert_mgr import -f media:1E181F1C181EF28F/ca2.cer -t
```

1.3.2 Если УЦ Certification_authority_2 является подчинённым по отношению к Certification_authority_1, то импортировать сертификат УЦ Certification_authority_2 нет необходимости.

1.4. Импортируйте сертификат криптошлюза в базу Продукта:

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

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

1.5.1 В случае, если Certification_authority_2 является корневым:

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

```
Found 3 certificates. No CRLs found.
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 1
2 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
3  Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
```

Примечание – Видно, что сертификаты УЦ импортированы как **trusted** (доверенный), а сертификат криптошлюза как **local** (local означает, что для данного сертификата есть соответствующий ключевой контейнер).

1.5.2 В случае, если Certification_authority_2 является подчинённым по отношению к Certification_authority_1:

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

1.5.3 В случае, если Certification_authority_1 является подчинённым по отношению к Certification_authority_2:

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

```
Found 3 certificates. No CRLs found.
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
2 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
3 Status: remote  C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Sub CA 1
```

1.6. Выполните проверку статуса сертификатов в базе Продукта. Вывод команды `cert_mgr check` будет отличаться в зависимости от того, является УЦ корневым или подчинённым по отношению к другому УЦ. Ниже приведен пример вывода команды, когда оба УЦ являются корневыми:

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

```
1 State: Inactive C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 1
                  Certificate can not be verified.
2 State: Inactive C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
                  Certificate can not be verified.
3 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.

Подробная настройка с описанием команд приведена в разделе «Настройки PKI (запросы и сертификаты)» базового сценария.

Настройка паролей доступа к консоли cisco-like

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

Подробная настройка с описанием команд приведена в разделе «Настройка паролей доступа к консоли cisco-like» базового сценария.

Настройка сетевых параметров

Изменение состояния интерфейсов и назначение IP адресов применяются сразу после ввода соответствующих команд. Политика безопасности применится после выхода из глобального конфигурационного режима.

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)#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
```

6. Проверьте доступность устройства Router1:

```
Hub1(config)#do ping 172.16.100.1
```

```
PING 172.16.100.1 (172.16.100.1) 56(84) bytes of data.
64 bytes from 172.16.100.1: icmp_seq=1 ttl=64 time=2.12 ms
64 bytes from 172.16.100.1: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 172.16.100.1: icmp_seq=3 ttl=64 time=0.969 ms
64 bytes from 172.16.100.1: icmp_seq=4 ttl=64 time=0.424 ms
64 bytes from 172.16.100.1: icmp_seq=5 ttl=64 time=0.386 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4035ms
rtt min/avg/max/mdev = 0.386/0.875/2.121/0.657 ms
```

Настройка шифрования

1. Параметры IKE.

   1.1. Задайте общие параметры IKE:

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

   1.2. Создайте политику, описывающую алгоритмы IKE туннеля:

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

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

```
Hub1(config)#crypto isakmp profile NEW_ALGS
Hub1(conf-isa-prof)#match identity dn
Hub1(conf-isa-prof)#set identity mapping
Hub1(conf-isa-prof)#set policy 1
Hub1(conf-isa-prof)#exit
```

2. Параметры IPsec.

2.1. Создайте IKECFG пул адресов, из которого будут назначаться IKECFG адреса для удаленных VPN клиентов:

```
Hub1(config)#ip local pool IKECFG_POOL 192.168.255.1 192.168.255.254
```

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

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

2.3. Создайте IPsec трансформ, задающий комбинированный алгоритм шифрования и имитозащиты ГОСТ 34.12-2015 Магма в режиме MGM для трафика:

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

2.4. 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
```

2.5. Создайте список доступа (ACL) для IPv4 трафика, который нужно защищать между центральным офисом и удаленными VPN клиентами:

```
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_AND_CLIENTS
Hub1(config-ext-nacl)#permit ip 192.168.100.0 0.0.0.255 192.168.255.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

Пояснение – destination часть этого списка соответствует диапазону адресов IKECFG пула.

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

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

2.6.1 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 динамическую крипто-карту (имя DMAP, раздел 1) для построения IPsec туннеля между центральным офисом и удаленными VPN клиентами:

```
Hub1(config)#crypto dynamic-map DMAP 1
Hub1(config-crypto-map)#match address IPSEC_ACL_HUB1_AND_CLIENTS
```

2.7.1 Настройте выдачу IKECFG адресов из созданного ранее пула:

```
Hub1(config-crypto-map)#set pool IKECFG_POOL
Hub1(config-crypto-map)#set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)#set isakmp-profile NEW_ALGS
Hub1(config-crypto-map)#set dead-connection history off
Hub1(config-crypto-map)#set pre-fragmentation
Hub1(config-crypto-map)#set security-association mss-fix out
Hub1(config-crypto-map)#exit
```

2.8. Создайте раздел 2 крипто-карты VPN и добавьте туда динамическую крипто-карту DMAP:

```
Hub1(config)#crypto map VPN 2 ipsec-isakmp dynamic DMAP
```

2.9. 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
```

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

Настройка политики обработки СОС (CRL)

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

1. Включите проверку СОС:

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

Примечание – Если по обоснованным причинам использование СОС невозможно, то выключите проверку СОС и не включайте автоматическую загрузку СОС (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 CA1 http://172.16.101.15/certcrl1.crl
Hub1(ca-trustpoint)# crl download group CA2 http://172.16.101.15/certcrl2.crl
```

3. Настройте периодичность загрузки СОС 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 СОС в базу Продукта. Ниже приведен пример вывода команды, когда оба УЦ являются корневыми:

```
Hub1#run cert_mgr show
```

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

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

Примечания:

5.1. Если СОС не загрузился, то проверьте файл журнала, например:

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

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

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

5.3. Начиная с версии 5.0 СОС можно удалить из базы Продукта:

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

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

Настройка межсетевого экрана

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
```

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

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

4. Прикрепите правила межсетевого экранирования на WAN интерфейс `GigabitEthernet0/0`:

```
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
```

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.

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст консоли cisco-like;
- текст LSP.

Настройка «С-Терра Клиент»

Разные корневые УЦ

Если сертификаты «С-Терра Клиент» и криптошлюза выпущены разными корневыми УЦ то, чтобы настроить продукт «С-Терра Клиент» для работы с двумя корневыми УЦ, нужно воспользоваться «С-Терра ГСУ» и установить корневые сертификаты как `trusted`, так как через Admin Tool можно добавить только один сертификат УЦ.

