Инспекция трафика в центре с филиалов и мобильных клиентов

Настоящий документ содержит описание способа совместного использования Продуктов компании ООО «С-Терра СиЭсПи» и Продуктов третьих производителей.
ООО «С-Терра СиЭсПи» осуществляет сопровождение настоящего сценария в части настроек Продуктов Компании. Упоминание наименований, продуктов, торговых марок третьих организаций исключительно неформально и не является поддержкой, рекомендацией либо рекламой. ООО «С-Терра СиЭсПи» не несет какой-либо ответственности в отношении работоспособности и использования этих Продуктов. Документ имеет статус вспомогательного материала, который может быть использован технологическими партнерами, компаниями-интеграторами, при разработке собственных решений.
Решения, разработанные на базе данного сценария, могут применяться в действующих сетях/системах только после тестовой и/или опытной эксплуатации.

Введение

Данный сценарий описывает настройку криптошлюзов «С-Терра Шлюз» для обеспечения взаимодействия между защищаемыми подсетями центрального офиса, филиала и удаленным VPN клиентом (компьютером пользователя с ПО «С-Терра Клиент») через устройство, инспектирующее трафик. Трафик между подсетями филиала и клиента не отправляется напрямую, а перенаправляется через центральный офис.

Предварительные требования

Требования к материально-техническому обеспечению

Для переноса запросов и сертификатов между криптошлюзами и центром выпуска сертификатов требуется USB Flash накопитель.

Требования к квалификации администратора

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

Администратор должен быть ознакомлен с базовым сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)» и «Организация защищенного взаимодействия между криптошлюзом «С-Терра Шлюз» и удаленным VPN клиентом «С-Терра Клиент»».

Требования к инфраструктуре

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

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

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

1.2.2. Для выпуска цифровых сертификатов допускается использование встроенного в ОС Windows Server 2016 (или новее) удостоверяющего центра, совместно с сертифицированным СКЗИ «КриптоПро» CSP 5.0 R2 (или новее).

1.2.3. Для тестовых целей можно использовать тестовый УЦ от «КриптоПро» (веб-интерфейс: https://testgost2012.cryptopro.ru/certsrv/certrqxt.asp), либо утилиту ca-tool, входящую в состав пакета auxiliary-tools (пакет auxiliary-tools предназначен для установки на «С-Терра Шлюз» 5.0, документация на ca-tool приведена в инструкции «Центр выпуска тестовых ГОСТ сертификатов для IPsec», документацию и пакет можно загрузить на портале https://doc.s-terra.ru).

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

1.3. Устройство Client1.

На компьютере пользователя (устройство Client1 на схеме) должна быть установлена ОС Windows (поддерживаемые версии 8, 8.1, 10, 11). Для работы с токенами должно быть установлено специальное ПО (см. Требования к материально-техническому обеспечению)

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

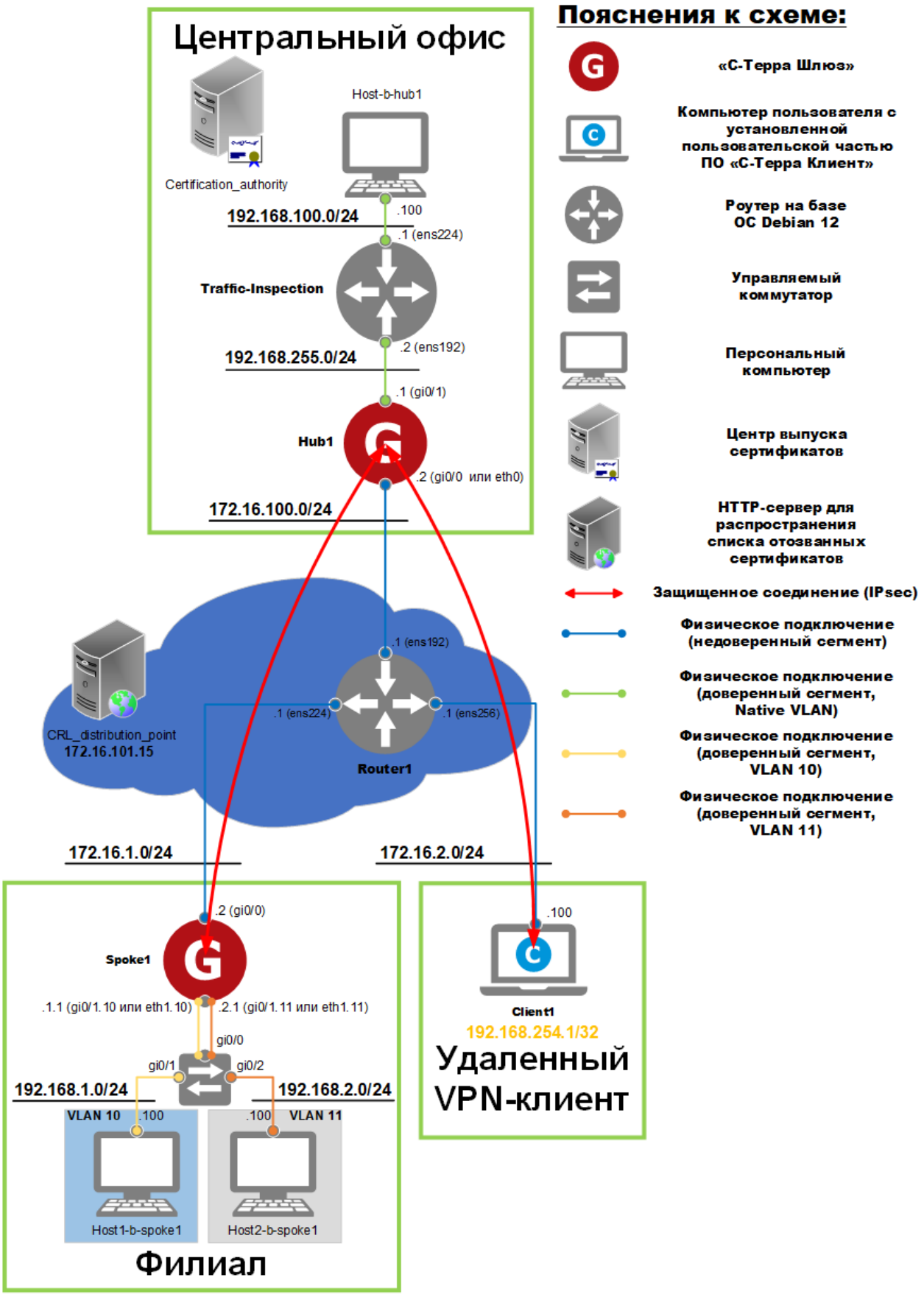


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Размещение устройств.

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

   1.2. В филиале размещаются: криптошлюз «С-Терра Шлюз» (Spoke1) версии 5.0 и персональные компьютеры (Host1-b-spoke1 и Host2-b-spoke1).

