

# Построение VPN туннеля между двумя подсетями, защищаемыми шлюзами безопасности «Bel VPN Gate» (один из шлюзов находится в топологии on-a-stick)

## Описание стенда

Сценарий иллюстрирует построение защищенного соединения между двумя подсетями SN1 и SN2, которые защищаются шлюзами безопасности «Bel VPN Gate». Для защиты будет построен VPN туннель между устройствами GW1 и GW2. Устройства IPhost1 и IPhost2 смогут общаться между собой по защищенному каналу (VPN). Все остальные соединения разрешены, но защищаться не будут. Шлюз GW1 находится в топологии on-a-stick.

В рамках данного сценария для аутентификации партнеры будут использовать сертификаты. Параметры защищенного соединения:

- IKE параметры:
  - Аутентификация – на сертификатах открытого ключа ЭЦП по СТБ 34.101.45;
  - Алгоритм шифрования – СТБ 34.101.31-2011 (6.4);
  - Алгоритм вычисления хеш-функции – СТБ 34.101.31-2011 (6.9);
  - Протокол согласования ключей – протокол Диффи-Хеллмана на эллиптических кривых (СТБ 34.101.66.2-1014).
- IPsec параметры:
  - Туннельный режим, протокол ESP:
    - Алгоритм шифрования – СТБ 34.101.31-2011 (6.4);
    - Алгоритм контроля целостности – СТБ 34.101.31-2011 (6.6).

Схема стенда (Рисунок):