Настройка удаленного VPN клиента Client1 выполняется в два этапа. Сначала выполняются действия на «С-Терра ГСУ», включающие в себя создание учетной записи для устройства Client1 и выпуск установочных дистрибутивов «С-Терра Клиент». Затем выполняются действия по установке полученных дистрибутивов «С-Терра Клиент» на устройстве Client1.

В данном сценарии процессы установки «С-Терра ГСУ», подключения к консоли «С-Терра ГСУ», настройки PKI (запросы и сертификаты), создания учетной записи для устройства Client1, создания установочных пакетов и установки полученных дистрибутивов «С-Терра Клиент» на устройстве Client1 описываться не будут. Будут описаны только действия по добавлению второго сертификата корневого УЦ Certification_authority_1 в процессе создания учетной записи для устройства Client1.

Добавление второго сертификата корневого УЦ

Добавление второго сертификата корневого УЦ Certification_authority_1 выполняется в процессе создания учетной записи для устройства Client1 после завершения работы Мастера настроек VPN профиля «С-Терра Клиент» для устройства Client1. Для этого выполните следующие действия:

1. В окне «Редактор VPN профиля» перейдите на вкладку «Сертификаты».
2. В разделе «Доверенные CA-сертификаты» нажмите кнопку «Добавить» (см. Рисунок 2). Выберите сертификат корневого УЦ Certification_authority_1 и добавьте его по кнопке «OK».

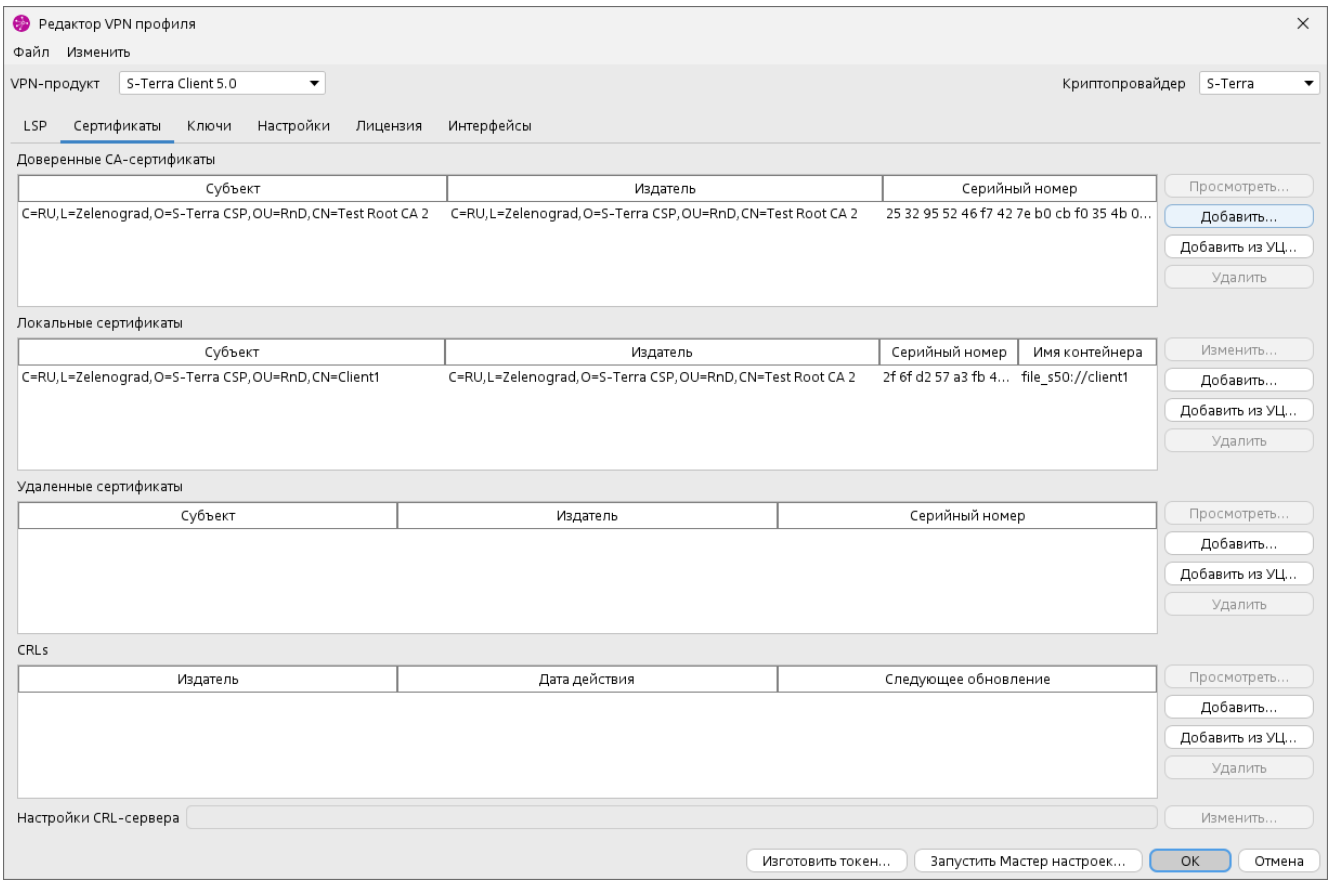


Рисунок 2

3. Убедитесь, что добавленный сертификат корневого УЦ отобразился в списке в разделе «Доверенные CA-сертификаты» (см. Рисунок 3).