   1.3. В неконтролируемом сегменте (синее облако на схеме) размещаются: HTTP сервер для распространения списка отозванных сертификатов (CRL_distribution_point), маршрутизатор (Router1), устройство мобильного клиента «С-Терра Клиент» (Client1).

2. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизатор Router1.

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

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

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

   2.3. Клиент Client1 подключается к сети Интернет с помощью статической маршрутизации (маршрут по умолчанию через маршрутизатор Router1).

3. Настройка маршрутизации на основе политик.

   Для перенаправления трафика через устройство, осуществляющее инспекцию трафика, требуется добавить правила, которые будут перенаправлять трафик.

   3.1. Трафик, приходящий на устройство Hub1 через интерфейс g0/0 (eth0), попадает под правило PBR и направляется в соответствующую таблицу маршрутизации. В данной таблице присутствует только маршрут по умолчанию через устройство Traffic-Inspection (192.168.255.2).

   3.2. Трафик, приходящий на устройство Spoke1 через интерфейс g0/1.10 (eth1.10) или через интерфейс g0/1.11 (eth1.11), попадает под правило PBR и направляется в соответствующую таблицу маршрутизации. В данной таблице присутствует только маршрут по умолчанию через устройство Router1 (172.16.1.1).

   3.3. Для устройства Traffic-Inspection, осуществляющую инспекцию трафика, в данном сценарии настраивается только IP адресация.

   За добавление правил и маршрутов в соответствующую таблицу маршрутизации отвечает сервис `FRR`.

4. IKECFG адрес.

   Удалённому VPN-клиенту на специальный виртуальный сетевой интерфейс будет назначен IP-адрес из IKECFG пула (192.168.254.0/24, данный пул задается на криптошлюзе Hub1). Все взаимодействия между VPN-клиентом и защищаемой подсетью центрального офиса, и между VPN-клиентом и защищаемой подсетью филиала будут осуществляться от этого IP-адреса, а не от IP-адреса Client1(172.16.2.2). 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.

5. Параметры шифрования IPsec.

   Весь IPv4 трафик между подсетями центрального офиса (192.168.100.0/24), подсетями филиала (192.168.1.0/24, 192.168.2.0/24) и клиентом (192.168.254.0/24) будет защищаться с использованием алгоритмов ГОСТ и протокола IPsec в туннельном IPv4 режиме.

   Инициировать защищенное соединение между криптошлюзами Hub1 и Spoke1 могут:

- трафик из подсетей филиала (192.168.1.0/24, 192.168.2.0/24) в подсеть центра (192.168.100.0/24) и обратно;
- трафик между подсетями филиала (между 192.168.1.0/24 и 192.168.2.0/24);
- трафик от клиента (192.168.1.0/24) до подсети филиала (192.168.1.0/24 или 192.168.2.0/24).

Инициировать защищенное соединение между клиентом Client1 и криптошлюзом Hub1 может трафик от клиента до подсети центра или до подсети филиала.

5.1. Параметры протокола IKE

5.1.1. Параметры протокола IKE для версии 5.0:

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

5.1.2. Параметры протокола IKE для совместимости с версией 4.3:

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

5.2. Параметры протокола ESP

5.2.1. Параметры протокола ESP для версии 5.0:

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

5.2.2. Параметры протокола ESP для совместимости с версией 4.3:

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

6. Параметры межсетевого экранирования.

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

6.1. Для «С-Терра Шлюз» версии 5.0:

6.1.1. На LAN интерфейсах IPv4 трафик не контролируется.

6.1.2. На LAN интерфейсах IPv6 трафик заблокирован.

6.1.3. На WAN интерфейсах разрешен входящий IPv4 трафик:

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

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

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

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

6.1.4. На WAN интерфейсах IPv6 трафик заблокирован.

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

6.2. Для «С-Терра Шлюз» версии 4.3:

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

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

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

Настройка стенда

Настройка устройства Router1

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

Настройка устройства Traffic-Inspection

1. Настройте IP адрес – 192.168.255.2 и маску – 255.255.255.0 на сетевом интерфейсе ens192.
2. Настройте IP адрес – 192.168.100.1 и маску – 255.255.255.0 на сетевом интерфейсе ens224.
3. Задайте маршрут по умолчанию через 192.168.255.1.
4. Разрешите прием и отправку ICMP пакетов (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 Host1-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 Host2-b-spoke1

1. Настройте IP адрес – 192.168.2.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.2.1.
3. Разрешите прием и отправку ICMP пакетов (если заблокировано).

Настройка криптошлюза Hub1

Настройка будет происходить локально при помощи консольного подключения.

Настройка может осуществляться и удаленно (по SSH), но исключительно по доверенному каналу связи. Доверенным каналом связи может считаться канал в пределах контролируемой зоны в случае отсутствия в нем нарушителя. 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.

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

Сгенерируйте запрос на сертификат криптошлюза, доставьте его на УЦ и выпустите по нему сертификат. Сертификат криптошлюза вместе с сертификатом УЦ доставьте на шлюз и импортируйте их в базу Продукта.

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

Настройка сетевых параметров

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

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

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

2. Войдите в режим конфигурации:

```
sterragate#configure terminal
```

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

3. Задайте имя устройства:

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

4. Включите внешний (`GigabitEthernet0/0`) и внутренний (`GigabitEthernet0/1`) интерфейсы:

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

5. 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.94c4
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 0050.569e.7f79
  MTU 1500 bytes
```

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

6. Настройте IP адреса на внешнем и внутренних интерфейсах в соответствии со схемой стенда:

```
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 192.168.255.1 255.255.255.0
Hub1(config-if)#exit
```

7. Задайте маршрут по умолчанию через устройство Router1:

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

8. Задайте маршрут до защищаемой подсети центрального офиса через устройство Traffic-Inspection:

```
Hub1(config)#ip route 192.168.100.0 255.255.255.0 192.168.255.2
```

9. Проверьте доступность устройства Router1:

```
Hub1(config)#do 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=63 time=0.884 ms
108 bytes from 172.16.100.1: icmp_seq=2 ttl=63 time=0.673 ms
108 bytes from 172.16.100.1: icmp_seq=3 ttl=63 time=0.719 ms
108 bytes from 172.16.100.1: icmp_seq=4 ttl=63 time=0.629 ms
108 bytes from 172.16.100.1: icmp_seq=5 ttl=63 time=0.740 ms

--- 172.16.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.629/0.729/0.884/0.086 ms
```

Настройка шифрования

1. Настройте параметры IKE.

   1.1. Задайте общие параметры IKE:

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

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

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

   1.3. Создайте политику **2**, описывающую алгоритмы IKE туннеля для совместимости с версией 4.3:

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

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

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

2.1. Опционально: для совместимости с версией 4.3 создайте профиль настроек ISAKMP (алгоритм IKE) под именем `OLD_ALGS`, чтобы далее его прикрепить к соответствующему разделу крипто-карты:

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

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