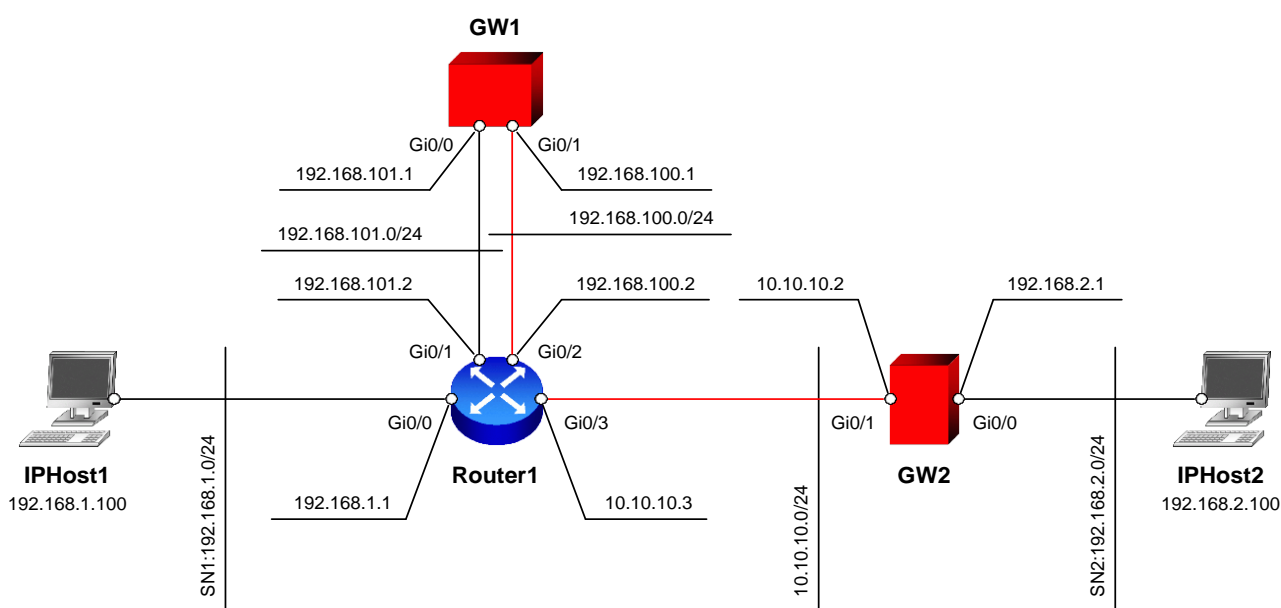


Рисунок 1

## Логика работы шлюза в топологии on-a-stick

Устройство Router1 должно быть настроено таким образом, чтобы трафик, подлежащий шифрованию, был направлен на шлюз безопасности. Сделать это можно обычным статическим маршрутом. Далее шлюз безопасности шифрует трафик и возвращает его на маршрутизатор. Так как трафик был инкапсулирован и адрес получателя в заголовках пакетов изменился – маршрутизатор посылает его уже по другому маршруту: к устройству GW2. Аналогично происходит процесс расшифровки. Технически шлюз безопасности может задействовать в топологии on-a-stick только один интерфейс.

# Настройка стенда

## Настройка устройства Router1

На устройстве Router1 должны быть прописаны следующие маршруты (синтаксис маршрутизатора Cisco):

```
ip route 192.168.2.0 255.255.255.0 192.168.101.1
ip route 0.0.0.0 0.0.0.0 10.10.10.2
```

Эти правила направляют трафик, подлежащий шифрованию, на шлюз GW1. Уже зашифрованный трафик отправляется на шлюз GW2.

## Настройка шлюза безопасности GW1

Настройку начните со шлюза безопасности GW1. Все настройки производятся через локальную консоль или удаленно (SSH с правами суперпользователя) по доверенному каналу связи.

Инициализация шлюза описывается в документации на ПАК «Bel VPN Gate 4.5» – [bel\\_vpn\\_gate\\_45\\_userguides](#) («Руководство пользователя. Настройка», раздел «Инициализация ПАК Bel VPN Gate при первом старте»).

В данном сценарии для аутентификации используются сертификаты. Для корректной работы необходимо зарегистрировать сертификат CA (УЦ) и локальный сертификат.

В данном сценарии список отозванных сертификатов (CRL) не используется и будет отключен. Информацию об использовании CRL можно найти в документации на ПАК «Bel VPN Gate 4.5» – [bel\\_vpn\\_gate\\_45\\_cisco-like](#) («Руководство пользователя. Cisco-like команды», раздел «Команды для работы с сертификатами»).

## Настройка интерфейсов

IP-адреса для интерфейсов рекомендуется настроить через cisco-like консоль.

1. Для входа в консоль запустите cs\_console:

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

Пароль по умолчанию: csp.

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:

```
sterragate(config)#interface GigabitEthernet 0/0
sterragate(config-if)#ip address 192.168.101.1 255.255.255.0
sterragate(config-if)#no shutdown
sterragate(config-if)#exit
sterragate(config)#interface GigabitEthernet 0/1
sterragate(config-if)#ip address 192.168.100.1 255.255.255.0
sterragate(config-if)#no shutdown
sterragate(config-if)#exit
```

4. Задайте статические маршруты:

```
sterragate(config)#ip route 0.0.0.0 0.0.0.0 192.168.100.2
sterragate(config)#ip route 192.168.1.0 255.255.255.0 192.168.101.2
```

5. Выйдите из cisco-like интерфейса:

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

## Формирование запроса и регистрация сертификата

Для регистрации CA сертификата (сертификата УЦ) необходимо выполнить следующие действия:

1. Установите правильное системное время.

```
root@sterragate:~# date MMDDHHmmYYYY
```

**MM** – месяц;

**DD** – день;

**HH** – часы;

**mm** – минуты;

**YYYY** – год

Пример установки даты:

```
root@sterragate:~# date 041013152013
Wed Apr 10 13:15:00 UTC 2013
```

Данная запись соответствует 10 апреля 2013 года 13:15.