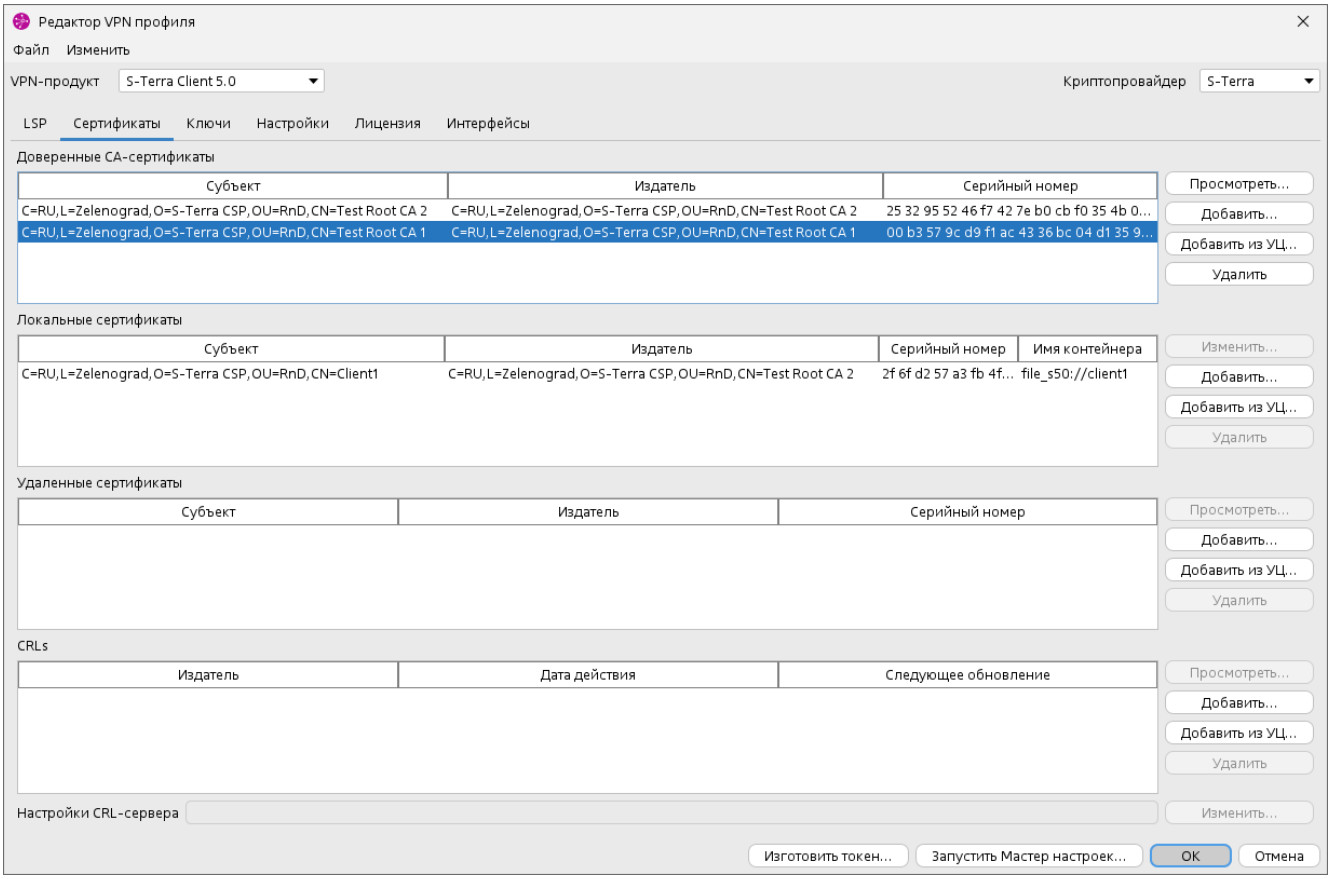


Рисунок 3

В Приложении представлены текст конфигурации для описанного выше варианта использования УЦ для удаленного VPN клиента Client1.

Корневой и подчинённый УЦ

Сертификат «С-Терра Клиент» выпущен подчинённым УЦ

Если сертификат «С-Терра Клиент» выпущен подчинённым УЦ, а сертификат «С-Терра Шлюз» – корневым, также следует воспользоваться «С-Терра ГСУ» и установить корневой сертификат как `trusted`, а подчинённый как `remote`.

Настройка удаленного VPN клиента Client1 выполняется аналогично в два этапа.

Для данного случая аналогично будут описаны только действия по добавлению сертификата, но уже подчинённого УЦ Certification_authority_2 в процессе создания учетной записи для устройства Client1.Также действия по изменению логики отправки локального сертификата партнеру.

Изменение логики отправки локального сертификата криптошлюзу

Изменение логики отправки локального сертификата криптошлюзу выполняется в процессе создания учетной записи для устройства Client1 в ходе работы с Мастером настроек VPN профиля «С-Терра Клиент» для устройства Client1. Для этого выполните следующие действия:

1. Выполните подключение к консоли «С-Терра ГСУ», настройку PKI (запросы и сертификаты) и начните выполнять процесс создания учетной записи для устройства Client1.
2. В окне «Расширенные настройки правила» на вкладке «IKE настройки» слева от параметра «отправить» в выпадающем списке выберите пункт «AUTO». Это позволит «С-Терра Клиент» при аутентификации, помимо локального сертификата, отправлять цепочку сертификатов УЦ в ответ на запрос криптошлюза, включая сертификат подчинённого УЦ (см. Рисунок 4).