```
Hub1(config)#crypto isakmp profile IKE_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)#set policy 2
Hub1(conf-isa-prof)#exit
```

В данном сценарии, при работе с криптошлюзами, по IP адресу можно заранее понять, с каким криптошлюзом работаем и какой профиль настроек следует использовать. При работе с клиентами, какая версия клиента по IKECFG адресу определить нельзя, потому что IKECFG адрес не закреплен за определённым клиентом. Поэтому к профилю `IKE_ALGS` прикрепляются две политики.

3. Параметры IPsec.

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

```
Hub1(config)#ip local pool IKECFG_POOL 192.168.254.1 192.168.254.254
```

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

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

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

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

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

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

3.5. Создайте список доступа (ACL) для трафика, который нужно защищать между центральным офисом и удаленным VPN клиентом:

```
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_AND_CLIENT1
Hub1(config-ext-nacl)# permit ip 192.168.100.0 0.0.0.255 192.168.254.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 192.168.254.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.2.0 0.0.0.255 192.168.254.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

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

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

```
Hub1(config)#ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE1
```

```
Hub1(config-ext-nacl)# permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.254.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.254.0 0.0.0.255 192.168.2.0 0.0.0.255
Hub1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Hub1(config-ext-nacl)#exit
```

3.7. Создайте динамическую крипто-карту (имя DMAP, раздел 1) для построения IPsec туннеля между криптошлюзом центрального офиса и удаленными VPN клиентами:

```
Hub1(config)# crypto dynamic-map DMAP 1
Hub1(config-crypto-map)# match address IPSEC_ACL_HUB1_AND_CLIENT1
Hub1(config-crypto-map)# set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
GOST_ENCRYPT_AND_INTEGRITY
Hub1(config-crypto-map)# set isakmp-profile IKE_ALGS
Hub1(config-crypto-map)# set pool IKECFG_POOL
Hub1(config-crypto-map)# set security-association mss-fix out
Hub1(config-crypto-map)# set pre-fragmentation
Hub1(config-crypto-map)# set dead-connection history off
Hub1(config-crypto-map)# exit
```

1.1. Создайте первый раздел крипто-карты `VPN` для построения IPsec туннеля между криптошлюзом центрального офиса и криптошлюзом филиала:

```
Hub1(config)# crypto map VPN 1 ipsec-isakmp
Hub1(config-crypto-map)# match address IPSEC_ACL_HUB1_AND_SPOKE1
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 security-association mss-fix out
Hub1(config-crypto-map)# set peer 172.16.1.2
Hub1(config-crypto-map)# set pre-fragmentation
Hub1(config-crypto-map)# set dead-connection history off
Hub1(config-crypto-map)# exit
```

3.8. Создайте второй раздел крипто-карты `VPN` и добавьте туда динамическую крипто-карту DMAP:

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

3.9. Прикрепите крипто-карту `VPN` к внешнему интерфейсу `GigabitEthernet0/0`:

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

4. Примените настройки:

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

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 ROOTCA http://172.16.101.15/certcrl.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:

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

Настройка межсетевого экрана осуществляются из cisco-like консоли.

По умолчанию на всех сетевых интерфейсах IPv6 трафик заблокирован – команда `ipv6 default filtering deny`, если требуется разрешить IPv6 трафик, то нужно выполнить команду `ipv6 default filtering permit` и далее уже фильтровать IPv6 трафик по аналогии с IPv4. Подробное описание работы данной команды приведено в сценарии «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv6)».

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

```
Hub1#configure terminal
```

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

```
Hub1(config)#object-group network SELF
```

```
Hub1(config-network-group)# remark WAN IP
Hub1(config-network-group)# host 172.16.100.2
Hub1(config-network-group)#exit
```

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

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

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

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

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

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

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

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

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

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

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

Пример работы аудита:

```
2024-01-11T14:11:01.961041+03:00 Hub1 vpnlogsvc: 00290014 TCP session created:
initiator 172.16.100.2:47778 -> responder 172.16.100.1:22
2024-01-11T14:11:32.757597+03:00 Hub1 vpnlogsvc: 00290001 TCP session closed: initiator
...
172.16.100.2:47778 sent 4592 bytes -> responder 172.16.100.1:22 sent 5800 bytes
2024-01-11T14:11:36.833870+03:00 Hub1 vpnlogsvc: 00290014 UDP session created:
initiator 192.168.100.100:40060 -> responder 10.0.0.16:53
```

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

Настройка PBR

PBR настраивается через консоль сервиса FRR.

1. Перейдите в linux bash, для этого выполните следующую команду в CLI разграничения доступа:

```
administrator@hub1] system
```

```
Entering system shell...
root@Hub1:~#
```

2. Добавьте сервис FRR в автозагрузку и запустите его:

```
root@Hub1:~# systemctl enable frr
```

```
Synchronizing state of frr.service with SysV service script with /lib/systemd/systemd-
sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable frr
Created     symlink     /etc/systemd/system/multi-user.target.wants/frr.service     →
/lib/systemd/system/frr.service.
```

```
root@Hub1:~# systemctl start frr
```

Данные сообщения не являются ошибкой при старте сервиса.

3. Войдите в консоль сервиса FRR, для этого нужно из linux bash набрать следующую команду:

```
root@Hub1:~# vtysh
```

```

Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

Hub1#
```

4. Настройте PBR (см. описание http://docs.frrouting.org):

```
Hub1# configure terminal
```

4.1. Создайте PBR-карту (имя PBR, раздел 1):

```
Hub1(config)# pbr-map PBR seq 1
```

4.2. Укажите для каких пакетов нужно применять политику PBR:

```
Hub1(config-pbr-map)# match src-ip 192.168.1.0/24
```

Пакеты с адресом отправителя из подсети 192.168.1.0/24 будут попадать под политику PBR.

4.3. Укажите куда надо перенаправлять пакеты:

```
Hub1(config-pbr-map)# set nexthop 192.168.255.2
Hub1(config-pbr-map)# exit
```

4.4. Создайте раздел 2 и раздел 3 PBR-карты для подсетей 192.168.2.0/24 и 192.168.254.0/24:

```
Hub1(config)# pbr-map PBR seq 2
Hub1(config-pbr-map)# match src-ip 192.168.2.0/24
Hub1(config-pbr-map)# set nexthop 192.168.255.2
Hub1(config-pbr-map)# exit
Hub1(config)# pbr-map PBR seq 3
Hub1(config-pbr-map)# match src-ip 192.168.254.0/24
Hub1(config-pbr-map)# set nexthop 192.168.255.2
Hub1(config-pbr-map)# exit
```

4.5. Прикрепите PBR-карту к интерфейсу:

```
Hub1(config)# interface eth0
Hub1(config-if)# pbr-policy PBR
Hub1(config-if)# exit
```

```
Hub1(config)# exit
```

4.6. Сохраните настройки сервиса FRR:

```
Hub1# write
```

```
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
```

Настройка криптошлюза Hub1 завершена.

В Приложении представлены тексты конфигураций для криптошлюза Hub1:

- текст консоли cisco-like;
- текст LSP;
- текст консоли FRR.

