Доступ пользователей «С-Терра Клиент» к разделяемому защищенному ресурсу при аутентификации с использованием RADIUS-сервера

Описание стенда

Сценарий иллюстрирует построение защищенного соединения между клиентами «С-Терра Клиент» (устройства Client1 и Client2) и подсетью SN1, защищаемой шлюзом безопасности «С-Терра Шлюз» (устройство GW1). Адрес мобильных клиентов неизвестен заранее – клиенты находятся за динамическим NAT-ом. В ходе построения защищенного соединения клиенты будут получать заранее определенные адреса от RADIUS-сервера (устройство Radius).

В рамках данного сценария для аутентификации партнеры будут использовать сертификаты. В качестве криптопровайдера будет использована криптографическая библиотека, разработанная компанией «С-Терра СиЭсПи». Шлюз безопасности «С-Терра Шлюз» версии 4.2. Клиенты «С-Терра Клиент» версии 4.2.

Параметры защищенного соединения:

Параметры протокола IKE:

- Аутентификация при помощи цифровых сертификатов, алгоритм подписи – ГОСТ Р 34.10-2012;
- 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).

Параметры протокола ESP:

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

Схема стенда (Рисунок 1):

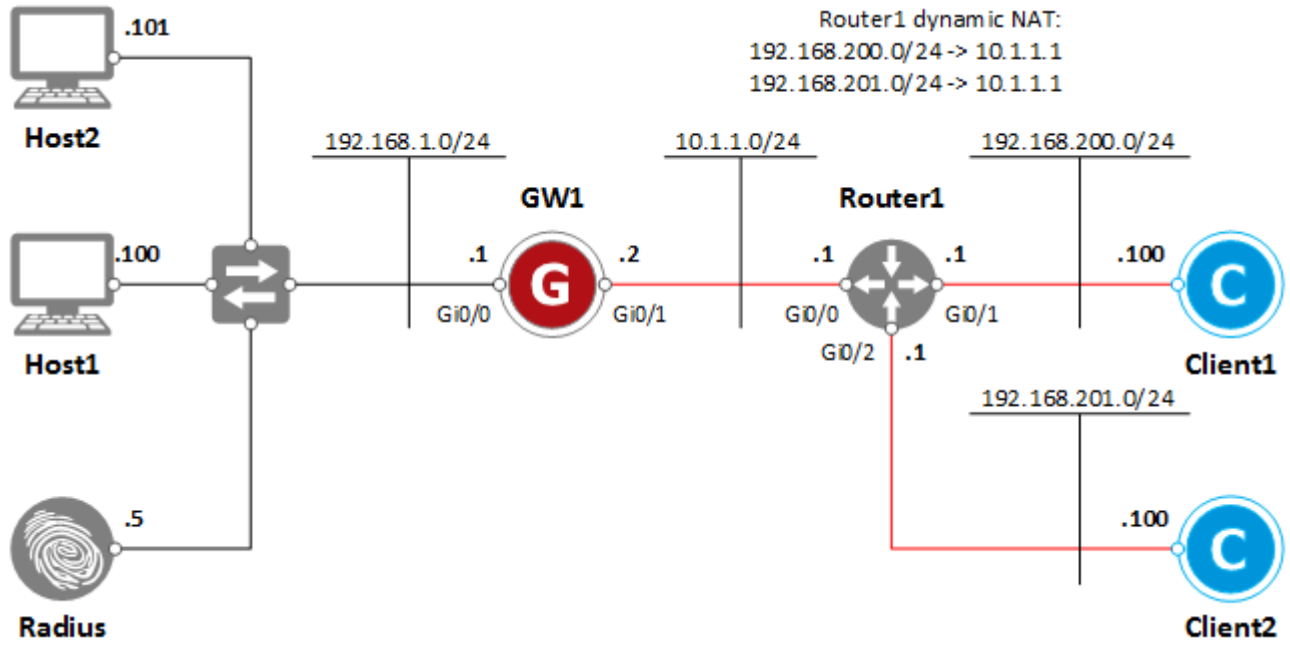

Рисунок 1

Логика работы

В данном сценарии настроено взаимодействие шлюза с RADIUS-сервером и использование XAuth. При построении IKE-сессии от клиента до шлюза на клиенте выводится окно "XAuth request dialog". Введенные в окне данные передаются на RADIUS-сервер для аутентификации. При удачной аутентификации происходит построение IKE и IPsec туннелей между клиентом и шлюзом. При неудачной аутентификации будет выдаваться окно "XAuth ERROR dialog" с сообщением "Extended authentication failed", а также окно для повторной аутентификации.

IKECFG-адрес будет запрашиваться у RADIUS-сервера.

Настройка стенда

Настройка шлюза безопасности GW1

Начальная настройка шлюза в S-Terra administrative console при первом включении состоит из следующих действий:

- Пройдите процедуру аутентификации (пользователь по умолчанию – `administrator`, пароль по умолчанию – `s-terra`).
- Пройдите процедуру инициализации (команда `initialize`).
- Активируйте политику драйвера по умолчанию (команда `run csconf_mgr activate`).
  - Команда `run csconf_mgr activate` применяет текущую политику драйвера. При первичной настройке шлюза применится политика драйвера по умолчанию, при которой прохождение трафика не блокируется.
- Для доступа через SSH установите пароль на пользователя `root` (команда `run passwd`).

Более подробно консоль разграничения доступа S-Terra administrative console описана в документации.

Настройка интерфейсов

1. Перейдите из консоли разграничения доступа (S-Terra Administrative console) в консоль настройки шлюза (cisco-like интерфейс). По умолчанию имя пользователя – `cscons`, пароль – `csp`:

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

```
Password:
...
sterragate#
```

2. Перейдите в режим настройки:

```
sterragate#conf t
```

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

3. В настройках интерфейсов задайте IP-адреса:

```
GW1(config)#interface GigabitEthernet 0/0
GW1(config-if)#ip address 192.168.1.1 255.255.255.0
GW1(config-if)#no shutdown
GW1(config-if)#exit
GW1(config)#interface GigabitEthernet 0/1
GW1(config-if)#ip address 10.1.1.2 255.255.255.0
GW1(config-if)#no shutdown
GW1(config-if)#exit
```

4. Задайте адрес шлюза по умолчанию:

```
GW1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1
```

5. Выйдите из cisco-like интерфейса:

```
sterragate(config)#end
sterragate#exit
```

Дальнейшую настройку можно проводить через SSH подключение.

Важно! Среда передачи в этом случае должна быть доверенной. Описание создания доверенной среды описано в соответствующей инструкции.

Регистрация сертификата УЦ

В данном сценарии для аутентификации используются сертификаты. Для корректной работы необходимо зарегистрировать доверенный сертификат УЦ и локальный сертификат, выданный данным УЦ.

1. Подключитесь по SSH к шлюзу.
2. Установите правильное системное время.

Например:

```
root@sterragate:~# date -s "01/31/2017 15:00"
Tue Jan 31 15:00:00 MSK 2017
```

Данная запись соответствует 31 января 2017 года 15:00.

В данном случае формат даты указывается в виде месяц/день/год (ММ/ДД/ГГГГ).