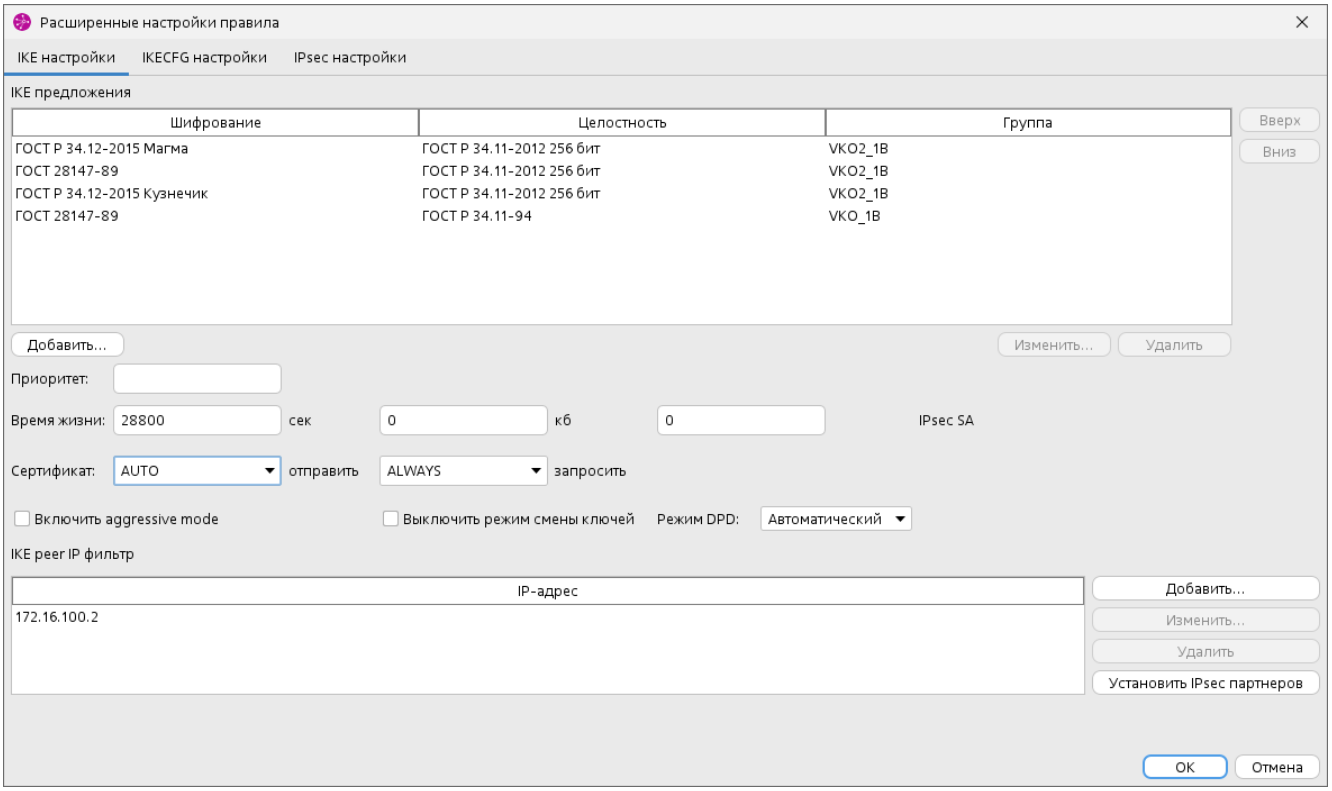


Рисунок 4

3. Завершите процесс создания учетной записи для устройства Client1 в Мастере настроек VPN профиля «С-Терра Клиент».

Добавление сертификата подчинённого УЦ

Добавление сертификата подчинённого УЦ Certification_authority_2 выполняется в процессе создания учетной записи для устройства Client1 после завершения работы Мастера настроек VPN профиля «С-Терра Клиент» для устройства Client1. Для этого выполните следующие действия:

1. В окне «Редактор VPN профиля» перейдите на вкладку «Сертификаты».
2. В разделе «Удаленные сертификаты» нажмите кнопку «Добавить» (см. Рисунок 5). Выберите сертификат подчинённого УЦ Certification_authority_2 и добавьте его по кнопке «OK».

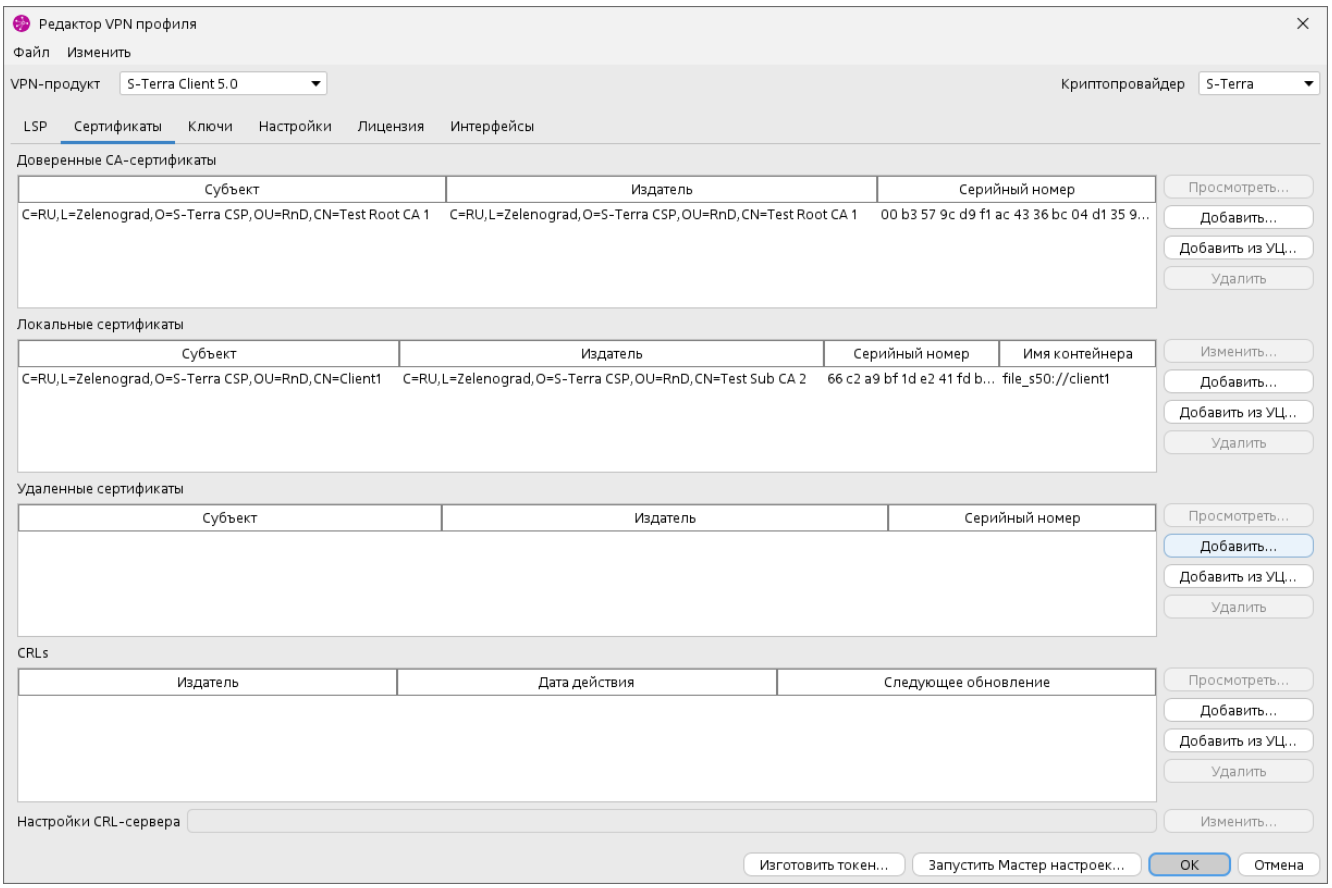


Рисунок 5

3. Убедитесь, что добавленный сертификат подчинённого УЦ отобразился в списке в разделе «Удаленные сертификаты» (см. Рисунок 6).