Настройка криптошлюза Spoke1

Настройка криптошлюза Spoke1 происходит аналогично настройке Spoke1 из базового сценария, за исключением настройки VLAN интерфейсов, списка доступа (ACL) для трафика, который нужно защищать между центральным офисом и филиалом и настроек PBR.

1. Создайте 802.1Q интерфейсы `GigabitEthernet 0/1.10` (VLAN ID 10) и `GigabitEthernet 0/1.11` (VLAN ID 11) (в нотации Linux имя этого интерфейса будет eth1_10) и настройте IP адрес:

```
Hub1(config)#interface GigabitEthernet 0/1.10
Hub1(config-if)#ip address 192.168.1.1 255.255.255.0
Hub1(config-if)#no shutdown
Hub1(config-if)#exit
Hub1(config)#interface GigabitEthernet 0/1.11
Hub1(config-if)#ip address 192.168.2.1 255.255.255.0
Hub1(config-if)#no shutdown
Hub1(config-if)#exit
```

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

```
Spoke1(config)# ip access-list extended IPSEC_ACL_HUB1_AND_SPOKE1
Spoke1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
Spoke1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 192.168.254.0 0.0.0.255
Spoke1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Spoke1(config-ext-nacl)# permit ip 192.168.2.0 0.0.0.255 192.168.100.0 0.0.0.255
Spoke1(config-ext-nacl)# permit ip 192.168.2.0 0.0.0.255 192.168.254.0 0.0.0.255
Spoke1(config-ext-nacl)# permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
Spoke1(config-ext-nacl)#exit
```

Примечание – из-за настроек PBR трафик между подсетями филиала пойдет через центральный офис, поэтому его нужно защищать.

Настройка PBR

PBR настраивается через консоль сервиса FRR.

1. Добавьте сервис FRR в автозагрузку и запустите его из linux bash:

```
root@Spoke1:~# systemctl enable frr
```

```
Synchronizing state of frr.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable frr
Created symlink /etc/systemd/system/multi-user.target.wants/frr.service → /lib/systemd/system/frr.service.
```

```
root@Spoke1:~# systemctl start frr
```

2. Войдите в консоль сервиса FRR, для этого нужно из linux bash набрать следующую команду:

```
root@Spoke1:~# vtysh

Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

Hub1#
```

3. Настройте PBR (см. описание http://docs.frrouting.org):

```
Spoke1# configure terminal
```

3.1. Создайте PBR-карту для VLAN10 (имя PBR.10, раздел 1):

```
Spoke1(config)# pbr-map PBR.10 seq 1
```

3.2. Укажите для каких пакетов нужно применять политику PBR:

```
Spoke1(config-pbr-map)# match src-ip 192.168.1.0/24
```

Пакеты с адресом отправителя из подсети 192.168.1.0/24 будут попадать под политику PBR

3.3. Укажите куда надо перенаправлять пакеты:

```
Spoke1(config-pbr-map)# set nexthop 172.16.1.1
Spoke1(config-pbr-map)# exit
```

3.4. Создайте аналогичную PBR-карту для VLAN11 (имя PBR.11, раздел 1):

```
Spoke1(config)# pbr-map PBR.11 seq 1
Spoke1(config-pbr-map)# match src-ip 192.168.2.0/24
Spoke1(config-pbr-map)# set nexthop 172.16.1.1
Spoke1(config-pbr-map)# exit
```

3.5. Прикрепите PBR-карты к интерфейсам:

```
Spoke1(config)# interface eth1_10
Spoke1(config-if)# pbr-policy PBR.10
Spoke1(config-if)# exit
Spoke1(config)# interface eth1_11
Spoke1(config-if)# pbr-policy PBR.11
Spoke1(config-if)# exit
Spoke1(config)# exit
```

3.6. Сохраните настройки сервиса FRR:

```
Spoke1# write
```

```
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
```

```
Spoke1# exit
```

В Приложении представлены тексты конфигураций для криптошлюза Spoke1:

- текст консоли cisco-like;
- текст LSP;
- текст консоли FRR.

Настройка устройства Client1

Настройка устройства Client1 происходит аналогично настройке Client1 из сценария «Организация защищенного взаимодействия между криптошлюзом «С-Терра Шлюз» и удаленным VPN клиентом «С-Терра Клиент»», за исключением настройки получения IP адреса, маршрута по умолчанию и настройкой IPsec Rules:

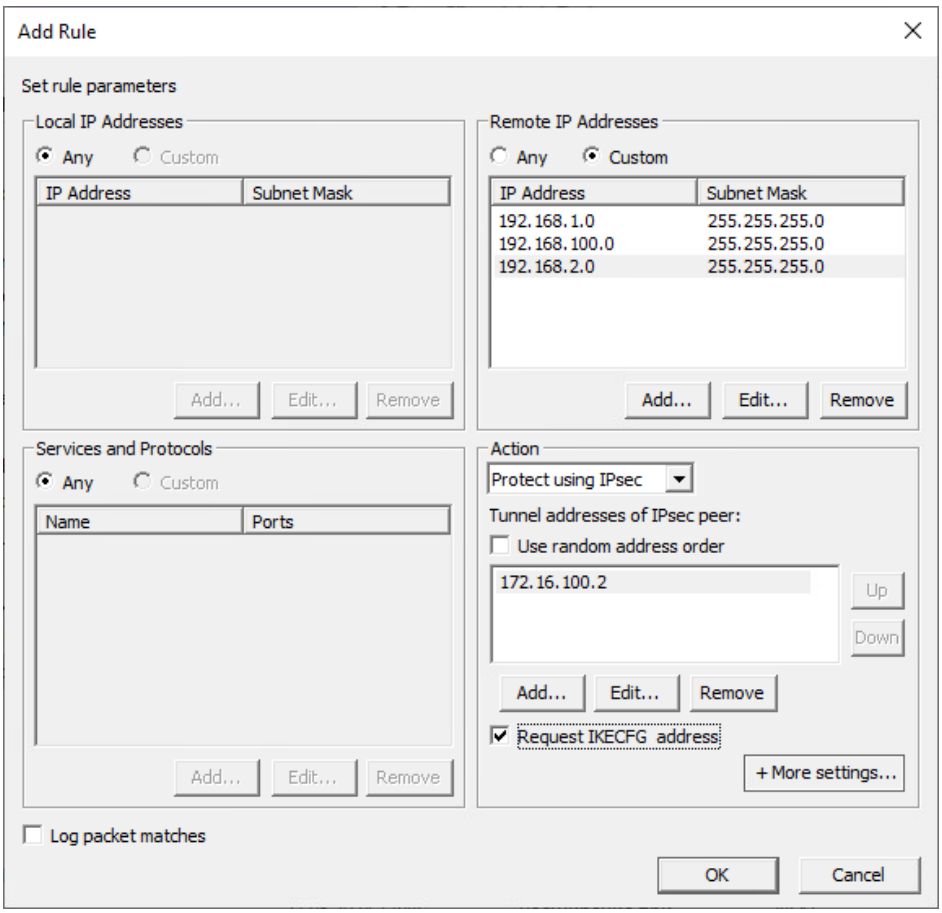


Рисунок 2

В Приложении представлены текст LSP конфигурации для Client1.

