Организация IPsec-соединения с IKEv2 при помощи криптошлюза «С-Терра Шлюз» на западных алгоритмах

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

Для организации IPsec-соединения с IKEv2 на западных алгоритмах нужно использовать «С-Терра Шлюз», начиная с версии 5.0.26253 (5.0 BF).

Данный сценарий содержит пример настройки криптошлюза «С-Терра Шлюз» с различными устройствами (Cisco, StrongSwan, Android, Windows, iOS, macOS, AnyConnect) для обеспечения безопасного взаимодействия между защищаемой подсетью центрального офиса, двумя филиалами и мобильными клиентами.

Безопасность взаимодействия обеспечивается применением шифрования и туннелирования трафика с использованием западных алгоритмов и протокола IPsec с IKEv2.

В рамках сценария, подсети филиалов между собой безопасно взаимодействовать не могут. Настройка безопасного взаимодействия подсетей филиалов между собой с перешифрованием в центре описана в главе «Дополнение».

В рамках сценария настройка шифрования производится с помощью создания LSP вручную. Настройка шифрования с помощью cisco-like консоли представлена в Приложении.

Настройка шифрования с помощью cisco-like консоли реализована на несертифицированном криптошлюзе, обратитесь в службу технической поддержки компании для уточнения возможности использования несертифицированного криптошлюза.

В Приложении представлены дополнительные разделы, посвященные использованию УЦ на базе MSCA на примере настройки криптошлюза и Windows, поддержке аутентификации с использованием предварительно распределенных ключей, настройке трассировки в Windows и настройке логирования событий на Windows, Cisco, StrongSwan, AnyConnect.

Предварительные требования

Требования к квалификации администратора

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

Требования к инфраструктуре

1. Требования к устройствам.

   1.1. «С-Терра Шлюз».

      1.1.1. Устройство «С-Терра Шлюз» должно быть инициализировано (подробнее на https://doc.s-terra.ru раздел С-Терра Шлюз -> С-Терра Шлюз <версия> -> «Подключение ПАК и инициализация С-Терра Шлюз на вычислительных системах архитектуры Intel x86-64»).

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

   1.2. 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.

2. Требования к сетевому взаимодействию.

   2.1. Между устройствами стенда должна быть обеспечена IP связность.