6. Создайте папку /opt/certs:

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

7. Создайте контейнер на ключевом носителе:

```
root@sterragate:~/opt/Avset/bin/cryptocont n -n=контейнер -p=пароль
```

**контейнер** – название создаваемого контейнера, для создания на НКИ ДОЛЖНО содержать в начале названия префикс **“av:”**;

**пароль** – пароль (PIN) для доступа к носителю ключевой информации AvPass/AvBign.

Пример создания криптоконтейнера на НКИ:

```
root@sterragate:~/opt/Avest/bin/cryptocont n -n=av:container -p=12345678
```

8. Сформируйте запрос на сертификат.

```
root@sterragate:~/opt/Avset/bin/cryptocont r -n=контейнер -p=пароль -cn=CommonName
-c=BY -o=OrgName -t=OrgUnitName -f=путь_к_файлу
```

**контейнер** – название контейнера, созданного на предыдущем шаге;

**пароль** – пароль (PIN) для доступа к носителю ключевой информации;

**CommonName** – идентификатор устройства;

**OrgName** – наименование организации;

**OrgUnitName** – наименование подразделения;

**путь\_к\_файлу** – путь к файлу с создаваемым запросом, рекомендуется указывать расширение **“.req”**.

Пример создания запроса:

```
root@sterragate:~/opt/Avest/bin/cryptocont r -n=av:container -p=12345678 -cn=GW1 -
c=BY -o=S-Terra -t=Research -f=/opt/certs/GW1.req
```

9. Передайте полученный запрос сертификата на УЦ и получите файл сертификата (в с расширением **r7b** или **cer**).

Если вы получили файл сертификата в формате r7b, выполните экспорт в отдельные cer файлы.

10. Доставьте файлы сертификатов на Шлюз безопасности в предварительно созданный на нем каталог /opt/certs. Для доставки можно воспользоваться утилитой pscp.exe из пакета Putty, применив команду:

```
pscp -P 22 исходный_файл root@адрес_шлюза:/путь_к_файлу
```

**исходный\_файл** – путь к файлу сертификата;

**адрес\_шлюза** – сетевой адрес Шлюза;

**путь\_к\_файлу** – полный путь для сохранения файла на Шлюзе.

Пример передачи файла на Шлюз безопасности:

```
pscp -P 22 D:\ca.cer root@192.168.1.1:/opt/certs
...
Store key in cache? (y/n)
root@192.168.1.1's password:
```

**Важно:** Среда передачи в этом случае должна быть доверенной. Описание создания доверенной среды через недоверенные каналы связи смотрите в документации на ПАК «Bel VPN Gate 4.5» [bel\\_vpn\\_gate\\_45\\_userguides](#) («Руководство пользователя. Настройка», раздел «Построение VPN туннеля между шлюзом безопасности Bel VPN Gate 4.5 и рабочим местом администратора для удаленной настройки шлюза»).

11. Выполните импорт сертификата УЦ в базу Шлюза используя утилиту `cert_mgr`:

```
root@sterragate:~# cert_mgr import -f путь_к_файлу -t
```

**путь\_к\_файлу** – полный путь к файлу сертификата УЦ

Пример импорта:

```
root@sterragate:~# cert_mgr import -f /opt/cert/UC.cer -t
1 OK C=BY,L=Minsk,O=S-Terra,OU=Research,CN=UC
```

12. Выполните импорт локального (личного) сертификата в базу Шлюза:

```
root@sterragate:~# cert_mgr import -f путь_к_файлу -kc контейнер -kcp пароль
```

**путь\_к\_файлу** – полный путь к файлу сертификата УЦ;

**контейнер** – название контейнера, созданного ранее. Если контейнер храниться на ключе, введите серийный номер ключа в формате

`av:серийный_номер:название_контейнера`;

**пароль** – пароль для доступа к ключевому носителю информации.

Пример импорта:

```
root@belvpngate:~#cert_mgr import -f /opt/cert/GW1.cer -
kc av:AVP012345678910:container -kcp 12345678
1 OK CN=GW1,C=BY,O=S-Terra,OU=Research
```

13. Выведите список сертификатов, находящихся в базе Шлюза, командой `cert_mgr show` и проверьте наличие записей **trusted** и **local**:

```
root@sterragate:~# cert_mgr show
```

Пример вывода:

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

## Создание политики безопасности

После регистрации сертификатов необходимо создать политику безопасности для GW1. Создавать политику рекомендуется в интерфейсе командной строки. Для входа в консоль запустите `cs_console`:

```
root@sterragate:~# cs_console
sterragate>en
```

```
Password:
```

Пароль по умолчанию: csp.