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 Host-b-spoke2 доступен по ICMP соответствующий криптошлюз, который является шлюзом по умолчанию. Для этого выполните команду `ping` из linux bash консоли защищаемых устройств.

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

```
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.469 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.398 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.411 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=2.47 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=64 time=0.428 ms

--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.398/0.834/2.467/0.816 ms
```

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

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.81 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.854 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.444 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.390 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=2.29 ms

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

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

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

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

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

3. Проверьте, что с устройства Client1 доступен шлюз по умолчанию Router1. Для этого выполните команду ping из консоли защищаемых устройств:

```
C:\Users\Administrator>ping 172.16.2.1
```

```
Обмен пакетами с 172.16.2.1 по с 32 байтами данных:
Ответ от 172.16.2.1: число байт=32 время<1мс TTL=64
Ответ от 172.16.2.1: число байт=32 время<1мс TTL=64
Ответ от 172.16.2.1: число байт=32 время<1мс TTL=64
Ответ от 172.16.2.1: число байт=32 время<1мс TTL=64

Статистика Ping для 172.16.2.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
```

4. Проверьте, что с криптошлюза Hub1 доступны по ICMP криптошлюз Spoke1 и с Client1 доступен криптошлюз Hub1 (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
```

```
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=63
Ответ от 172.16.100.2: число байт=32 время=1мс TTL=63
Ответ от 172.16.100.2: число байт=32 время=1мс TTL=63
Ответ от 172.16.100.2: число байт=32 время=1мс TTL=63

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

Видно, что с криптошлюза Hub1 доступty по ICMP криптошлюз Spoke1 и с Client1 доступен криптошлюз Hub1. С Hub1 может быть недоступен Client1 для ICMP из-за настроек Firewall на Client1.

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 Status: remote  C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1
4 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
3 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1
```

```
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.

3. Проверьте, что на устройстве Client1 присутствуют локальный и доверенный сертификаты. Для этого выполните следующие команды из консоли устройства Client1:

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

```
Found 3 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=Client1
3 Status: remote  C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
```

```
C:\Program Files\S-Terra Client>cert_mgr.exe 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=Client1
3 State: Active   C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
```

Примечание – если при проверке вход в «С-Терра Клиент» не выполнен, то Client1 выведет следующее сообщение:

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

```
Unable obtain certificates from local settings. Error: NOT LOGGED IN.
Unable obtain CRLs from local settings. Error: NOT LOGGED IN.
```

Проверка построения защищенного соединения

1. Инициируйте построение защищенного соединения между криптошлюзами Hub1 и Spoke1 при помощи ICMP трафика, посылаемого с устройства Host1-b-spoke1 на Host-b-hub1 (можно и наоборот). Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-spoke1.

```
root@Host1-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=61 time=771 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=61 time=23.3 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=61 time=10.1 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=61 time=1.59 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=61 time=2.18 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 1.589/161.665/771.258/304.896 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 3 (172.16.100.2,500)-(172.16.2.100,500) active 2032 2152
2 4 (172.16.100.2,500)-(172.16.1.2,500) active 2032 2072

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 19 (192.168.1.0-192.168.1.255,*)-(192.168.254.1,*) * ESP tunn 0 0
2 20 (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 2072 2032

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. Проверьте, что трафик между Host1-b-spoke1 и Host2-b-spoke1 ходит через устройство Traffic-Inspection при помощи ICMP трафика, посылаемого с устройства Host1-b-spoke1 на Host2-b-spoke1 (можно и наоборот). Для этого:

   3.1. Включите tcpdump на интерфейсе ens192.

```
root@Traffic-Inspection:~# tcpdump -i ens192 -c 8 host 192.168.1.100 and host
192.168.2.100 -n
```

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens192, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

   После включения tcpdump завершит работу только после получения 8 пакетов между 192.168.1.100 и 192.168.2.100

   3.2. Выполните команду `ping` из linux bash консоли защищаемого устройства Host1-b-spoke1.

```
root@Host1-b-spoke1:~# ping 192.168.2.100 -c 5
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_seq=1 ttl=59 time=1231 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=59 time=210 ms
64 bytes from 192.168.2.100: icmp_seq=3 ttl=59 time=2.32 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=59 time=2.68 ms
64 bytes from 192.168.2.100: icmp_seq=5 ttl=59 time=3.32 ms

--- 192.168.2.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4027ms
rtt min/avg/max/mdev = 2.324/289.891/1231.027/477.370 ms, pipe 2
```

3.3. Проверьте вывод tcpdump.

```
root@Traffic-Inspection:~# tcpdump -i ens192 -c 8 host 192.168.1.100 and host
192.168.2.100 -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens192, link-type EN10MB (Ethernet), snapshot length 262144 bytes
02:05:41.153355 IP 192.168.1.100 > 192.168.2.100: ICMP echo request, id 11518, seq 1,
length 64
02:05:41.153380 IP 192.168.1.100 > 192.168.2.100: ICMP echo request, id 11518, seq 1,
length 64
02:05:41.154316 IP 192.168.2.100 > 192.168.1.100: ICMP echo reply, id 11518, seq 1,
length 64
02:05:41.154335 IP 192.168.2.100 > 192.168.1.100: ICMP echo reply, id 11518, seq 1,
length 64
02:05:42.154717 IP 192.168.1.100 > 192.168.2.100: ICMP echo request, id 11518, seq 2,
length 64
02:05:42.154733 IP 192.168.1.100 > 192.168.2.100: ICMP echo request, id 11518, seq 2,
length 64
02:05:42.155928 IP 192.168.2.100 > 192.168.1.100: ICMP echo reply, id 11518, seq 2,
length 64
02:05:42.155939 IP 192.168.2.100 > 192.168.1.100: ICMP echo reply, id 11518, seq 2,
length 64
8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

Видно, что трафик между 192.168.1.100 и 192.168.2.100 ходит через устройство Traffic-Inspection. Из-за того, что пакеты входят и выходят из интерфейса ens192, каждый пакет видно 2 раза.