Схема взаимодействия

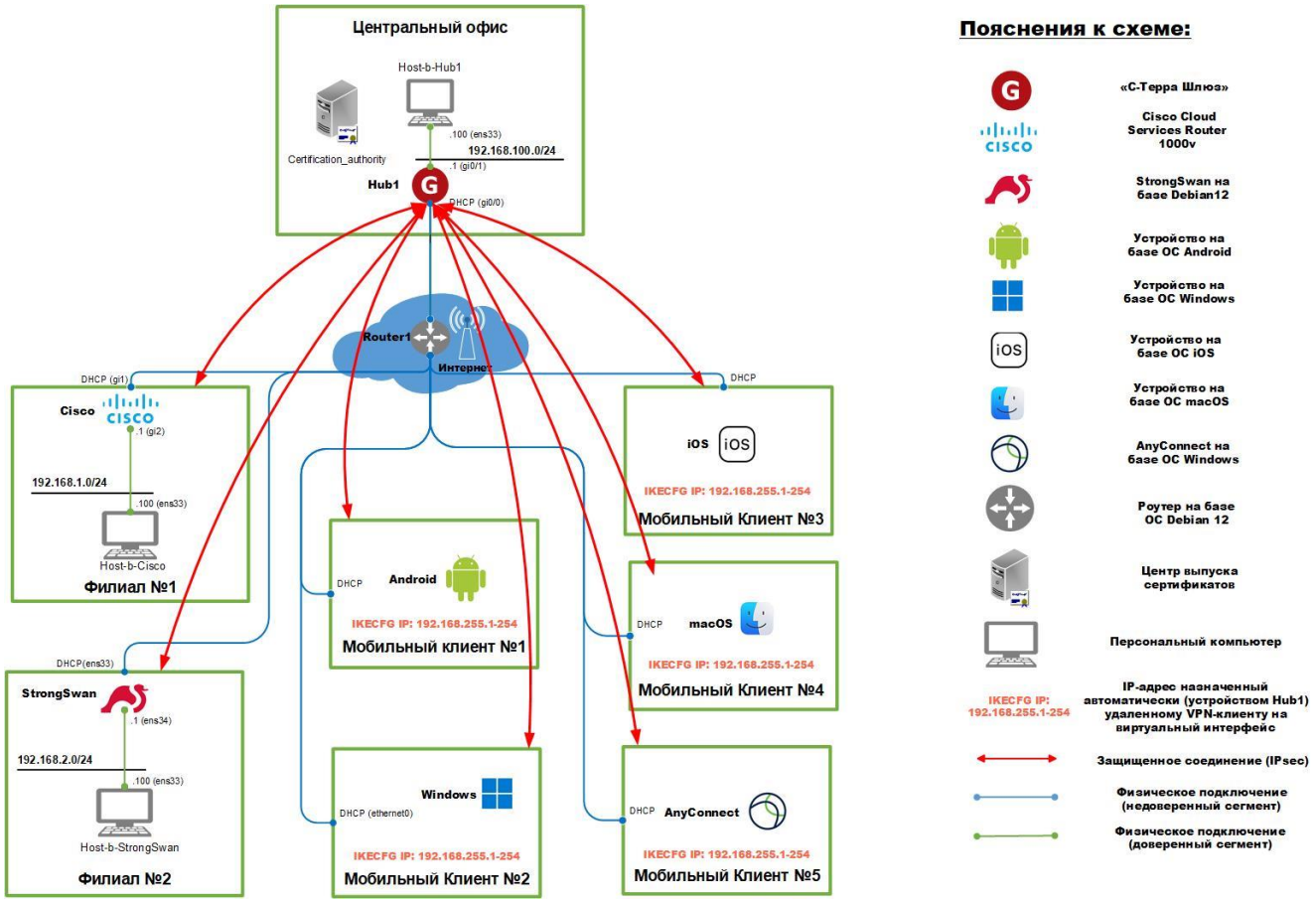


Рисунок 1. Схема взаимодействия

Общая логика работы

1. Размещение устройств.

   1.1. В центральном офисе размещаются: центр выпуска сертификатов (Certification_authority), криптошлюз «С-Терра Шлюз» (Hub1) и персональный компьютер (Host-b-Hub1).

   1.2. В филиале № 1 размещаются: устройство Cisco Cloud Services Router 1000v (Cisco) в редакции AX версии 17.3.5 и персональный компьютер (Host-b-Cisco).

   1.3. В филиале № 2 размещаются: устройство на базе Debian12 с установленным StrongSwan версии 5.9.8 и персональный компьютер (Host-b-StrongSwan).

   1.4. В качестве мобильного клиента № 1 используется устройство на базе ОС Android.

   1.5. В качестве мобильного клиента № 2 используется компьютер пользователя на базе ОС Windows 10 версии 22H2 (сборка ОС 19045.3930) или ОС Windows 11 версии 22H2 (сборка ОС 22621.2134).

   1.6. В качестве мобильного клиента № 3 используется устройство на базе ОС iOS.

   1.7. В качестве мобильного клиента № 4 используется компьютер пользователя с чипом Apple M1 с ОС macOS версии Ventura 13.4.1.

   1.8. В качестве мобильного клиента № 5 используется Cisco AnyConnect Secure Mobility Client версии 4.8. Клиент разворачивается на ОС на базе ОС Windows 10 версии 22H2 (сборка ОС 19045.3930) или ОС Windows 11 версии 22H2 (сборка ОС 22621.2134).

   Использование Cisco AnyConnect Secure Mobility Client версии 4.9 и выше не представляется возможным в связи с тем, что AnyConnect при построении защищенного соединения IPsec с IKEv2 использует только криптоалгоритмы AES-GCM, которые не реализованы в криптошлюзе.

2. Подключение к сети Интернет.

   В данном сценарии для эмуляции сети Интернет используются маршрутизатор Router1.

   2.1. Криптошлюз Hub1 подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.2. Устройство Cisco подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.3. Устройство StrongSwan на базе Debian12 подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.4. Устройство на базе ОС Android подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.5. Устройство на базе ОС Windows подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.6. Устройство на базе ОС iOS подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.7. Устройство на базе ОС macOS подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

   2.8. Устройство AnyConnect на базе ОС Windows подключается к сети Интернет с помощью DHCP-сервера (маршрут по умолчанию через маршрутизатор Router1).