Для автоматической настройки правильного времени рекомендуется настроить NTP-клиент по соответствующей инструкции.

3. Создайте папку `/certs`:

```
root@sterragate:~# mkdir /certs
```

4. Перенесите доверенный сертификат УЦ на шлюз.

   Способы передачи данных на шлюз описаны в документации.

5. С помощью утилиты `cert_mgr` зарегистрируйте сертификат в базе продукта:

```
root@sterragate:~# cert_mgr import -f /certs/ca.cer -t
1 OK C=RU,L=Moscow,O=S-Terra CSP,OU=Research,CN=Research Root CA
```

Ключ `-t` в данной команде указывает на то, что импортируемый сертификат – доверенный сертификат УЦ.

Регистрация локального сертификата

Для регистрации локального сертификата в базе продукта выполните следующие действия:

1. Сформируйте запрос на сертификат при помощи утилиты `cert_mgr`:

```
root@sterragate:~# cert_mgr create -subj "C=RU,O=S-Terra CSP,OU=Research,CN=GW1" -GOST_R341012_256 -fb64 /home/gw_req
```

   - Ключ `-subj <DN>` задает поля сертификата.
   - Ключ `-GOST_R341012_256` предполагает использование ГОСТ Р 34.10-2012. На УЦ для его поддержки должно быть установлено СКЗИ «КриптоПро CSP» версии 4.0 или новее. При необходимости, есть возможность использовать старый алгоритм (ГОСТ Р 34.10-94), который задается ключом `-GOST_R3410EL`.
   - Ключ `-fb64 <путь до файла>` позволяет сохранить запрос в файл по указанному пути.

2. Передайте полученный запрос сертификата на УЦ. Процедура выдачи сертификата на УЦ по запросу описана в документации.

3. Зарегистрируйте локальный сертификат в базе продукта, применив утилиту `cert_mgr`:

```
root@sterragate:~# cert_mgr import -f /certs/GW1.cer
1 OK C=RU,O=S-Terra CSP,OU=Research,CN=GW1
```

4. Убедитесь, что сертификаты импортированы успешно:

```
root@sterragate:~# cert_mgr show
Found 2 certificates. No CRLs found.
1 Status: trusted C=RU,L=Moscow,O=S-Terra CSP,OU=Research,CN=Research Root CA
2 Status: local   C=RU,O=S-Terra CSP,OU=Research,CN=GW1
```

Создание политики безопасности

После регистрации сертификатов необходимо создать политику безопасности для шлюза GW1

1. Для входа в консоль запустите `cs_console`:

```
root@sterragate:~# cs_console
sterragate>enable
Password:
```

Пароль по умолчанию – `csp`.

Важно! Пароль по умолчанию необходимо сменить.

2. Перейдите в режим настройки:

```
sterragate#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

3. Смените пароль по умолчанию:

```
sterragate(config)#username cscons password <пароль>
```

4. Смените название шлюза:

```
sterragate(config)#hostname GW1
```

5. Задайте тип идентификации:

```
GW1(config)#crypto isakmp identity dn
```

В данном сценарии для идентификации будет использоваться поле DN сертификата.

6. Задайте параметры DPD (dead peer detection)

```
GW1(config)#crypto isakmp keepalive 10 2
GW1(config)#crypto isakmp keepalive retry-count 5
```

Если в течение 10 секунд отсутствует входящий трафик в IPsec туннеле, то с интервалом в 2 секунды посылается 5 keepalive-пакетов в IKE туннеле, чтобы удостовериться в работоспособности туннеля. Если партнер не отвечает на keepalive-пакеты, то существующий IKE туннель переходит в состояние disabled, а связанные с ним IPsec туннели удаляются. В случае наличия исходящего трафика происходит попытка создать новый IKE туннель.

7. Задайте параметры для IKE:

```
GW1(config)#crypto isakmp policy 1
GW1(config-isakmp)#authentication gost-sig
GW1(config-isakmp)#encr gost
GW1(config-isakmp)#hash gost341112-256-tc26
GW1(config-isakmp)#group vko2
GW1(config-isakmp)#exit
```

8. Задайте параметры для IPsec:

```
GW1(config)#crypto ipsec transform-set TSET esp-gost28147-4m-imit
GW1(cfg-crypto-trans)#exit
```

9. Опишите трафик, который планируется защищать. Для этого создайте расширенный список доступа:

```
GW1(config)#ip access-list extended LIST
GW1(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 any
GW1(config-ext-nacl)#exit
```

10. Создайте список доступа для фильтрации трафика:

```
GW1(config)#ip access-list extended FILTER_LIST
GW1(config-ext-nacl)#permit ip 192.168.11.0 0.0.0.255 host 192.168.1.100
GW1(config-ext-nacl)#permit ip 192.168.12.0 0.0.0.255 host 192.168.1.101
GW1(config-ext-nacl)#permit ip host 192.168.1.1 host 192.168.1.5
GW1(config-ext-nacl)#exit
```

11. Настройте взаимодействие с RADIUS-сервером:

```
GW1(config)#aaa new-model
GW1(config)#aaa authentication login RADIUS1 group radius
GW1(config)#radius-server host 192.168.1.5
GW1(config)#radius-server key secret1
```

где:

- `RADIUS1` – наименование списка аутентификации для использования в крипто-карте;
- `secret1` – пароль для аутентификации шлюза на RADIUS-сервере;
- `192.168.1.5` – IP-адрес RADIUS-сервера;

12. Создайте динамическую крипто-карту:

```
GW1(config)#crypto dynamic-map DMAP 1
GW1(config-crypto-map)#match address LIST
GW1(config-crypto-map)#set transform-set TSET
GW1(config-crypto-map)#reverse-route
GW1(config-crypto-map)#set client authentication list RADIUS1
GW1(config-crypto-map)#set client authentication xauth
GW1(config-crypto-map)#set client username interactive
GW1(config-crypto-map)#exit
```

где:

- `set client authentication list <наименование списка>` – ссылка на список аутентификации;
- `set client authentication xauth` – включение поддержки XAuth;
- `set client username interactive` – на RADIUS-сервер будут передаваться данные, интерактивно введенные в процессе проведения XAuth-сессии.

13. Привяжите динамическую карту к статической:

```
GW1(config)#crypto map CMAP 1 ipsec-isakmp dynamic DMAP
```

14. Для фильтрации трафика, привяжите к интерфейсу список доступа:

```
GW1(config)#interface GigabitEthernet 0/0
GW1(config-if)#ip access-group FILTER_LIST out
GW1(config-if)#exit
```

15. Привяжите крипто-карту к интерфейсу, на котором будет туннель:

```
GW1(config)#interface GigabitEthernet 0/1
GW1(config-if)#crypto map CMAP
GW1(config-if)#exit
```

16. Настройте получение списка отозванных сертификатов (CRL) по HTTP:

```
GW1(config)#crypto pki trustpoint s-terra_technological_trustpoint
GW1(ca-trustpoint)#crl download group GROUP http://10.0.221.170/certsrv/certcrl.crl
GW1(ca-trustpoint)#exit
```

Предполагается, что CRL выкладывается на общедоступное место (доступ к которому обеспечен без использования IPsec) для всех шлюзов. При указании имени домена, вместо IP-адреса, необходимо настроить адрес DNS-сервера в системном файле `/etc/resolv.conf`.

Также необходимо учитывать, что у CRL есть срок действия и нужно обеспечивать своевременное их обновление в данном общедоступном месте.

По умолчанию CRL будет запрашиваться раз в сутки (раз в 1440 минут), для изменения интервала запросов можно воспользоваться командой `crl download time <интервал в минутах>`.

При необходимости отключения CRL (не рекомендуется отключать CRL) воспользуйтесь командой `revocation-check none`.

17. Настройка устройства GW1 в cisco-like консоли завершена. При выходе из конфигурационного режима происходит загрузка конфигурации:

```
GW1(config)#end
GW1#exit
```

В приложении представлен текст cisco-like конфигурации и текст LSP конфигурации для шлюза GW1.

Настройка клиента Client1

Настройка клиента состоит из следующих этапов:

- установка приложения AdminTool на компьютере администратора;
- получение запросов на сертификаты (на основе новых контейнеров с ключами) на компьютере администратора;

- перенос запросов на УЦ, получение сертификатов из запросов, перенос сертификатов на компьютер администратора;
- формирование установочного пакета для целевого клиентского компьютера с помощью AdminTool;
- перенос пакета и его установка на целевом клиентском компьютере.

Предполагается, что формирование установочного пакета 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.

В данном сценарии не описывается процесс установки AdminTool, а также процесс выпуска и переноса сертификатов.

Более подробное описание программы AdminTool представлено в документации.

Процедура выдачи сертификата по запросу описана в документации.

1. Создайте запрос на сертификат с помощью утилиты `excont_mgr`, входящей в состав AdminTool

   1.1. Запустите командую строку от имени Администратора

   1.2. Перейдите в директорию AdminTool:

   ```
   cd "C:\Program Files (x86)\S-Terra Client AdminTool st"
   ```

   1.3. Создайте запрос на сертификат. При этом будет создан новый контейнер с ключами.

   ```
   excont_mgr.exe create_req -subj "C=RU,O=S-Terra CSP,OU=Research,CN=Client1" -GOST_R341012_256 -kc Client1 -kcp 1234 -fo C:\Client1.req
   ```

   - `create_req` – создать запрос на сертификат;