4. Проверьте, что на криптошлюзах Hub1 и Spoke1 установлено защищенное IPsec соединение между 192.168.1.0 и 192.168.2.0. 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 3 (172.16.100.2,500)-(172.16.2.100,500) active 2032 2152
2 4 (172.16.100.2,500)-(172.16.1.2,500) active 2768 2808

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 19 (192.168.1.0-192.168.1.255,*)-(192.168.254.1,*) * ESP tunn 0 0
2 20 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 440
440
3 21 (192.168.2.0-192.168.2.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 440 440
4 22 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 440 440
```

```
Spoke2#run sa_mgr show
```

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

```
ISAKMP connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (172.16.1.2,500)-(172.16.100.2,500) active 2808 2768

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
2 2 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 440 440
3 3 (192.168.2.0-192.168.2.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 440 440
```

5. 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 защищаемых подсетей с устройства Client1, отправьте ICMP запросы на устройства Host-b-hub1, Host1-b-spoke1 и Host2-b-spoke1 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=31мс TTL=61
Ответ от 192.168.100.100: число байт=32 время=20мс TTL=61
Ответ от 192.168.100.100: число байт=32 время=16мс TTL=61
Ответ от 192.168.100.100: число байт=32 время=1мс TTL=61

Статистика Ping для 192.168.100.100:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 1мсек, Максимальное = 31 мсек, Среднее = 17 мсек
```

```
C:\Users\Administrator>ping 192.168.1.100

Обмен пакетами с 192.168.1.100 по с 32 байтами данных:
Ответ от 192.168.1.100: число байт=32 время=64мс TTL=59
Ответ от 192.168.1.100: число байт=32 время=12мс TTL=59
Ответ от 192.168.1.100: число байт=32 время=3мс TTL=59
Ответ от 192.168.1.100: число байт=32 время=2мс TTL=59

Статистика Ping для 192.168.1.100:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 2мсек, Максимальное = 64 мсек, Среднее = 20 мсек
```

```
C:\Users\Administrator>ping 192.168.2.100

Обмен пакетами с 192.168.2.100 по с 32 байтами данных:
Ответ от 192.168.2.100: число байт=32 время=10мс TTL=59
Ответ от 192.168.2.100: число байт=32 время=3мс TTL=59
Ответ от 192.168.2.100: число байт=32 время=3мс TTL=59
Ответ от 192.168.2.100: число байт=32 время=2мс TTL=59

Статистика Ping для 192.168.2.100:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 2мсек, Максимальное = 10 мсек, Среднее = 4 мсек
```

6. 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:

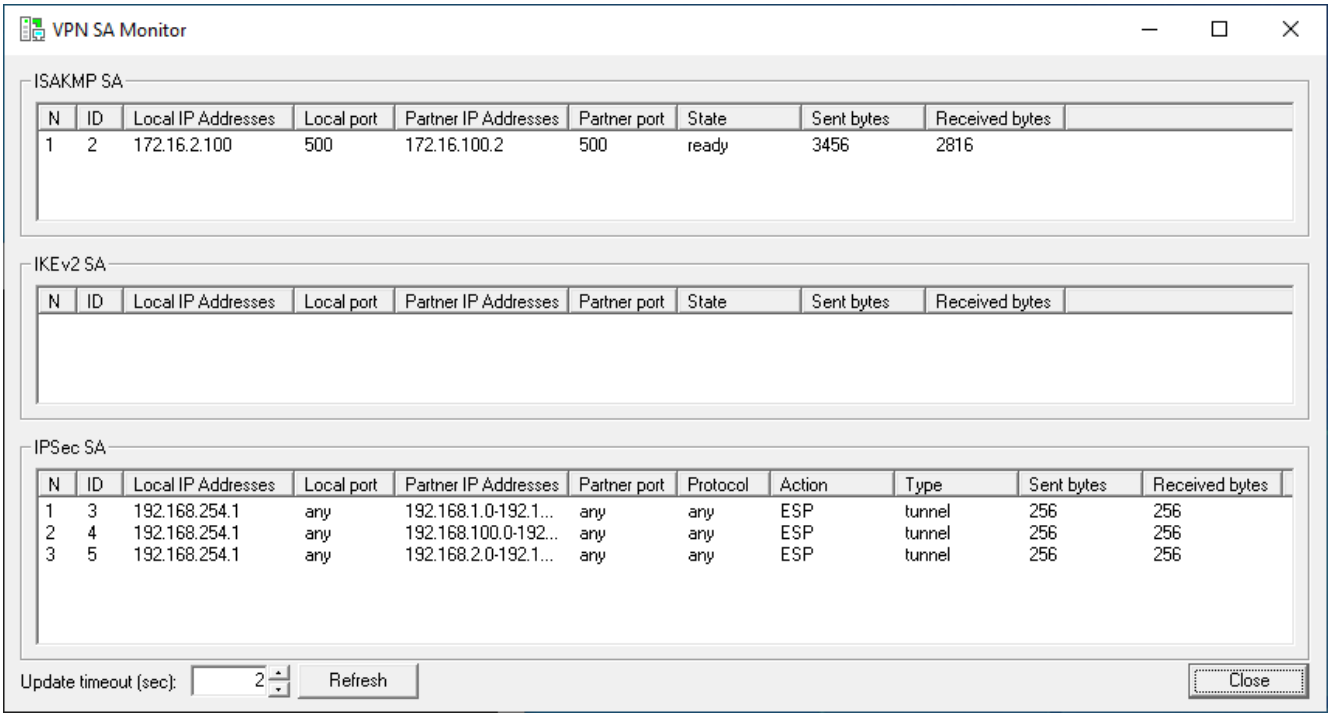


Рисунок 3

7. Проверка, что трафик от Client1 до Host1-b-spoke1 и Host2-b-spoke1 ходит через устройство Traffic-Inspection осуществляется аналогично проверке трафика между Host1-b-spoke1 и Host2-b-spoke1 из пункта 3. Из-за того, что трафик идёт от Client1, IP адрес пакета будет из диапазона 192.168.254.1-192.168.254.254 на Traffic-Inspection:

```
root@Traffic-Inspection:~# tcpdump -i ens192 -c 8 host 192.168.1.100 or host 192.168.2.100 -n
```

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens192, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:18:09.973457 IP 192.168.254.2 > 192.168.1.100: ICMP echo request, id 1, seq 21, length 40
04:18:09.973469 IP 192.168.254.2 > 192.168.1.100: ICMP echo request, id 1, seq 21, length 40
04:18:10.035561 IP 192.168.1.100 > 192.168.254.2: ICMP echo reply, id 1, seq 21, length 40
04:18:10.035580 IP 192.168.1.100 > 192.168.254.2: ICMP echo reply, id 1, seq 21, length 40
04:18:15.091616 IP 192.168.254.2 > 192.168.2.100: ICMP echo request, id 1, seq 22, length 40
04:18:15.091632 IP 192.168.254.2 > 192.168.2.100: ICMP echo request, id 1, seq 22, length 40
04:18:15.101086 IP 192.168.2.100 > 192.168.254.2: ICMP echo reply, id 1, seq 22, length 40
04:18:15.101103 IP 192.168.2.100 > 192.168.254.2: ICMP echo reply, id 1, seq 22, length 40
8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

Видно, что на криптошлюзах Hub1 и Spoke1 и на устройстве Client1 установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log` на криптошлюзах). При необходимости увеличьте уровень логирования с