3. Параметры шифрования IPsec.

   Весь IPv4 трафик между подсетями центрального офиса (192.168.100.0/24), филиалов (192.168.1.0/24, 192.168.2.0/24) и мобильными клиентами защищается с использованием западных алгоритмов и протокола IPsec с IKEv2 в туннельном IPv4 режиме.

Инициировать защищенное соединение между криптошлюзом и Cisco может трафик из подсети филиала № 1 (192.168.1.0/24) в подсеть центра (192.168.100.0/24) и наоборот из центра в филиал № 1.

Инициировать защищенное соединение между криптошлюзом и StrongSwan может трафик из подсети филиала № 2 (192.168.2.0/24) в подсеть центра (192.168.100.0/24) и наоборот из центра в филиал № 2.

Инициировать защищенное соединение между криптошлюзом и мобильными клиентами в рамках данного сценария возможно только трафиком от мобильных клиентов в подсеть центра (192.168.100.0/24).

3.1. Параметры протокола IKE:

- Аутентификация при помощи цифровых сертификатов, алгоритм подписи – RSA 2048;
- Алгоритм шифрования – ENCR_AES_CBC_256;
- Алгоритм проверки целостности пакета – AUTH_HMAC_SHA2_256_128;
- Псевдослучайная функция – PRF_HMAC_SHA2_256;
- Алгоритм выработки общего ключа (аналог алгоритма Диффи-Хеллмана) – group 14.

3.2. Параметры протокола ESP:

- Комбинированный алгоритм шифрования – ENCR_AES_CBC_256;
- Алгоритм проверки целостности пакета – AUTH_HMAC_SHA2_256_128;
- PFS (Perfect Forward Secrecy) – Алгоритм выработки общего ключа (аналог алгоритма Диффи-Хеллмана) – group 14.

Устройства Cisco и AnyConnect не поддерживают алгоритмы выработки общего ключа (PFS). В связи с этой особенностью PFS исключён из параметров протокола ESP для этих устройств.

Настройка стенда

Настройка устройства Host-b-Hub1

1. Настройте IP адрес – 192.168.100.100 и маску – 255.255.255.0 на сетевом интерфейсе.
2. Задайте маршрут по умолчанию через 192.168.100.1.
3. Разрешите прием и отправку ICMP пакетов (если заблокировано).

Настройка устройства Host-b-Cisco

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 Host-b-StrongSwan

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 (в нашем примере это подсеть 192.168.100.0/24). Доверенным каналом связи также считается канал, защищенный при помощи протокола IPsec.

Начальные настройки

Дата и время на криптошлюзе и других устройствах должны быть одинаковы, так как для аутентификации используются цифровые сертификаты, в которых зафиксированы дата и время начала их действия и окончания. Также одинаковые дата и время на всей инфраструктуре облегчают поиск неисправностей по лог-файлам.

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.

Настройка паролей доступа к консоли cisco-like

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

Подробная настройка с описанием команд приведена в базовом сценарии.

Настройка сетевых параметров

Изменение состояния интерфейсов и назначение IP адресов применяются сразу после ввода соответствующих команд. **Политика безопасности применяется после выхода из режима конфигурирования.**

1. Задайте имя устройства:

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

2. Включите внешний `GigabitEthernet0/0` и внутренний `GigabitEthernet0/1` интерфейсы. В зависимости от параметров сетевого интерфейса его имя может отличаться от `GigabitEthernetN/M` и иметь имена, например `FastEthernetN/M`, `TenGigabitEthernetN/M`:

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

3. Убедитесь, что интерфейсы административно включены и line protocol (способность интерфейса передавать пакеты в данный момент) находится в состоянии up:

```
Hub1(config)#do show interfaces
```

```
GigabitEthernet0/0 is up, line protocol is up
  Hardware address is 000c.2968.b16d
  MTU 1500 bytes
GigabitEthernet0/1 is up, line protocol is up
  Hardware address is 000c.2968.b177
  MTU 1500 bytes
...
```

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

4. Задайте IP адрес в соответствии со схемой стенда на внутреннем `GigabitEthernet0/1` интерфейсе:

```
Hub1(config)#interface GigabitEthernet0/1
Hub1(config-if)#ip address 192.168.100.1 255.255.255.0
Hub1(config-if)#end
Hub1#exit
```