   - `-subj <DN>` – указать поля сертификата;
   - `-GOST_R341012_256` – формат запроса;
   - `-kc <имя контейнера>` – задать имя контейнера;
   - `-kcp <PIN>` – задать пароль на контейнер;
   - `-fo <путь до файла>` – указать путь, по которому будет сохранен запрос на сертификат.

   Ключ `-GOST_R341012_256` предполагает использование ГОСТ 2012. На УЦ для его поддержки должно быть установлено СКЗИ «КриптоПро CSP» версии 4.0 или новее. При необходимости, можно воспользоваться более старым ключом `-GOST_R3410EL`.

   Более полное описание утилиты `excont_mgr` представлено в документации.

2. Созданный запрос перенесите на УЦ и получите на его основе пользовательский сертификат. Также получите сертификат данного УЦ. Перенесите сертификат УЦ и пользовательский сертификат на компьютер администратора.

3. Создайте установочный пакет для AdminHost.

   3.1. Запустите установленное приложение AdminTool.

   3.2. Во вкладке **Auth** выполните следующие действия (Рисунок 2):

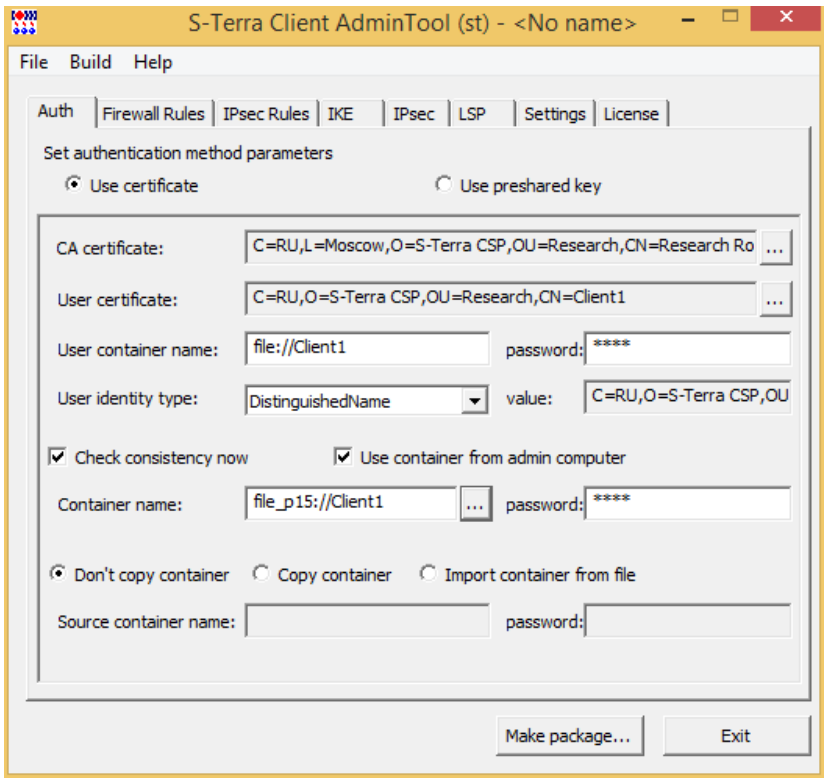

Рисунок 2

3.2.1 В данном сценарии используется метод аутентификации на сертификатах – переключатель установлен на **Use certificate** по умолчанию.

3.2.2 Укажите путь к сертификату УЦ и пользовательскому сертификату.

3.2.3 Отметьте флаг **Check consistency now** и нажмите кнопку **"..."**, где выберите созданный ранее контейнер. Если при создании запроса на сертификат указывался пароль на контейнер, введите его в поле **password**.

3.2.4 Отметьте флаг **Use container from admin computer**. Указанный в п.2.2.3 контейнер будет помещен в установочный пакет.

3.2.5 Задайте имя контейнера в поле **User container name.** В данном случае указано – `file://Client1`. Данное поле указывает по какому пути искать контейнер при работе. Так как отмечен флаг **User container name,** то контейнер будет скопирован по указанному пути.

3.2.6 В поле **User identity type** необходимо использовать **DistiguishedName** (выбрано по умолчанию).

3.3. Во вкладке **Firewall Rules** (Рисунок 3) можно настроить правила фильтрации трафика. В данном сценарии оставьте настройки по умолчанию - разрешать весь трафик.

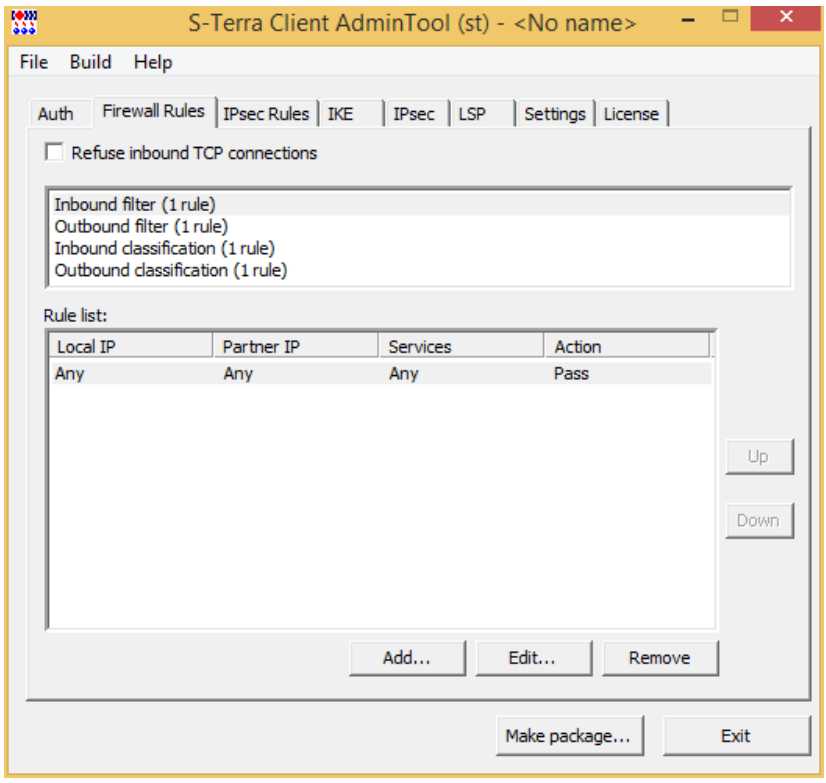

Рисунок 3

3.4. Вкладка **IPsec Rules**:

3.4.1 Добавьте правило для трафика, подлежащего шифрованию. Нажмите кнопку **Add** и в открывшемся окне **Add Rule** проведите следующие настройки (Рисунок 4):

3.4.1.1 В разделе **Partner IP Address** укажите адрес подсети SN1 – 192.168.1.0 и ее маску – 255.255.255.0.