помощью (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Приложение

Конфигурации криптошлюза Hub1

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 secret 5 $6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOW
Zlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA1aTRi20ElR4wtMXTl0
aaa new-model
!
!
hostname Hub1
enable secret 5 PC9d7N5HlAyLrzuA3qRJvQ==
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp policy 2
 encr gost
 hash gost341112-256-tc26
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto isakmp profile OLD_ALGS
 match identity dn
 set identity mapping
 set policy 2
!
crypto isakmp profile IKE_ALGS
 match identity dn
 set identity mapping
 set policy 1
 set policy 2
!
ip local pool IKECFG_POOL 192.168.254.1 192.168.254.254
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
```

```
!
crypto ipsec transform-set GOST_ENCRYPT_AND_INTEGRITY esp-gost28147-4m-imit
!
object-group network SELF
 remark WAN IP
 host 172.16.100.2
!
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
 permit ip 192.168.254.0 0.0.0.255 192.168.1.0 0.0.0.255
 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
 permit ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
 permit ip 192.168.254.0 0.0.0.255 192.168.2.0 0.0.0.255
 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
!
ip access-list extended IPSEC_ACL_HUB1_AND_CLIENT1
 permit ip 192.168.100.0 0.0.0.255 192.168.254.0 0.0.0.255
 permit ip 192.168.1.0 0.0.0.255 192.168.254.0 0.0.0.255
 permit ip 192.168.2.0 0.0.0.255 192.168.254.0 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_CLIENT1
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile IKE_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 192.168.255.1 255.255.255.0
!
```

```
interface GigabitEthernet0/2
 no ip address
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 172.16.100.1
ip route 192.168.100.0 255.255.255.0 192.168.255.2
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Mon Jun 24 14:11:24 2024

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Mon Jun 24 14:11:24 2024 (user: cscons)"
    Version                      = LSP_5_0
    CRLHandlingMode              = ENABLE
    PreserveIPsecSA              = FALSE
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 172.16.100.1
        ),
        Route(
            Destination = 192.168.100.0/24
            Gateway = 192.168.255.2
        )

)

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
    TCPSessionsMax = 256000
    TCPStrictnessLevel = 3
```

```
    UDPNotEstablishedTimeout = 30
    UDPEstablishedTimeout = 120
    UDPSessionsMax = 65536
)

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

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

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

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

FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 17
        DestinationPort = 4500
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 1
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
```

```
    Filter (
        DestinationIP = 172.16.100.2
        ProtocolID = 50
        Action = PASS
        LogEventID = "FW_INPUT_WAN"
    ),
    Filter (
        DestinationIP = 0.0.0.0/0
        Action = DROP
    )
    IPv6Filters = Filter (
        DestinationIP = ::/0
        Action = DROP
    ))

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

AddressPool IKECFG_POOL
(
    IPAddresses = 192.168.254.1..192.168.254.254
)

IdentityEntry Remote:IKE_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:IKE_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

FilterChain IPv6Default (
    Filters = Filter (
```

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

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

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

AuthMethodGOSTSign GOST:Sign:IKE_ALGS
(
    RemoteID             = Remote:IKE_ALGS
    LocalID              = Local:Cert:IKE_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, crypto:isakmp:policy:2
    AggrModeAuthMethod  = GOST:Sign:IKE_ALGS
    MainModeAuthMethod  = GOST:Sign:IKE_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    ),    (
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 (
        SourceIP = 192.168.254.0/24
        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 (
        SourceIP = 192.168.2.0/24
        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 (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.2.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        SourceIP = 192.168.254.0/24
        DestinationIP = 192.168.2.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
```

```
        SourceIP = 192.168.1.0/24
        DestinationIP = 192.168.2.0/24
        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_CLIENT1"
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.254.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2:DMAP:1 >
        LogEventID = "IPsec:Protect:VPN:2:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENT1"
    ),
    Filter (
        SourceIP = 192.168.1.0/24
        DestinationIP = 192.168.254.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2:DMAP:1 >
        LogEventID = "IPsec:Protect:VPN:2:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENT1"
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.254.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:2:DMAP:1 >
        LogEventID = "IPsec:Protect:VPN:2:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENT1"
    )
)

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
)
```

3. Настройка frr:

```
!
frr version 9.0.2
frr defaults traditional
hostname Hub1
log syslog informational
service integrated-vtysh-config
!
interface eth0
 pbr-policy PBR
exit
!
pbr-map PBR seq 1
```

```
 match src-ip 192.168.1.0/24
 set nexthop 192.168.255.2
exit
!
pbr-map PBR seq 2
match src-ip 192.168.2.0/24
set nexthop 192.168.255.2
exit
!
pbr-map PBR seq 3
match src-ip 192.168.254.0/24
set nexthop 192.168.255.2
exit
!
end
```

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 $6$tHtq8SR6$t3CWE6udI6L/ARr9jQowUYR7wEbOW
Zlx61OvLi7goonOFUYhNSGV49BA.RDGEZ7oKXBA1aTRi20ElR4wtMXTl0
aaa new-model
!
!
hostname Spoke1
enable secret 5 PC9d7N5HlAyLrzuA3qRJvQ==
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group 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
 permit ip 192.168.1.0 0.0.0.255 192.168.254.0 0.0.0.255
```

```
 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
 permit ip 192.168.2.0 0.0.0.255 192.168.100.0 0.0.0.255
 permit ip 192.168.2.0 0.0.0.255 192.168.254.0 0.0.0.255
 permit ip 192.168.2.0 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 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
 no ip address
!
interface GigabitEthernet0/1.11
 ip address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/1.10
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/2
 no ip address
 shutdown
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group ROOTCA http://172.16.101.15/certcrl.crl
 crl download time 60
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
```

```
end
```

2. Конфигурация LSP:

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Mon Jun 24 14:12:47 2024

GlobalParameters(
    Title                       = "This LSP was automatically generated by CSP Converter
at Mon Jun 24 14:12:47 2024 (user: cscons)"
    Version                        = LSP_5_0
    CRLHandlingMode             = ENABLE
    PreserveIPsecSA             = FALSE
)

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

)

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

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

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

FilterChain FilterChain:FW_INPUT_WAN (
    Filters = Filter (
        DestinationIP = 172.16.1.2
```

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

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

IdentityEntry Remote:NEW_ALGS(
    DistinguishedName = USER_SPECIFIC_DATA
)

IdentityEntry Local:Cert:NEW_ALGS(
```

```
    DistinguishedName = USER_SPECIFIC_DATA
)

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

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

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

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

IPsecAction IPsecAction:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 172.16.100.2
        Prefragmentation = ENABLE
    )
    ContainedProposals = ( MAGMA_GOST_ENCRYPT_AND_INTEGRITY:ESP )
    NoDeadConnectionHistory = TRUE
    MSSFixOutgoing = TRUE
    DFHandling = COPY
    IKERule = IKERule:VPN:1
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.1.0/24
```

```
        DestinationIP = 192.168.100.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        SourceIP = 192.168.1.0/24
        DestinationIP = 192.168.254.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        SourceIP = 192.168.1.0/24
        DestinationIP = 192.168.2.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.100.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.254.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_SPOKE1"
    )
)

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 = "GigabitEthernet0/1.11"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