5. Войдите в linux bash консоль из CLI разграничения доступа:
6. Настройте получение IP адреса на внешний интерфейс `GigabitEthernet0/0`. Для этого измените файл настройки сети `/etc/network/interfaces` так, чтобы описание интерфейса было вне секции, начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###`, а описание интерфейса в секции удалите, например:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

# loopback configuration
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

###netifcfg-begin###
auto eth1
iface eth1 inet static
mtu 1500
address 192.168.100.1
netmask 255.255.255.0
broadcast 192.168.100.255
###netifcfg-end###
```

7. Сохраните изменения в файле и перезапустите сетевые настройки на криптошлюзе:

```
root@Hub1:~#systemctl restart networking.service
```

8. По умолчанию, весь сетевой трафик заблокирован. Для разблокировки прохождения только пакетов по протоколу DHCP выполните команду:

```
root@Hub1:~# dp_mgr set -ddp passdhcp
```

```
Default driver policy is set successfully
```

Настройки PKI (запросы и сертификаты)

В Приложении описан альтернативный способ настройки PKI с использованием УЦ на базе MSCA на примере криптошлюза и мобильного клиента на базе ОС Windows. Однако его использование не рекомендуется, так как не все клиенты имеют возможность генерировать закрытый ключ и запрос на сертификат. В архитектуре УЦ MSCA клиент сам генерирует закрытый ключ и запрос на сертификат, который затем подписывается УЦ. Для удобства и упрощения сценария используется УЦ на базе OpenSSL, так как он предоставляет возможность создавать запросы на сертификаты для всех типов клиентов.

1. Генерация закрытого ключа и запроса на сертификат для криптошлюза.

1.1. Перед генерацией закрытого ключа и запроса на сертификат криптошлюза посмотрите, какой IP адрес был получен по DHCP на интерфейсе `eth0`:

```
root@Hub1:~#ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever

    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 10000
    link/ether 00:0c:29:68:b1:6d brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname ens33
    altname GigabitEthernet0_0
    inet 10.0.102.25/16 brd 10.0.255.255 scope global eth0:1
       valid_lft forever preferred_lft forever
    inet6 fe80::3131:e59c:1b76:9759/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 10000
    link/ether 00:0c:29:68:b1:77 brd ff:ff:ff:ff:ff:ff
    altname enp2s2
    altname ens34
    altname GigabitEthernet0_1
    inet 192.168.100.1/24 brd 192.168.100.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::8a20:427b:dc62:8d40/64 scope link tentative stable-privacy
       valid_lft forever preferred_lft forever
```

1.2. Запустите процесс генерации закрытого ключа и запроса на сертификат криптошлюза с сохранением файла запроса:

В данной команде необходимо индивидуально вписывать IP адрес внешнего интерфейса криптошлюза, на котором будет закреплена крипто-карта.

```
root@Hub1:~#cert_mgr create -subj "C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1" -
ip 10.0.102.25 -RSA -2048 -fb64 /root/hub1.request
```

2. Выпускать сертификаты будет предустановленный на криптошлюз УЦ на базе OpenSSL.

2.1. Выберите каталог `/root/ca` для хранения всех ключей и сертификатов:

```
root@Hub1:~#mkdir /root/ca
```

2.2. Создайте структуру каталогов. Файлы `index.txt` и `serial` действуют как база данных плоских файлов для отслеживания подписанных сертификатов:

```
root@Hub1:~#cd /root/ca
root@Hub1:~/ca#mkdir certs crl newcerts private csr
root@Hub1:~/ca#chmod 700 private
root@Hub1:~/ca#touch index.txt
root@Hub1:~/ca#echo 1000 > serial
```

3. Для выпуска сертификатов понадобится файл конфигурации OpenSSL, в нем можно менять различные расширения и задавать их. Создайте его под именем `openssl.cnf` в папке `/root/ca/`:

При работе с мобильными клиентами есть определенные требования к сертификатам для безопасного соединения IKEv2/IPsec, подробно о них описано в главе «Создание и экспорт сертификата для мобильного клиента с криптошлюза». Ниже представлен конфигурационный файл для создания сертификатов по шаблону:

3.1. Раздел `[ ca ]` является обязательным. Укажите OpenSSL использовать параметры из раздела `[ CA_default ]`.