3.4.1.2 В разделе **Action** выберите из списка **Protect using IPsec**, укажите адрес шлюза GW1 – 10.1.1.2 и отметьте флаг **Request IKECFG address**.

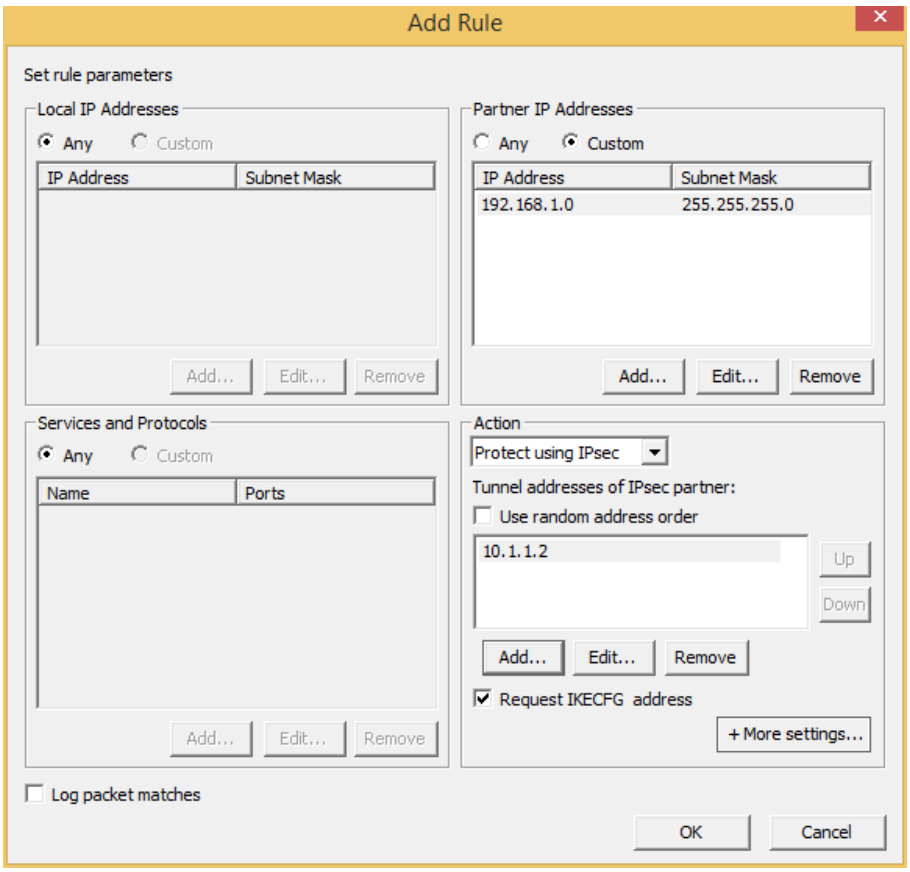

Рисунок 4

3.4.2 Добавленное правило поднимите вверх (Рисунок 5).

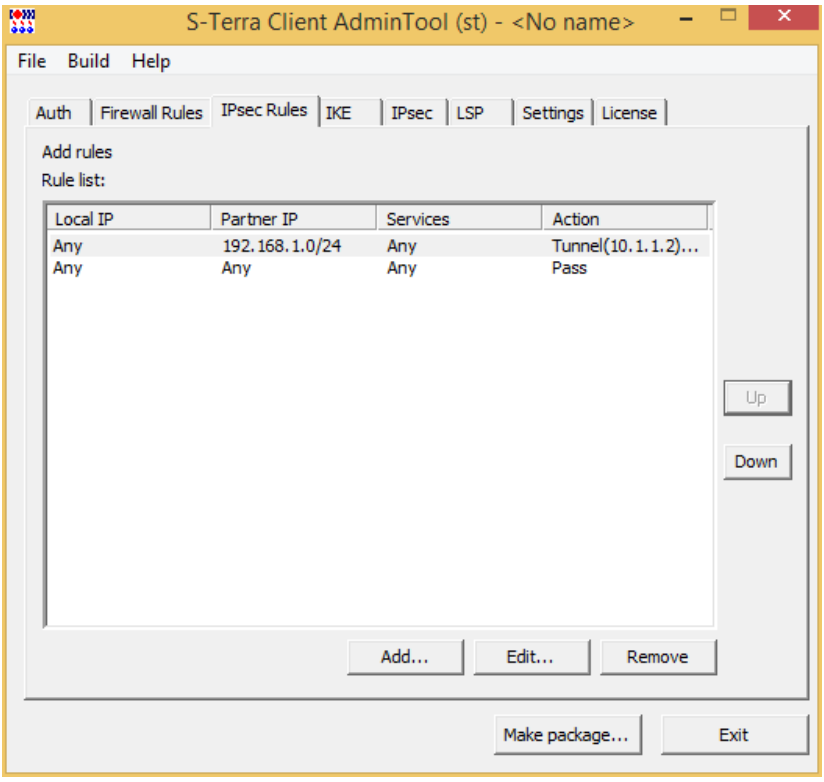

Рисунок 5

3.5. Во вкладке **IKE** по умолчанию установлены нужные настройки (Рисунок 6). При необходимости можно поднять в приоритете используемое правило.

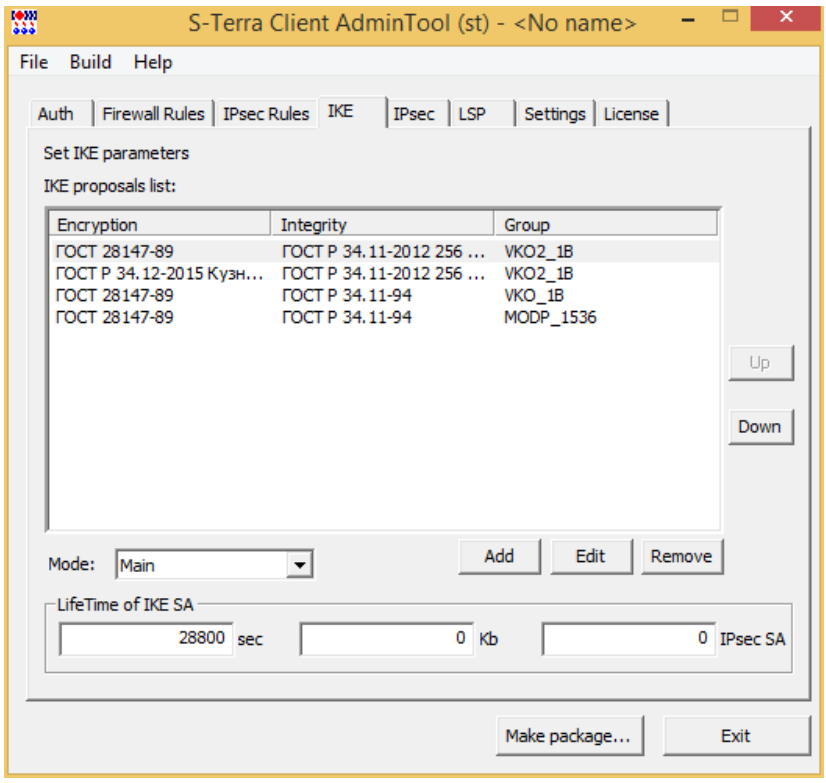

Рисунок 6

3.6. Во вкладке **IPsec** по умолчанию установлены нужные настройки (Рисунок 7). При необходимости можно поднять в приоритете используемое правило.

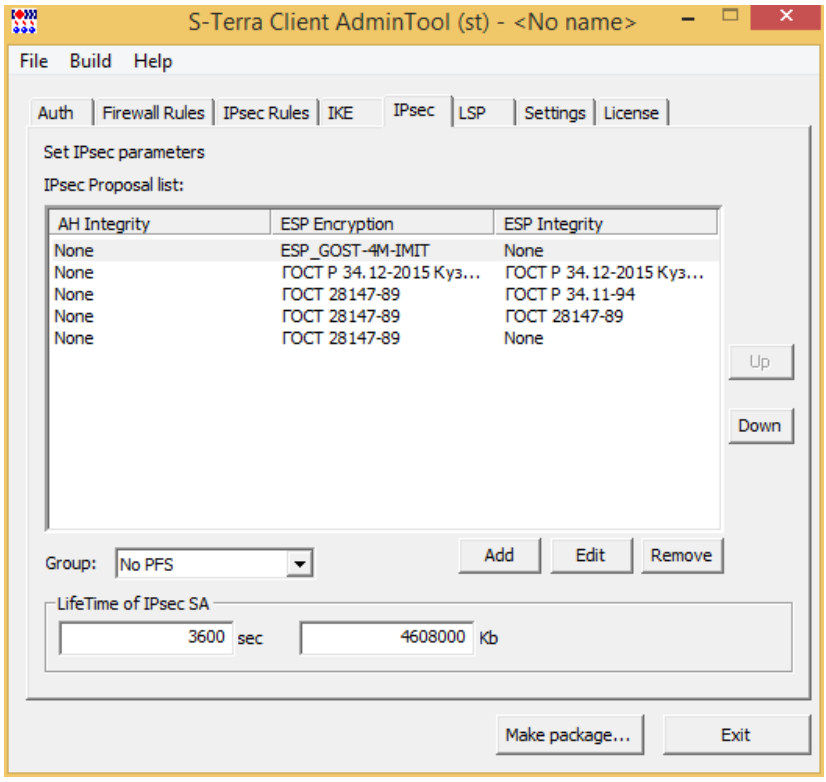

Рисунок 7

3.7. Во вкладке **LSP** можно просмотреть получившуюся политику безопасности.

3.8. Во вкладке **License** введите лицензию на продукт «С-Терра Клиент» версии 4.2.

3.9. Сохраните файл созданного проекта, на тот случай, если захотите в будущем сделать похожий клиентский пакет. Для этого в меню **File** выберите **Save project**.

3.10. Создайте установочный exe-файл для «С-Терра Клиент», нажав кнопку **Make package...**.

4. Установка на целевой клиентский компьютер.

4.1. Установите на клиентском компьютере полученный exe-файл.

4.2. В области уведомлений появится иконка «С-Терра Клиент» (Рисунок 8). Для начала работы необходимо пройти процедуру аутентификации (Рисунок 9). Имя пользователя по умолчанию – `user`. Пароль по умолчанию отсутствует, в дальнейшем его можно установить.

Рисунок 8

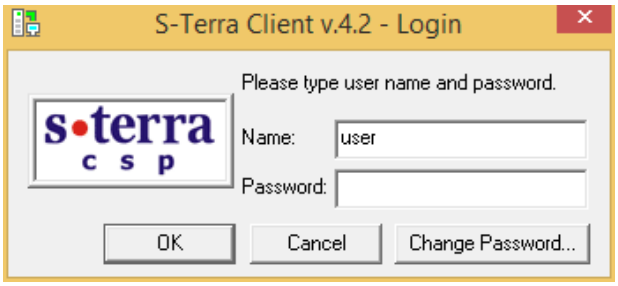

Рисунок 9

4.3. Так как на шлюзе было настроено взаимодействие с RADIUS-сервером, XAuth и интерактивный ввод, то при установлении сессии будет появляться дополнительное окно аутентификации (Рисунок 10). Логин и пароль должны соответствовать данным, настроенным на RADIUS-сервере. При успешной аутентификации будет построены IKE и IPsec туннели.

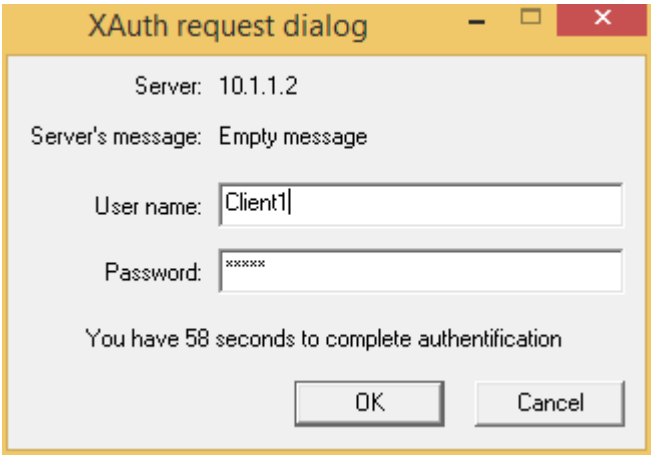

Рисунок 10

В приложении представлен текст LSP конфигурации для клиента Client1.

Настройка клиента Client2

Настройка клиента Client2 происходит аналогично настройке клиента Client1.

В приложении представлен текст LSP конфигурации для клиента Client2.

Настройка устройства Router1

На устройстве Router1 необходимо настроить динамический NAT, который будет преобразовывать IP-адреса из подсети 192.168.200.0/24 во внешний IP-адрес 10.1.1.1, а также IP-адреса из подсети 192.168.201.0/24 во внешний IP-адрес 10.1.1.1.

Настройка устройства Radius

Пример настройки RADIUS-сервера на базе freeradius 2.1.12.

1. Настройте аутентификацию со шлюзом:

```
root@Radius:~# vi /etc/freeradius/clients.conf
```

```
client GW {
ipaddr = 192.168.1.1
secret = secret1
}
```

где:

- `192.168.1.1` – IP-адрес шлюза GW1;
- `secret1` – пароль для аутентификации шлюза GW1/

2. Пропишите данные для аутентификации клиентов:

```
root@Radius:~# vi /etc/freeradius/users
```

```
Client1 Cleartext-Password := Pass1

Framed-IP-Address = 192.168.11.1
Client2 Cleartext-Password := Pass2

Framed-IP-Address = 192.168.12.1
```

где:

- `Client1` – имя пользователя клиента;
- `Pass1` – пароль клиента;
- `192.168.11.1` – Framed-IP-Address, который будет передаваться на клиент в качестве адреса IKECFG-интерфейса.

3. По умолчанию для сервиса `freeradius` будет использоваться порт 1812. Необходимо сменить его на 1645. Для этого необходимо изменить значение переменной `port` структуры `listen` в файле `/etc/freeradius/radiusd.conf`. Для смены порта выполните следующую команду:

```
root@Radius:~# sed -i '0,/\tport = 0/{s/\tport = 0/\tport = 1645/}' /etc/freeradius/radiusd.conf
```

4. После произведенных настроек, необходимо перезапустить сервис freeradius:

```
root@Radius:~# service freeradius restart
```

4.1. Для отладки работоспособности взаимодействия с RADIUS-сервером можно воспользоваться debug-режимом, запустив сервис с флагом -X:

```
root@Radius:~# freeradius -X
```

Настройка устройства Host1

На устройстве Host1 задайте IP-адрес, а в качестве шлюза по умолчанию укажите адрес внутреннего интерфейса шлюза безопасности GW1 – 192.168.1.1.

Проверка работоспособности стенда

После того, как настройка всех устройств завершена, проверьте работоспособность стенда.

На устройстве Client1 выполните команду ping IP-адреса Host1:

```
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<1мс TTL=62
Ответ от 192.168.1.100: число байт=32 время<1мс TTL=62
Ответ от 192.168.1.100: число байт=32 время<1мс TTL=62
Ответ от 192.168.1.100: число байт=32 время<1мс TTL=62

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

Информацию о соединении можно увидеть в программе VPN SA Monitor:

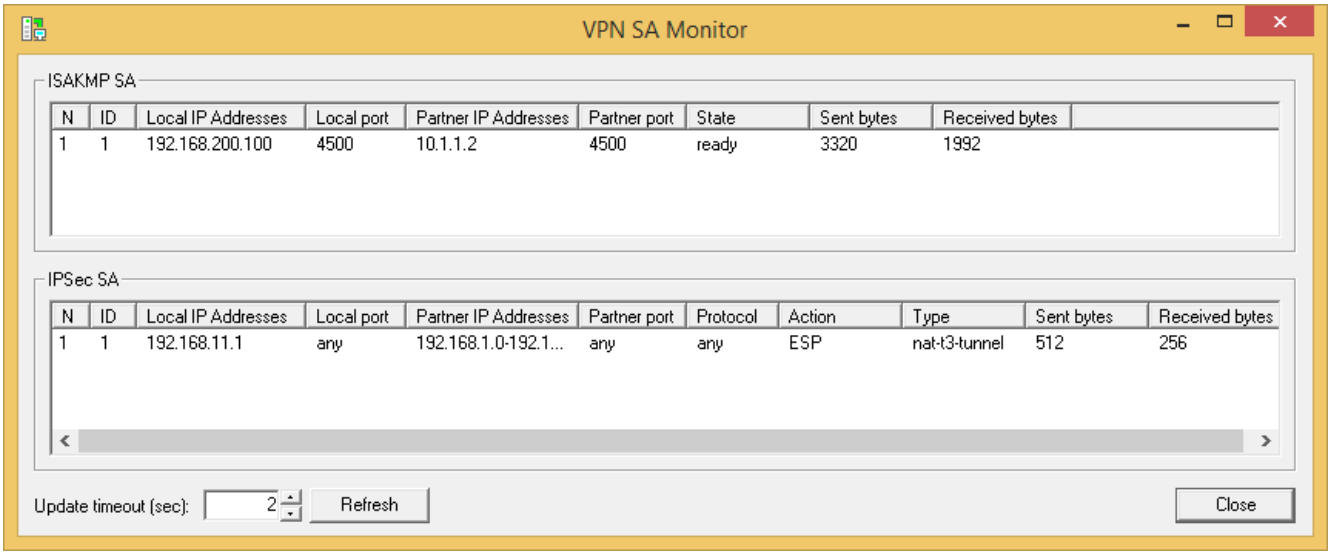

Рисунок 11

На устройстве Client1 выполните команду ping IP-адреса Host2:

```
C:\Users\Administrator> ping 192.168.1.101
```

```
Обмен пакетами с 192.168.1.101 по с 32 байтами данных:
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.

Статистика Ping для 192.168.1.101:
    Пакетов: отправлено = 4, получено = 0, потеряно = 4
    (100% потерь)
```

На шлюзе GW1 данные пакеты будут отбрасываться.

Аналогично на клиенте Client2 не должен проходить ping до 192.168.1.100, но должен проходить ping до 192.168.1.101.

Выявление ошибок

Чтобы разобраться на каком этапе возникла ошибка можно воспользоваться руководством, которое представлено на портале документации: http://doc.s-terra.ru/rh_output/4.2/Scenarios/output/mergedProjects/1main/ver_4_2_troubleshooting_guide.pdf.