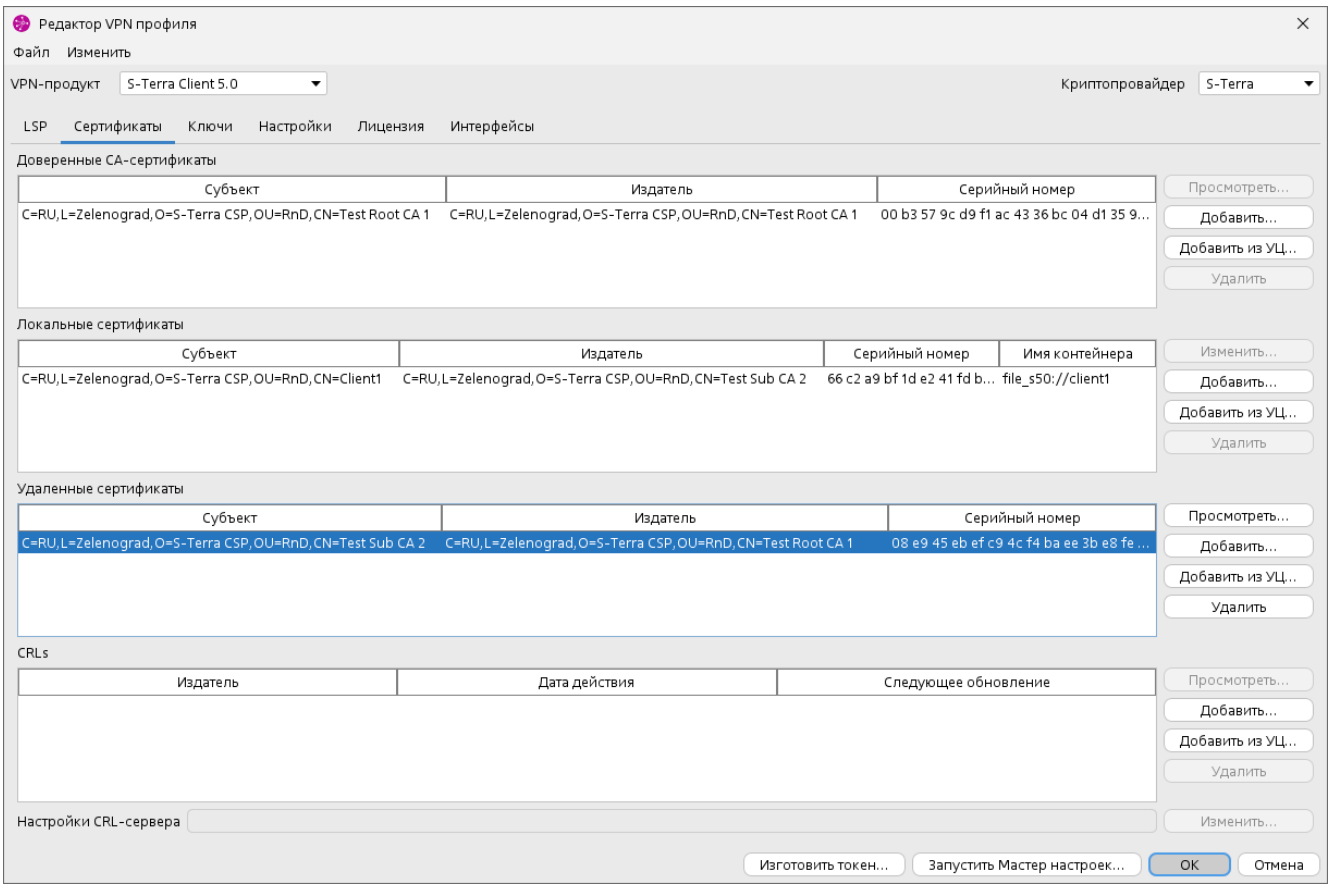


Рисунок 6

Сертификат «С-Терра Клиент» выпущен корневым УЦ

Если сертификат для «С-Терра Клиент» выдан корневым УЦ, а сертификат криптошлюза – подчинённым, можно настроить криптошлюз так, чтобы при аутентификации он отправлял локальный сертификат, подписанный подчинённым УЦ, вместе с цепочкой сертификатов УЦ при запросе от «С-Терра Клиент». Это позволит передавать «С-Терра Клиент» сертификат подчинённого УЦ. Настройка приведена в п. 1.2 главы «Настройки PKI (запросы и сертификаты)» данного сценария.

Предполагается, что формирование установочного пакета «С-Терра Клиент» будет происходить на компьютере администратора. Контейнеры с ключами будут генерироваться также на компьютере администратора. На базе сгенерированных ключей будут выпускаться запросы на сертификаты. Запросы на сертификаты будут переданы на УЦ и на их основе будут получены сертификаты. Затем выполняются действия по установке полученного пакета «С-Терра Клиент» на устройстве Client1.

В данном сценарии процесс установки административной части ПО «С-Терра Клиент» (AdminTool), настройки PKI (запросы и сертификаты), формирования установочного пакета для устройства Client1 и установки полученных дистрибутивов «С-Терра Клиент» на устройстве Client1 описываться не будет.

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, Host-b-spoke1 и Client1 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
```

```
C:\Users\Administrator> ping 192.168.2.1
```

```
Обмен пакетами с 192.168.2.1 по с 32 байтами данных:
Ответ от 192.168.2.1: число байт=32 время<1мс TTL=64
Ответ от 192.168.2.1: число байт=32 время<1мс TTL=64
Ответ от 192.168.2.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.2.1: число байт=32 время<1мс TTL=64

Статистика Ping для 192.168.2.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 1 мсек, Среднее = 0 мсек
```

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

3. Проверьте, что с криптошлюза Hub1 доступны по ICMP криптошлюз Spoke1 (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
```

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

4. Проверьте, что с устройства Client1 доступен криптошлюз Hub1 (по внешнему интерфейсу). Для этого выполните команду `ping` из консоли устройства Client1:

```
C:\Users\Administrator>ping 172.16.100.2
```

```
Обмен пакетами с 172.16.100.2 по с 32 байтами данных:
Ответ от 172.16.100.2: число байт=32 время=1мс TTL=62
Ответ от 172.16.100.2: число байт=32 время<1мс TTL=62
Ответ от 172.16.100.2: число байт=32 время<1мс TTL=62
Ответ от 172.16.100.2: число байт=32 время=1мс TTL=62

Статистика Ping для 172.16.100.2:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 1 мсек, Среднее = 0 мсек
```

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. Ниже приведен пример вывода команды, когда оба УЦ являются корневыми:

```
Hub1#run cert_mgr show
```

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

```
Spoke1#run cert_mgr show
```

```
Found 3 certificates. Found 2 CRLs.
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
2 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 1
3 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Spoke1
4 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 1
5 CRL: C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
```

2. Проверьте, что на устройстве Client1 присутствуют локальный и два доверенных сертификата. Для этого выполните следующие команды из консоли устройства Client1 (при этом должна быть выполнена авторизация в «С-Терра Клиент»):

```
C:\Users\Administrator>cd "C:\Program Files\S-Terra Client"
C:\Program Files\S-Terra Client>cert_mgr.exe show
```

```
Found 4 certificates. No CRLs found.
1 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
2 Status: trusted C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 1
3 Status: local   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1
4 Status: remote  C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
```

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`).