**Важно:** пароль по умолчанию необходимо сменить.

1. Перейдите в режим настройки:

```
sterragate#conf t
```

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

2. Смените пароль по умолчанию:

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

3. Смените название шлюза:

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

4. Задайте тип идентификации:

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

5. Задайте параметры для IKE:

```
GW1(config)#crypto isakmp policy 1
GW1(config-isakmp)#hash gost
GW1(config-isakmp)#encryption belt
GW1(config-isakmp)#authentication belt-sig
GW1(config-isakmp)#group beltdh
GW1(config-isakmp)#exit
```

6. Создайте набор преобразований для IPsec:

```
GW1(config)#crypto ipsec transform-set TSET esp-belt esp-belt-mac
GW1(cfg-crypto-trans)#mode tunnel
GW1(cfg-crypto-trans)#exit
```

7. 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 0.0.0.255 192.168.2.0 0.0.0.255
GW1(config-ext-nacl)#exit
```

8. Создайте крипто-карту:

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

```
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
```

```
GW1(config-crypto-map)#match address LIST
GW1(config-crypto-map)#set transform-set TSET
GW1(config-crypto-map)#set pfs beltdh
GW1(config-crypto-map)#set peer 10.10.10.2
GW1(config-crypto-map)#exit
```

9. 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
```

10. Отключите обработку списка отзыванных сертификатов (CRL):

```
GW1(config)#crypto pki trustpoint s-terra_technological_trustpoint
```

```
GW1 (ca-trustpoint)#revocation-check none
GW1 (ca-trustpoint)#exit
```

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

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

12. Убедитесь что все сертификаты активны – статус сертификата должен быть **active**:

```
root@sterragate:~# cert_mgr check
```

Пример:

```
root@sterragate:~# cert_mgr check
1 State: Active C=BY,L=Minsk,O=S-Terra,OU=Research,CN=UC
2 State: Active CN=GW1,C=BY,O=S-Terra,OU=Research
```

В Приложении представлен текст [cisco-like конфигурации](#) для шлюза GW1.

## Настройка шлюза GW2

Настройка шлюза безопасности GW2 происходит аналогично настройке устройства GW1, с заменой IP-адресов в соответствующих разделах конфигурации.

Отдельно необходимо отметить, что в крипто-карте следует указывать «внешний» адрес шлюза GW1:

```
GW2 (config)#crypto map CMAP 1 ipsec-isakmp
```

```
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
```

```
GW2 (config-crypto-map)#match address LIST
GW2 (config-crypto-map)#set transform-set TSET
GW2 (config-crypto-map)#set pfs beltdh
GW2 (config-crypto-map)#set peer 192.168.100.1
GW2 (config-crypto-map)#exit
```

В Приложении представлен текст [cisco-like конфигурации](#) для шлюза GW2.

## Настройка устройства IPHost1

На устройстве IPHost1 задайте IP-адрес, а в качестве шлюза по умолчанию укажите IP-адрес интерфейса Gi0/0 устройства Router1 – 192.168.1.1.

## Настройка устройства IPHost2

На устройстве IPHost2 задайте IP-адрес, а в качестве шлюза по умолчанию укажите IP-адрес внутреннего интерфейса шлюза безопасности GW2 – 192.168.2.1.

## Проверка работоспособности стенда

После того, как настройка всех устройств завершена, иницируйте создание защищенного соединения.