```
# OpenSSL root CA configuration file.
# Copy to `/root/ca/openssl.cnf`.

[ ca ]
# `man ca`
default_ca = CA_default
```

3.2. В `[ CA_default ]` разделе содержится диапазон значений по умолчанию. Убедитесь, что в качестве директории объявлен выбранный ранее каталог `/root/ca`.

```
[ CA_default ]
# Directory and file locations.
dir               = /root/ca
certs             = $dir/certs
crl_dir           = $dir/crl
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand

# The root key and root certificate.
private_key       = $dir/private/ca.key.pem
certificate       = $dir/certs/ca.cert.cer

# For certificate revocation lists.
crlnumber         = $dir/crlnumber
crl               = $dir/crl/ca.crl.pem
crl_extensions    = crl_ext
default_crl_days  = 30

# SHA-1 is deprecated, so use SHA-2 instead.
default_md        = sha256

name_opt          = ca_default
cert_opt          = ca_default
default_days      = 375
preserve          = no
policy            = policy_loose

# This extension is copied from the subjectAltName query
copy_extensions   = copy
```

3.3. В этом разделе указана свободная политика для подписи сертификатов без ограничений по какому-либо полю, так как подписываются серверные и клиентские сертификаты.

```
[ policy_loose ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

3.4. Параметры из раздела `[ req ]` применяются при создании сертификатов или запросов на подписание сертификатов.

```
[ req ]
# Options for the `req` tool (`man req`).
default_bits        = 2048
distinguished_name  = req_distinguished_name
string_mask         = utf8only

# SHA-1 is deprecated, so use SHA-2 instead.
default_md          = sha256

# Extension to add when the -x509 option is used.
x509_extensions     = v3_ca
```

3.5. В разделе `[ req_distinguished_name ]` указывается информация, которая обычно требуется в запросе на подписание сертификата. При желании можно указать некоторые значения по умолчанию.

```
[ req_distinguished_name ]
# See <https://en.wikipedia.org/wiki/Certificate_signing_request>.
commonName                      = Common Name
countryName                     = Country Name (2 letter code)
stateOrProvinceName             = State or Province Name
localityName                    = Locality Name
0.organizationName              = Organization Name
organizationalUnitName          = Organizational Unit Name
emailAddress                    = Email Address

# Optionally, specify some defaults.
countryName_default             =
stateOrProvinceName_default     =
localityName_default            =
0.organizationName_default      =
organizationalUnitName_default  =
emailAddress_default            =
```

3.6. Используются расширения при создании корневого сертификата.

```
[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints       = critical, CA:true
keyUsage               = critical, digitalSignature, cRLSign, keyCertSign
```

3.7. В разделе `[ usr_cert]` указаны расширения, используемые при подписании клиентских сертификатов, которые используются для удаленной аутентификации пользователя.

```
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid,issuer:always
keyUsage                = digitalSignature
extendedKeyUsage        = clientAuth
```

3.8. В разделе `[ server_cert ]` указаны расширения, используемые при подписании сертификатов криптошлюза или граничных шлюзов филиалов.

```
[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid,issuer:always
keyUsage                = digitalSignature, keyEncipherment
extendedKeyUsage        = serverAuth
```

3.9. В разделе `[ crl_ext ]` расширение автоматически применяется при создании списков отзыва сертификатов.

```
[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier = keyid:always
```

В Приложении представлен текст конфигураций OpenSSL для криптошлюза.

4. Создайте корневой ключ `ca.key.pem` и храните его в абсолютной безопасности. Задайте также права только на чтение.

```
root@Hub1:~/ca#openssl genrsa -aes256 -out private/ca.key.pem 2048
```

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

```
root@Hub1:~/ca#chmod 400 private/ca.key.pem
```

Любой, у кого есть корневой ключ, может выдавать доверенные сертификаты. Зашифруйте корневой ключ с помощью 256-битного шифрования AES и надежного пароля.