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

```
Spoke1#run cert_mgr check
```

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

4. Проверьте, что на устройстве Client1 статус всех сертификатов Active. Для этого выполните команду `cert_mgr.exe check` из консоли устройства Client1:

```
C:\Program Files\S-Terra Client>cert_mgr.exe check
```

```
1 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 2
2 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Test Root CA 1
3 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1
4 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
```

Видно, что на криптошлюзах и устройстве Client1 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 Client1

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. Построение защищенного соединения между устройством Client1 и криптошлюзом Hub1 выполняется автоматически после выполнения авторизации в «С-Терра Клиент», так как VPN клиент отправляет запрос на получение IKECFG адреса. Для проверки доступности подсети центрального офиса с устройства Client1, отправьте ICMP запросы на устройство Host-b-hub1 из консоли устройства Client1:

```
C:\Users\Administrator>ping 192.168.100.100
```

```
Обмен пакетами с 192.168.100.100 по с 32 байтами данных:
Ответ от 192.168.100.100: число байт=32 время=53мс TTL=62
Ответ от 192.168.100.100: число байт=32 время=17мс TTL=62
Ответ от 192.168.100.100: число байт=32 время=7мс TTL=62
Ответ от 192.168.100.100: число байт=32 время=5мс TTL=62

Статистика Ping для 192.168.100.100:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 5мсек, Максимальное = 53 мсек, Среднее = 20 мсек
```

Для просмотра установленных соединений на криптошлюзе Hub1 выполните следующую команду из cisco-like консоли:

```
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 11 (172.16.100.2,4500)-(172.16.2.2,4500) active 1864 2272
```

```
IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 2 (192.168.100.0-192.168.100.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 256 256
```

Для просмотра установленных соединений на устройстве Client1 в контекстном меню «С-Терра Клиент» выберите пункт «Show SA Information». Откроется окно «VPN SA Monitor», где будут отображаться установленные соединения (см. Рисунок 7).

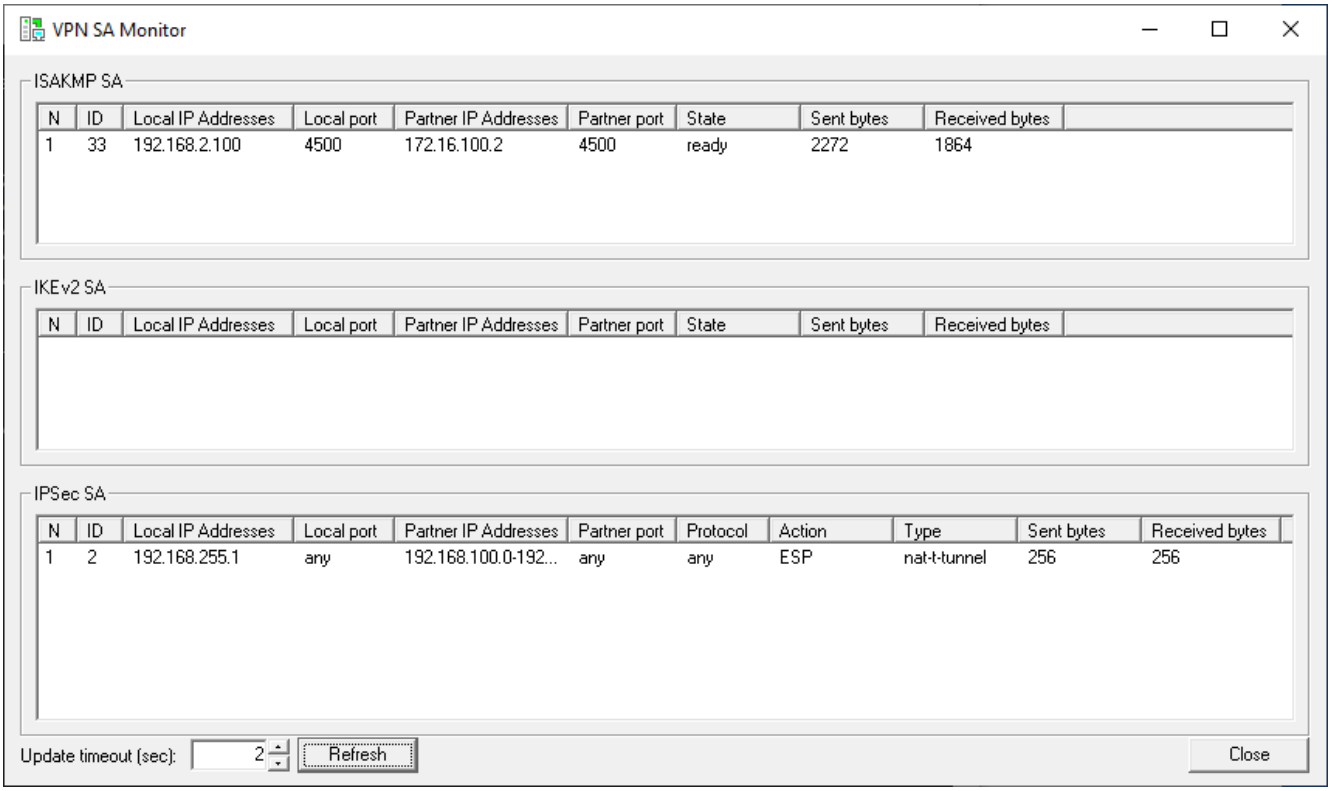


Рисунок 7

Если защищенные соединения не построились, проверьте файл журнала `/var/log/cspvpngate.log` на криптошлюзе Hub1 (при необходимости увеличьте уровень логирования командой `logging trap debugging` в cisco-like консоли) и файл журнала `C:\ProgramData\s-terra\vpn_client_filelog.log` на устройстве Client1.