Приложение

Текст cisco-like конфигурации для шлюза GW1

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp keepalive 10
crypto isakmp keepalive retry-count 5
username cscons privilege 15 password 0 csp
aaa new-model
aaa authentication login RADIUS1 group radius
!
radius-server host 192.168.1.5
radius-server key secret1
!
hostname GW1
enable password csp
!
!
!
!
!
crypto isakmp policy 1
 encr gost
 hash gost341112-256-tc26
 authentication gost-sig
 group vko2
!
crypto ipsec transform-set TSET esp-gost28147-4m-imit
!
ip access-list extended LIST
 permit ip 192.168.1.0 0.0.0.255 any
!
ip access-list extended FILTER_LIST
 permit ip 192.168.11.0 0.0.0.255 host 192.168.1.100
 permit ip 192.168.12.0 0.0.0.255 host 192.168.1.101
 permit ip host 192.168.1.1 host 192.168.1.5
!
!
crypto dynamic-map DMAP 1
 match address LIST
 set transform-set TSET
 reverse-route
 set client authentication list RADIUS1
 set client authentication xauth
 set client username interactive
!
crypto map CMAP 1 ipsec-isakmp dynamic DMAP
!
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group FILTER_LIST out
!
interface GigabitEthernet0/1
 ip address 10.1.1.2 255.255.255.0
 crypto map CMAP
!
interface GigabitEthernet0/2
 no ip address
 shutdown
!
```

```
interface GigabitEthernet0/3
 no ip address
 shutdown
!
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
 crl download group GROUP http://10.0.221.170/certsrv/certcrl.crl
crypto pki certificate chain s-terra_technological_trustpoint
certificate 2A7932A877BCEC9E431A5C2CDC26E2A3
30820212308201C1A00302010202102A7932A877BCEC9E431A5C2CDC26E2A330
0806062A85030202033062310B30090603550406130252553 10F300D06035504
0713064D6F73636F7731143012060355040A130B532D5465727261204353503 1
11300F060355040B13085265736561726368311930170603550403131052657 3
656172636820526F6F74204341301E170D3136313232323230353532335A170D
3336313232323231303435325A3062310B30090603550406130252553 10F300D
060355040713064D6F73636F7731143012060355040A130B532D546572726120
4353503111300F060355040B13085265736561726368311930170603550403 13
105265736561726368205 26F6F742043413063301C06062A8503020213301206
072A85030202301060 72A850302021E0103430004402204 0A6B8407256B40D5
A491C45AD7F792B18476B76405DA7F80AD44AA67D85CC1C9EF68F6EB38F1C31C
873E374C348245A862A3C3046C8CF8E1450C0598B985A351304F300B0603551D
0F040403020186300F0603551D130101FF040530030101FF301D0603551D0E04
16041414EA3F573705DE01C3E80F23BF64A13A96CD1FB4301006092B06010401
8237150104030201003008060 62A8503020203034100F8B18E0C87FE16CD93CA
7829F4FA981112A57201251CF36E3D10C95BD53D702E5EFB4B4ED111248AC475
389FAB5087EC9CACB597010B88E5F638C8CF8A4F6767

quit
!
end
```

Текст LSP конфигурации для шлюза GW1

```
#   This is automatically generated LSP
#
#   Conversion Date/Time:   Thu Jul  6 18:17:14 2017

GlobalParameters(
    Title                       = "This LSP was automatically generated by CSP Converter
at Thu Jul  6 18:17:14 2017"
    Version                       = LSP_4_2
    CRLHandlingMode               = ENABLE
    PreserveIPsecSA               = FALSE
)

IKEParameters(
    FragmentSize = 0
)

RoutingTable(
    Routes =
        Route(
            Destination = 0.0.0.0/0
            Gateway = 10.1.1.1
        )
)

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
```

```
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
)

AAASettings(
    RadiusServer = 192.168.1.5
    Secret = "cs_radius_server_key__"
    Retries = 4
    ResponseTimeout = 5
)

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

ESPProposal TSET:ESP
(
    Transform* = ESPTransform
    (
        CipherAlg*          = "G2814789CPRO2-K288-CNTMAC-253"
        LifetimeSeconds     = 3600
        LifetimeKilobytes   = 4608000
    )
)

FilterChain FilterChain:FILTER_LIST (
    Filters = Filter (
        SourceIP = 192.168.11.0/24
        DestinationIP = 192.168.1.100
        Action = PASS
        LogEventID = "FILTER_LIST"
    ),
    Filter (
        SourceIP = 192.168.12.0/24
        DestinationIP = 192.168.1.101
        Action = PASS
        LogEventID = "FILTER_LIST"
    ),
    Filter (
        SourceIP = 192.168.1.1
        DestinationIP = 192.168.1.5
        Action = PASS
        LogEventID = "FILTER_LIST"
    ),
    Filter (
        Action = DROP
    )
)

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

IKERule IKERule:CMAP:1:DMAP:1
(
    Transform = crypto:isakmp:policy:1
```

```
    AggrModeAuthMethod  = GOST:Sign
    MainModeAuthMethod  = GOST:Sign
    DPDIdleDuration     = 10
    DPDResponseDuration = 2
    DPDRetries          = 5
    XAuthServerEnabled  = TRUE
    AAAUserName         = INTERACTIVE
    Priority            = 100
)

IPsecAction IPsecAction:CMAP:1:DMAP:1
(
    TunnelingParameters = TunnelEntry(
        DFHandling=COPY
        Assemble=TRUE
    )
    ContainedProposals = ( TSET:ESP )
    ReverseRoute = TRUE
    IKERule = IKERule:CMAP:1:DMAP:1
)

FilterChain IPsecPolicy:CMAP (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:CMAP:1:DMAP:1 >
        LogEventID = "IPsec:Protect:CMAP:1:DMAP:1:LIST"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    OutputFilter = FilterChain:FILTER_LIST
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    IPsecPolicy = IPsecPolicy:CMAP
)
```

Текст LSP конфигурации для клиента Client1

```
GlobalParameters (
       Title = "This LSP was automatically generated by S-Terra Client AdminTool (st)
at 2017.07.05 16:40:16"
       Version = LSP_4_2
       CRLHandlingMode = BEST_EFFORT
)
LDAPSettings (
       ResponseTimeout = 200
       HoldConnectTimeout = 60
       DropConnectTimeout = 5
)
IdentityEntry auth_identity_01(
       DistinguishedName *= CertDescription(
              Subject *= COMPLETE,"C=RU,O=S-Terra CSP,OU=Research,CN=Client1"
              )
)
CertDescription local_cert_dsc_01(
```

```
      Subject *= COMPLETE,"C=RU,O=S-Terra CSP,OU=Research,CN=Client1"
      Issuer  *=  COMPLETE,"C=RU,L=Moscow,O=S-Terra  CSP,OU=Research,CN=Research  Root
CA"
      SerialNumber = "5600000306D0D055E6ABA08001000000000306"
      FingerprintMD5 = "CE63DB090D2145CBFCA9F904A7944EF9"
)
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  *= "G2814789CPRO1-K256-CBC-65534"
      HashAlg *= "GR341112_256TC26-65128"
      GroupID *= VKO2_1B
)
IKETransform ike_trf_02(
      LifetimeSeconds = 28800
      CipherAlg  *= "GR341215K-K256-CFB-65528"
      HashAlg *= "GR341112_256TC26-65128"
      GroupID *= VKO2_1B
)
IKETransform ike_trf_03(
      LifetimeSeconds = 28800
      CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
      HashAlg *= "GR341194CPRO1-65534"
      GroupID *= VKO_1B
)
IKETransform ike_trf_04(
      LifetimeSeconds = 28800
      CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
      HashAlg *= "GR341194CPRO1-65534"
      GroupID *= MODP_1536
)
ESPTransform esp_trf_01(
      CipherAlg *= "G2814789CPRO2-K288-CNTMAC-253"
      LifetimeSeconds = 3600
      LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_01(
      Transform *=esp_trf_01
)
ESPTransform esp_trf_02(
      CipherAlg *= "GR341215K-K256-CFB-248"
      IntegrityAlg *= "GR341215K-K256-MAC-65529"
      LifetimeSeconds = 3600
      LifetimeKilobytes = 4608000
```