5. Создайте корневой сертификат. Для этого используйте корневой ключ `ca.key.pem` для создания корневого сертификата `ca.cert.cer`. Укажите корневому сертификату длительный срок действия, например, двадцать лет. По истечении срока действия корневого сертификата все сертификаты, подписанные центром сертификации, становятся недействительными.

```
root@Hub1:~/ca# openssl req -config openssl.cnf -key private/ca.key.pem -new -x509 -
days 7300 -sha256 -extensions v3_ca -out certs/ca.cert.cer
```

```
Enter pass phrase for private/ca.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name []:S-Terra CSP Test Root CA-OpenSSL
Country Name (2 letter code) []:RU
State or Province Name []:
Locality Name []:Zelenograd
Organization Name []:S-Terra CSP
Organizational Unit Name []:RnD
Email Address []:
```

```
root@Hub1:~/ca#chmod 444 certs/ca.cert.cer
```

6. Подпишите сертификат для криптошлюза:

   6.1. Выпустите сертификат на УЦ OpenSSL по сгенерированному запросу на сертификат:

В команде необходимо указать раздел `server_cert`, чтобы сертификат был выпущен с необходимыми расширениями. В сертификате должно быть «Дополнительное имя субъекта» с IP адресом криптошлюза, который был указан при генерации запроса, а также должно быть расширение «Улучшенный ключ» со значением «Проверка подлинности сервера». Все необходимые расширения для сертификата криптошлюза указаны в файле конфигурации `openssl.cnf` в разделе `server_cert`.

```
root@Hub1:~/ca#openssl ca -config openssl.cnf -extensions server_cert -days 375 -notext
-md sha256 -in /root/hub1.request -out certs/hub1.cer
```

```
 Using configuration from openssl.cnf
Enter pass phrase for /root/ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4100 (0x1004)
        Validity
            Not Before: Jul 29 01:37:40 2024 GMT
            Not After : Aug  8 01:37:40 2025 GMT
        Subject:
            countryName               = RU
            localityName              = Zelenograd
            organizationName          = S-Terra CSP
            organizationalUnitName    = RnD
            commonName                = Hub1
```

```
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                EC:8D:93:40:7C:30:48:A7:95:8D:1E:F6:B9:1B:9B:A6:A6:97:9D:F4
            X509v3 Authority Key Identifier:
                keyid:3C:6E:22:57:B8:50:A5:95:A5:0E:CC:95:78:02:04:49:7C:5C:28:CD
                DirName:/CN=S-Terra CSP Test Root CA-OpenSSL/C=RU/L=Zelenograd/O=S-
Terra CSP/OU=RnD
                serial:77:0C:FD:38:EF:6F:E6:9B:B1:69:CE:48:FE:D6:F4:50:FF:01:36:A6
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                IP Address:10.0.102.25
Certificate is to be certified until Aug  8 01:37:40 2025 GMT (375 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
```

6.2. Импортируйте сертификат криптошлюза в базу Продукта:

```
root@Hub1:~/ca#cert_mgr import -l -f certs/hub1.cer
```

6.3. Импортируйте сертификат УЦ в базу Продукта:

```
root@Hub1:~/ca#cert_mgr import -t -f certs/ca.cert.cer
```

- ключ `-f` задает месторасположение файла сертификата;
- ключ `-t` используется для импортирования доверенного (trusted) сертификата УЦ.

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

```
root@Hub1:~/ca#cert_mgr show
```

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

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

6.5. Выполните проверку статуса сертификатов в базе Продукта:

```
root@Hub1:~/ca#cert_mgr check
```

```
1 State: Inactive C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1
                  Certificate can not be verified.
2 State: Inactive CN=S-Terra CSP Test Root CA-OpenSSL,C=RU,L=Zelenograd,O=S-Terra
CSP,OU=RnD
                  Certificate can not be verified.
```

Примечание – Видно, что все сертификаты имеют статус **Inactive** (неактивный) с пометкой: «Certificate can not be verified» (сертификат не может быть проверен). Причиной этому является включенный по умолчанию в консоли cisco-like механизм проверки списка отозванных сертификатов (далее COC или CRL). Так как в базе Продукта COC отсутствует, поэтому проверка не может быть осуществлена.

Настройка шифрования с помощью LSP

1. Настройка будет осуществляться при помощи конфигурационного файла LSP, для этого создайте файл `lsp.txt`, содержимое которого потом будет загружено в LSP.