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 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/x
heIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXWA9dZ/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 profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
ip local pool IKECFG_POOL 192.168.255.1 192.168.255.254
!
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_CLIENTS
 permit ip 192.168.100.0 0.0.0.255 192.168.255.0 0.0.0.255
!
ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE1
 permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
```

```
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto dynamic-map DMAP 1
 match address IPSEC_ACL_HUB1_AND_CLIENTS
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set pool IKECFG_POOL
 set security-association mss-fix out
 set pre-fragmentation
 set dead-connection history off
!
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 dynamic DMAP
!
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 CA1 http://172.16.101.15/certcrl1.crl
 crl download group CA2 http://172.16.101.15/certcrl2.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
certificate 463A75B37E337C964B2B195231F78DA3
30820213308201C0A0030201020210463A75B37E337C964B2B195231F78DA330
...
```

```
FA757770ACA6C55C45D31FB5043DD96BF90B1AE3357B3C
quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Wed Mar 26 12:40:57 2025

GlobalParameters(
    Title                     = "This LSP was automatically generated by CSP Converter
at Wed Mar 26 12:40:57 2025 (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
)

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

```
FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
        DestinationIP = 172.16.100.2
        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
)

AddressPool IKECFG_POOL
(
    IPAddresses = 192.168.255.1..192.168.255.254
)

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 = 172.16.1.2
    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:DMAP:1
(
    Transform = crypto:isakmp:policy:1
    AggrModeAuthMethod  = GOST:Sign:NEW_ALGS
    MainModeAuthMethod  = GOST:Sign:NEW_ALGS
    IKECFGPool          = IKECFG_POOL
    DPDIdleDuration     = 3
    DPDResponseDuration = 2
    DPDRetries          = 5
    Priority            = 100
)

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:DMAP:1
(
    TunnelingParameters = TunnelEntry(
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_GOST_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:2:DMAP: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 = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        Label = "VPN:2:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS"
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.255.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2:DMAP:1 >
        LogEventID = "IPsec:Protect:VPN:2:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS"
    )
)

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/x
heIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXWA9dZ/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 CA1 http://172.16.101.15/certcrl1.crl
 crl download group CA2 http://172.16.101.15/certcrl2.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 463A75B37E337C964B2B195231F78DA3
30820213308201C0A0030201020210463A75B37E337C964B2B195231F78DA330
...
FA757770ACA6C55C45D31FB5043DD96BF90B1AE3357B3C

quit
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A0030201020210058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Wed Mar 26 12:42:41 2025

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Wed Mar 26 12:42:41 2025 (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 VPN клиента Client1

1. Конфигурация LSP:

```
GlobalParameters (
    CRLHandlingMode = BEST_EFFORT
    Title = "This LSP was automatically generated by S-Terra GSU (st) at 2025.03.24
08:53:07"
    Version = LSP_5_0
)
LDAPSettings (
    DropConnectTimeout = 5
    HoldConnectTimeout = 60
    ResponseTimeout = 200
)
SNMPPollSettings (
    LocalIPAddress = 127.0.0.1
    Port = 161
    ReadCommunity = "public"
)
IdentityEntry local_auth_identity_01(
    DistinguishedName *= USER_SPECIFIC_DATA
)
CertDescription local_cert_dsc_01(
    FingerprintMD5 = "E6D2D10FB5EAE4A6AFE82E77112549DA"
    Issuer *= COMPLETE,"OU=RnD,O=S-Terra CSP,L=Zelenograd,C=RU,CN=Test Root CA 2"
    SerialNumber = "250000001EDF9B7E81F42A1D7400000000001E"
    Subject *= COMPLETE,"C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1"
)
CertDescription partner_cert_dsc_01(
)
IKEParameters (
    BlacklogRelaxTime = 120
    BlacklogSessionsMax = 16
    BlacklogSessionsMin = 0
    BlacklogSilentSessions = 4
    DefaultPort = 500
    IKECFGPreferDefaultAddress = FALSE
    InitiatorSessionsMax = 30
    ResponderSessionsMax = 20
    RetryTimeBase = 1
    RetryTimeMax = 30
    SendRetries = 5
    SessionTimeMax = 60
)
AuthMethodGOSTSign auth_method_01(
```

```
    DoNotMapLocalIDToCert = TRUE
    LocalID = local_auth_identity_01
    RemoteCredential *= partner_cert_dsc_01
    SendCertMode = ALWAYS
    SendRequestMode = ALWAYS
)
IKETransform ike_trf_01(
    CipherAlg *= "GR341215M-K256-CFB-65526"
    GroupID *= VKO2_1B
    HashAlg *= "GR341112_256TC26-65128"
    LifetimeSeconds = 28800
    NoSmoothRekeying = FALSE
)
IKETransform ike_trf_02(
    CipherAlg *= "G2814789CPRO1-K256-CBC-65534"
    GroupID *= VKO2_1B
    HashAlg *= "GR341112_256TC26-65128"
    LifetimeSeconds = 28800
    NoSmoothRekeying = FALSE
)
IKETransform ike_trf_03(
    CipherAlg *= "GR341215K-K256-CFB-65528"
    GroupID *= VKO2_1B
    HashAlg *= "GR341112_256TC26-65128"
    LifetimeSeconds = 28800
    NoSmoothRekeying = FALSE
)
IKETransform ike_trf_04(
    CipherAlg *= "G2814789CPRO1-K256-CBC-65534"
    GroupID *= VKO_1B
    HashAlg *= "GR341194CPRO1-65534"
    LifetimeSeconds = 28800
    NoSmoothRekeying = FALSE
)
ESPTransform esp_trf_01(
    CipherAlg *= "GR341215M-K288-MGM-33"
    LifetimeKilobytes = 4608000
    LifetimeSeconds = 3600
)
ESPProposal  esp_proposal_01(
    Transform = esp_trf_01
)
ESPTransform esp_trf_02(
    CipherAlg *= "G2814789CPRO2-K288-CNTMAC-253"
    LifetimeKilobytes = 4608000
    LifetimeSeconds = 3600
)
ESPProposal  esp_proposal_02(
    Transform = esp_trf_02
)
ESPTransform esp_trf_03(
    CipherAlg *= "GR341215K-K352-MGM-251"
    LifetimeKilobytes = 4608000
    LifetimeSeconds = 3600
)
ESPProposal  esp_proposal_03(
    Transform = esp_trf_03
)
ESPTransform esp_trf_04(
    CipherAlg *= "G2814789CPRO1-K256-CBC-254"
    IntegrityAlg *= "GR341194CPRO1-H96-HMAC-65534"
    LifetimeKilobytes = 4608000
```