На устройстве IPHost1 выполните команду ping:

```
ping 192.168.2.100
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.  
64 bytes from 192.168.2.100: icmp_req=1 ttl=61 time=1293 ms  
64 bytes from 192.168.2.100: icmp_req=2 ttl=61 time=284 ms  
64 bytes from 192.168.2.100: icmp_req=3 ttl=61 time=4.85 ms  
64 bytes from 192.168.2.100: icmp_req=4 ttl=61 time=6.28 ms  
64 bytes from 192.168.2.100: icmp_req=5 ttl=61 time=4.83 ms  
^C  
--- 192.168.2.100 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4012ms  
rtt min/avg/max/mdev = 4.835/318.784/1293.168/499.070 ms, pipe 2
```

В результате выполнения этой команды между устройствами GW1 и GW2 будет установлен VPN туннель.

Убедиться в этом можно, выполнив на устройстве GW1 команду:

```
root@GW1:~# 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 (192.168.100.1,500)-(10.10.10.2,500) active 1976 1904  
  
IPsec connections:  
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd  
1 3 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 440 440
```

Согласно созданной политике безопасности весь трафик между сетями SN1 и SN2 будет зашифрован. Прохождение остального трафика будет разрешено, но не будет защищаться шифрованием.



## Приложение

### Текст cisco-like конфигурации для шлюза GW1

```
!  
version 12.4  
no service password-encryption  
!  
crypto ipsec df-bit copy  
crypto isakmp identity dn  
username ccons privilege 15 password 0 csp  
aaa new-model  
!  
!  
hostname GW1  
enable password csp  
!  
!  
logging trap debugging  
!  
!  
crypto isakmp policy 1  
  encr belt  
  hash belt  
  authentication belt-sig  
  group beltdh  
!  
crypto ipsec transform-set TSET esp-belt esp-belt-mac  
!  
ip access-list extended LIST  
  permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255  
!  
!  
crypto map CMAP 1 ipsec-isakmp  
  match address LIST  
  set transform-set TSET  
  set pfs beltdh  
  set peer 10.10.10.2  
!  
interface GigabitEthernet0/0  
  ip address 192.168.101.1 255.255.255.0  
!  
interface GigabitEthernet0/1  
  ip address 192.168.100.1 255.255.255.0  
  crypto map CMAP  
!  
interface GigabitEthernet0/2  
  no ip address  
  shutdown  
!  
interface GigabitEthernet0/3  
  no ip address  
  shutdown  
!  
!  
ip route 0.0.0.0 0.0.0.0 192.168.100.2  
ip route 192.168.1.0 255.255.255.0 192.168.101.2  
!
```

```
crypto pki trustpoint s-terra_technological_trustpoint
  revocation-check none
crypto pki certificate chain s-terra_technological_trustpoint
certificate 4E4B0B11EFDB389E4E86244CDAA1B275
30820216308201C5A00302010202104E4B0B11EFDB389E4E86244CDAA1B27530
...
06010401823715010403020100300806062A85030202030341004A63F022F03D
009B097DD81A81CFC792664AAC 9E6908587195AE17A5D526DE196CB0D5B7E713
E9D07F4DC61F04CDBC87579FC44CE66D524CF742F2784805733F

quit
!
end
```

## Текст cisco-like конфигурации для шлюза GW2

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
username cscons privilege 15 password 0 csp
aaa new-model
!
!
hostname GW2
enable password csp
!
!
logging trap debugging
!
!
crypto isakmp policy 1
  encr belt
  hash belt
  authentication belt-sig
  group beltdh
!
crypto ipsec transform-set TSET esp-belt esp-belt-mac
!
ip access-list extended LIST
  permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
!
!
crypto map CMAP 1 ipsec-isakmp
  match address LIST
  set transform-set TSET
  set pfs beltdh
  set peer 192.168.100.1
!
interface GigabitEthernet0/0
  ip address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/1
  ip address 10.10.10.2 255.255.255.0
  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 10.10.10.3
!
crypto pki trustpoint s-terra_technological_trustpoint
  revocation-check none
crypto pki certificate chain s-terra_technological_trustpoint
certificate 4E4B0B11EFDB389E4E86244CDAA1B275
30820216308201C5A00302010202104E4B0B11EFDB389E4E86244CDAA1B27530
...
E9D07F4DC61F04CDBC87579FC44CE66D524CF742F2784805733F
quit
!
end
```