```
)
ESPProposal esp_proposal_02(
	Transform *=esp_trf_02
)
ESPTransform esp_trf_03(
	CipherAlg *= "G2814789CPRO1-K256-CBC-254"
	IntegrityAlg *= "GR341194CPRO1-H96-HMAC-65534"
	LifetimeSeconds = 3600
	LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_03(
	Transform *=esp_trf_03
)
ESPTransform esp_trf_04(
	CipherAlg *= "G2814789CPRO1-K256-CBC-254"
	IntegrityAlg *= "G2814789CPRO1-K256-MAC-65535"
	LifetimeSeconds = 3600
	LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_04(
	Transform *=esp_trf_04
)
ESPTransform esp_trf_05(
	CipherAlg *= "G2814789CPRO1-K256-CBC-254"
	LifetimeSeconds = 3600
	LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_05(
	Transform *=esp_trf_05
)
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,ike_trf_04
	IKECFGRequestAddress = TRUE
)
IPsecAction ipsec_action_01(
	PersistentConnection = TRUE
	TunnelingParameters *=
		TunnelEntry(
			PeerAddress = 10.1.1.2
			Assemble = TRUE
			ReRoute = FALSE
			TCPEncapsulation = FALSE
		)
	ContainedProposals                                                    *=
(esp_proposal_01),(esp_proposal_02),(esp_proposal_03),(esp_proposal_04),(esp_proposal
_05)
	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 *= 10.1.1.2
            ProtocolID *= 50
            Action = PASS
            LogEventID = "pass_action_03_01"
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            SourceIP *= 10.1.1.2
            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 *= 10.1.1.2
            ProtocolID *= 50
            Action = PASS
            LogEventID = "pass_action_06_01"
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            DestinationIP *= 10.1.1.2
            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 *= 10.1.1.2
            ProtocolID *= 50
            Action = PASS
            LogEventID = "pass_action_11_01"
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            DestinationIP *= 10.1.1.2
            ProtocolID *= 51
            Action = PASS
            LogEventID = "pass_action_11_02"
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            DestinationIP *= 192.168.1.0/24
            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
)
```

Текст LSP конфигурации для клиента Client2

```
GlobalParameters (
      Title = "This LSP was automatically generated by S-Terra Client AdminTool (st)
at 2017.07.05 17:18:36"
      Version = LSP_4_2
      CRLHandlingMode = BEST_EFFORT
)
LDAPSettings (
      ResponseTimeout = 200
      HoldConnectTimeout = 60
      DropConnectTimeout = 5
)
IdentityEntry auth_identity_01(
      DistinguishedName *= CertDescription(
            Subject *= COMPLETE,"C=RU,O=S-Terra CSP,OU=Research,CN=Client2"
            )
)
CertDescription local_cert_dsc_01(
      Subject *= COMPLETE,"C=RU,O=S-Terra CSP,OU=Research,CN=Client2"
      Issuer  *=  COMPLETE,"C=RU,L=Moscow,O=S-Terra  CSP,OU=Research,CN=Research  Root
CA"
      SerialNumber = "5600000307307DACF5395AB2FE000000000307"
      FingerprintMD5 = "9ABACCC8489E67215A6AC39257B8704E"
)
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  *= "G2814789CPRO1-K256-CBC-65534"
      HashAlg *= "GR341112_256TC26-65128"
      GroupID *= VKO2_1B
)
IKETransform ike_trf_02(
      LifetimeSeconds = 28800
      CipherAlg  *= "GR341215K-K256-CFB-65528"
      HashAlg *= "GR341112_256TC26-65128"
      GroupID *= VKO2_1B
)
IKETransform ike_trf_03(
      LifetimeSeconds = 28800
      CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
      HashAlg *= "GR341194CPRO1-65534"
      GroupID *= VKO_1B
)
IKETransform ike_trf_04(
      LifetimeSeconds = 28800
      CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
      HashAlg *= "GR341194CPRO1-65534"
      GroupID *= MODP_1536
)
ESPTransform esp_trf_01(
      CipherAlg *= "G2814789CPRO2-K288-CNTMAC-253"
      LifetimeSeconds = 3600
      LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_01(
      Transform *=esp_trf_01
)
ESPTransform esp_trf_02(
      CipherAlg *= "GR341215K-K256-CFB-248"
      IntegrityAlg *= "GR341215K-K256-MAC-65529"
      LifetimeSeconds = 3600
      LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_02(
      Transform *=esp_trf_02
)
ESPTransform esp_trf_03(
      CipherAlg *= "G2814789CPRO1-K256-CBC-254"
      IntegrityAlg *= "GR341194CPRO1-H96-HMAC-65534"
      LifetimeSeconds = 3600
      LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_03(
```

```
	Transform *=esp_trf_03
)
ESPTransform esp_trf_04(
	CipherAlg *= "G2814789CPRO1-K256-CBC-254"
	IntegrityAlg *= "G2814789CPRO1-K256-MAC-65535"
	LifetimeSeconds = 3600
	LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_04(
	Transform *=esp_trf_04
)
ESPTransform esp_trf_05(
	CipherAlg *= "G2814789CPRO1-K256-CBC-254"
	LifetimeSeconds = 3600
	LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_05(
	Transform *=esp_trf_05
)
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,ike_trf_04
	IKECFGRequestAddress = TRUE
)
IPsecAction ipsec_action_01(
	PersistentConnection = TRUE
	TunnelingParameters *=
		TunnelEntry(
			PeerAddress = 10.1.1.2
			Assemble = TRUE
			ReRoute = FALSE
			TCPEncapsulation = FALSE
		)
	ContainedProposals                                                              *=
(esp_proposal_01),(esp_proposal_02),(esp_proposal_03),(esp_proposal_04),(esp_proposal
_05)
	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 *= 10.1.1.2
		ProtocolID *= 50
		Action = PASS
		LogEventID = "pass_action_03_01"
		PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
	),Filter(
		SourceIP *= 10.1.1.2
		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 *= 10.1.1.2
            ProtocolID *= 50
            Action = PASS
            LogEventID = "pass_action_06_01"
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            DestinationIP *= 10.1.1.2
            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 *= 10.1.1.2
            ProtocolID *= 50
            Action = PASS
            LogEventID = "pass_action_11_01"
```

```
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            DestinationIP *= 10.1.1.2
            ProtocolID *= 51
            Action = PASS
            LogEventID = "pass_action_11_02"
            PacketType = LOCAL_UNICAST,LOCAL_MISDIRECTED
      ),Filter(
            DestinationIP *= 192.168.1.0/24
            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
)
```