```
    LifetimeSeconds = 3600
)
ESPProposal  esp_proposal_04(
    Transform = esp_trf_04
)
IKERule ike_rule_01(
    DPDIdleDuration = 60
    DPDMode = ONDEMAND
    DPDResponseDuration = 5
    DPDRetries = 3
    DoNotUseDPD = FALSE
    IKECFGRequestAddress = TRUE
    IKEPeerIPFilter *= 172.16.100.2
    MainModeAuthMethod *= auth_method_01
    Transform *= ike_trf_01,ike_trf_02,ike_trf_03,ike_trf_04
)
IPsecAction ipsec_action_01(
    AdminControlledConnection = FALSE
    ContainedProposals                                                              *=
(esp_proposal_01),(esp_proposal_02),(esp_proposal_03),(esp_proposal_04)
    GroupID *= NO_PFS
    IKERule = ike_rule_01
    NoDeadConnectionHistory = TRUE
    PersistentConnection = TRUE
    TunnelingParameters *= TunnelEntry (
                               Assemble = TRUE
                               DFHandling = COPY
                               PeerAddress = 172.16.100.2
                               ReRoute = FALSE
                               TCPEncapsulation = FALSE
                           )
)
FilterChain filter_chain_input_01(
    IPv4Filters *= Filter (
                        Action = PASS
                        LogEventID = "pass_action_03_01"
                        ProtocolID *= 17
                        SourcePort *= 500
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_03_02"
                        ProtocolID *= 17
                        SourcePort *= 4500
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_04_01"
                        DestinationPort *= 500
                        ProtocolID *= 17
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_05_01"
                        DestinationPort *= 4500
                        ProtocolID *= 17
                    ),Filter (
                        Action = PASS
                        SourceIP *= 172.16.100.2
                        LogEventID = "pass_action_06_01"
                        ProtocolID *= 50
                    ),Filter (
                        Action = PASS
                        SourceIP *= 172.16.100.2
                        LogEventID = "pass_action_07_01"
```

```
                        ProtocolID *= 51
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_08"
                    ),Filter (
                        Action = DROP
                        LogEventID = "drop_action_09"
                    )
    IPv6Filters *= Filter (
                        Action = PASS
                        LogEventID = "pass_action_01_01"
                        ProtocolID *= 17
                        SourcePort *= 500
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_01_02"
                        ProtocolID *= 17
                        SourcePort *= 4500
                    ),Filter (
                        Action = DROP
                        LogEventID = "drop_action_02"
                    )
)
FilterChain filter_chain_output_01(
    IPv4Filters *= Filter (
                        Action = PASS
                        LogEventID = "pass_action_12_01"
                        ProtocolID *= 17
                        SourcePort *= 500
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_13_01"
                        ProtocolID *= 17
                        SourcePort *= 4500
                    ),Filter (
                        Action = PASS
                        DestinationIP *= 172.16.100.2
                        LogEventID = "pass_action_14_01"
                        ProtocolID *= 50
                    ),Filter (
                        Action = PASS
                        DestinationIP *= 172.16.100.2
                        LogEventID = "pass_action_15_01"
                        ProtocolID *= 51
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_16"
                    ),Filter (
                        Action = DROP
                        LogEventID = "drop_action_17"
                    )
    IPv6Filters *= Filter (
                        Action = PASS
                        LogEventID = "pass_action_10_01"
                        ProtocolID *= 17
                        SourcePort *= 500
                    ),Filter (
                        Action = PASS
                        LogEventID = "pass_action_10_02"
                        ProtocolID *= 17
                        SourcePort *= 4500
                    ),Filter (
```

```
                    Action = DROP
                    LogEventID = "drop_action_11"
                )
)
FilterChain filter_chain_classification_input_01(
    IPv4Filters *= Filter (
                    Action = PASS
                    LogEventID = "pass_action_18"
                )
)
FilterChain filter_chain_classification_output_01(
    IPv4Filters *= Filter (
                    Action = PASS
                    LogEventID = "pass_action_19"
                )
)
FilterChain filter_chain_ipsec_01(
    IPv4Filters *= Filter (
                    Action = PASS
                    DestinationIP *= 172.16.100.2
                    LogEventID = "pass_action_21_01"
                    ProtocolID *= 50
                ),Filter (
                    Action = PASS
                    DestinationIP *= 172.16.100.2
                    LogEventID = "pass_action_21_02"
                    ProtocolID *= 51
                ),Filter (
                    Action = PASS
                    DestinationIP *= 192.168.100.0/24
                    ExtendedAction = ipsec<sa=ipsec_action_01>
                    LogEventID = "ipsec_action_22"
                ),Filter (
                    Action = PASS
                    LogEventID = "pass_action_23_01"
                    ProtocolID *= 17
                    SourcePort *= 500
                ),Filter (
                    Action = PASS
                    LogEventID = "pass_action_24_01"
                    ProtocolID *= 17
                    SourcePort *= 4500
                ),Filter (
                    Action = PASS
                    DestinationIP *= 172.16.100.2
                    LogEventID = "pass_action_25_01"
                    ProtocolID *= 50
                ),Filter (
                    Action = PASS
                    DestinationIP *= 172.16.100.2
                    LogEventID = "pass_action_26_01"
                    ProtocolID *= 51
                ),Filter (
                    Action = PASS
                    LogEventID = "pass_action_27"
                )
    IPv6Filters *= Filter (
                    Action = PASS
                    LogEventID = "ike_autopass_action_01"
                    PacketType *= LOCAL_UNICAST,LOCAL_MISDIRECTED
                    ProtocolID *= 17
                    SourcePort *= 500, 4500
```

```
                    ),Filter (
                        Action = DROP
                        LogEventID = "drop_action_20"
                    )
)
NetworkInterface (
    IPsecPolicy = filter_chain_ipsec_01
    InputClassification = filter_chain_classification_input_01
    InputFilter = filter_chain_input_01
    OutputClassification = filter_chain_classification_output_01
    OutputFilter = filter_chain_output_01
)
```