NetworkInterface (
```

```
    LogicalName = "GigabitEthernet0/1.10"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)

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

3. Настройка frr:

```
!
frr version 9.0.2
frr defaults traditional
hostname Spoke1
log syslog informational
service integrated-vtysh-config
!
interface eth1_10
 pbr-policy PBR.10
exit
!
interface eth1_11
 pbr-policy PBR.11
exit
!
pbr-map PBR.10 seq 1
 match src-ip 192.168.1.0/24
 set nexthop 172.16.1.1
exit
!
pbr-map PBR.11 seq 1
 match src-ip 192.168.2.0/24
 set nexthop 172.16.1.1
exit
!
end
```

Конфигурации устройства Client1

1. Конфигурация LSP:

```
GlobalParameters (
    Title = "This LSP was automatically generated by S-Terra Client AdminTool (st) at
2025.11.10 14:32:54"
    Version = LSP_5_0
    CRLHandlingMode = BEST_EFFORT
)
LDAPSettings (
    ResponseTimeout = 200
    HoldConnectTimeout = 60
    DropConnectTimeout = 5
)
IdentityEntry auth_identity_01(
    DistinguishedName *= CertDescription(
        Subject *= COMPLETE,"C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1"
        )
)
CertDescription local_cert_dsc_01(
    Subject *= COMPLETE,"C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Client1"
    Issuer  *=  COMPLETE,"C=RU,L=Zelenograd,O=S-Terra  CSP,OU=RnD,CN=S-Terra  CSP  Test
Root CA"
```

```
    SerialNumber = "2F00005B65EA4888B6F348046B000000005B65"
    FingerprintMD5 = "22BA4AEB2E443BAF5EFEA81FD826A5A6"
)
CertDescription partner_cert_dsc_01(
)
AuthMethodGOSTSign auth_method_01(
    LocalID = auth_identity_01
    LocalCredential = local_cert_dsc_01
    RemoteCredential = partner_cert_dsc_01
    SendRequestMode = AUTO
    SendCertMode = AUTO
)
IKEParameters (
    DefaultPort = 500
    SendRetries = 5
    RetryTimeBase = 1
    RetryTimeMax = 30
    SessionTimeMax = 60
    InitiatorSessionsMax = 30
    ResponderSessionsMax = 20
    BlacklogSessionsMax = 16
    BlacklogSessionsMin = 0
    BlacklogSilentSessions = 4
    BlacklogRelaxTime = 120
    IKECFGPreferDefaultAddress = FALSE
)
IKETransform ike_trf_01(
    LifetimeSeconds = 28800
    CipherAlg  *= "GR341215M-K256-CFB-65526"
    HashAlg *= "GR341112_256TC26-65128"
    GroupID *= VKO2_1B
)
IKETransform ike_trf_02(
    LifetimeSeconds = 28800
    CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
    HashAlg *= "GR341112_256TC26-65128"
    GroupID *= VKO2_1B
)
IKETransform ike_trf_03(
    LifetimeSeconds = 28800
    CipherAlg  *= "GR341215K-K256-CFB-65528"
    HashAlg *= "GR341112_256TC26-65128"
    GroupID *= VKO2_1B
)
ESPTransform esp_trf_01(
    CipherAlg *= "GR341215M-K288-MGM-33"
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_01(
    Transform *=esp_trf_01
)
ESPTransform esp_trf_02(
    CipherAlg *= "G2814789CPRO2-K288-CNTMAC-253"
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_02(
    Transform *=esp_trf_02
)
ESPTransform esp_trf_03(
    CipherAlg *= "GR341215K-K352-MGM-251"
```

```
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_03(
    Transform *=esp_trf_03
)
IKERule ike_rule_with_ikecfg_01(
    DoNotUseDPD = FALSE
    DPDIdleDuration = 60
    DPDResponseDuration = 5
    DPDRetries = 3
    MainModeAuthMethod *= auth_method_01
    Transform *= ike_trf_01,ike_trf_02,ike_trf_03
    IKECFGRequestAddress = TRUE
)
IPsecAction ipsec_action_01(
    PersistentConnection = TRUE
    TunnelingParameters *=
        TunnelEntry(
            PeerAddress = 172.16.100.2
            Assemble = TRUE
            ReRoute = FALSE
            TCPEncapsulation = FALSE
        )
    ContainedProposals *= (esp_proposal_01),(esp_proposal_02),(esp_proposal_03)
    IKERule = ike_rule_with_ikecfg_01
)
FilterChain filter_chain_input(
    Filters *= Filter(
        ProtocolID *= 17
        DestinationPort *= 500
        Action = PASS
        LogEventID = "pass_action_02_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        ProtocolID *= 17
        DestinationPort *= 4500
        Action = PASS
        LogEventID = "pass_action_02_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        SourceIP *= 172.16.100.2
        ProtocolID *= 50
        Action = PASS
        LogEventID = "pass_action_03_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        SourceIP *= 172.16.100.2
        ProtocolID *= 51
        Action = PASS
        LogEventID = "pass_action_03_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        Action = PASS
        LogEventID = "pass_action_04"
    )
)
FilterChain filter_chain_output(
    Filters *= Filter(
        ProtocolID *= 17
        SourcePort *= 500
        Action = PASS
```

```
        LogEventID = "pass_action_05_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        ProtocolID *= 17
        SourcePort *= 4500
        Action = PASS
        LogEventID = "pass_action_05_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 50
        Action = PASS
        LogEventID = "pass_action_06_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 51
        Action = PASS
        LogEventID = "pass_action_06_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        Action = PASS
        LogEventID = "pass_action_07"
    )
)
FilterChain filter_chain_classification_input(
    Filters *= Filter(
        Action = PASS
        LogEventID = "pass_action_08"
    )
)
FilterChain filter_chain_classification_output(
    Filters *= Filter(
        Action = PASS
        LogEventID = "pass_action_09"
    )
)
FilterChain filter_chain_ipsec(
    Filters *= Filter(
        ProtocolID *= 17
        SourcePort *= 500
        Action = PASS
        LogEventID = "pass_action_10_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        ProtocolID *= 17
        SourcePort *= 4500
        Action = PASS
        LogEventID = "pass_action_10_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 50
        Action = PASS
        LogEventID = "pass_action_11_01"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
    ),Filter(
        DestinationIP *= 172.16.100.2
        ProtocolID *= 51
        Action = PASS
        LogEventID = "pass_action_11_02"
        PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
```

```
    ),Filter(
        DestinationIP *= 192.168.1.0/24,192.168.2.0/24,192.168.100.0/24
        Action = PASS
        ExtendedAction *= ipsec<sa=ipsec_action_01>
        LogEventID = "ipsec_action_01"
    ),Filter(
        Action = PASS
        LogEventID = "pass_action_12"
    )
)
NetworkInterface(
    InputFilter = filter_chain_input
    OutputFilter = filter_chain_output
    InputClassification = filter_chain_classification_input
    OutputClassification = filter_chain_classification_output
    IPsecPolicy = filter_chain_ipsec
)
```