1.1. Для начала в файле LSP создайте глобальные параметры:

```
GlobalParameters(
      Title                     = "Hub1 5.0 LSP IKEv2"
      Version                   = LSP_5_0
      CRLHandlingMode           = DISABLE
      PreserveIPsecSA           = FALSE
)
```

1.2. Укажите структуру, которая будет задавать аутентификационную информацию. В данном случае аутентификация будет происходить для локальной аутентификации и для аутентификации партнера по сертификатам:

```
IKEAuth2 IKEV2-AUTH-RSA-SIG-LOCAL (
      Identity                  = IdentityEntry(
            DistinguishedName*  = USER_SPECIFIC_DATA)
      Credential                = CertDescription()
      SendCertMode              = ALWAYS
      SendRequestMode           = ALWAYS
)

IKEAuth2 IKEV2-AUTH-RSA-SIG-REMOTE (
      Credential                = CertDescription()
)
```

1.3. Создайте отдельную структуру `IdentityEntry` под именем `AnyConnect`, которая будет описывать индентификационную информацию для мобильного клиента AnyConnect. Для этого выберите атрибут `DistinguishedName` с указанием `Subject` сертификата AnyConnect:

```
IdentityEntry ANYCONNECT (
      DistinguishedName         = CertDescription(
      Subject                   = TEMPLATE,"L=Zelenograd,O=S-Terra CSP,CN=AnyConnect"
      )
)
```

1.4. Укажите структуру, которая будет иметь требования к идентификационной информации партнёра. Для данной структуры укажите в требования идентификационную информацию структуры `IdentityEntry` под именем `AnyConnect`:

```
IKEAuth2 IKEV2-AUTH-RSA-SIG-ANYCONNECT (
      Identity                  = ANYCONNECT
      Credential                = CertDescription()
)
```

1.5. Задайте пул для адресов и адрес сервера DNS, передаваемые клиенту по протоколу IKECFG:

```
AddressPool IKECFG_POOL_CLIENTS(
      IPAddresses               = 192.168.255.1..192.168.255.254
      DNSServers                = 8.8.8.8
)
```

1.6. Создайте политику, описывающую параметры: аутентификации, шифрования, проверки целостности пакета, и выработки общего ключа, для IKE туннеля:

```
IKEProposal2 IKE2-PROPOSAL (
      EncryptionAlg             = ENCR_AES_CBC_256
      IntegrityAlg              = AUTH_HMAC_SHA2_256_128
      PrfAlg                    = PRF_HMAC_SHA2_256
      KeyAlg                    = MODP_2048
)
```

1.7. Создайте структуру `IKERule2`, которая описывает правило создания контекста соединения с Cisco для протокола IKEv2. Параметр `IKEPeerIPFilter` должен содержать IP адрес внешнего интерфейса Cisco:

```
IKERule2 IKERule2:VPN:1
(
```

```
      IKEPeerIPFilter           = 10.0.108.195
      Proposals                 = IKE2-PROPOSAL
      LocalAuth                 = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                = IKEV2-AUTH-RSA-SIG-REMOTE
      DPDMode                   = PERIODIC
      DPDIdleDuration           = 5
      Priority                  = 10
)
```

1.8. Создайте структуру `IKERule2`, которая описывает правило создания контекста соединения с StrongSwan для протокола IKEv2. Параметр `IKEPeerIPFilter` должен содержать IP адрес внешнего интерфейса StrongSwan:

```
IKERule2 IKERule2:VPN:2
(
      IKEPeerIPFilter           = 10.0.108.237
      Proposals                 = IKE2-PROPOSAL
      LocalAuth                 = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                = IKEV2-AUTH-RSA-SIG-REMOTE
      DPDMode                   = PERIODIC
      DPDIdleDuration           = 5
      Priority                  = 20
)
```

1.9. Создайте структуру `IKERule2`, которая описывает правило создания контекста соединения с Мобильным клиентом AnyConnect для протокола IKEv2. Параметр `RemoteAuth` должен содержать `RemoteSign_AnyConnect`:

```
IKERule2 IKERule2:VPN:3:DMAP:1
(
      Proposals                 = IKE2-PROPOSAL
      LocalAuth                 = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                = IKEV2-AUTH-RSA-SIG-ANYCONNECT
      IKECFGPool                = IKECFG_POOL_CLIENTS
      DPDMode                   = PERIODIC
      DPDIdleDuration           = 14
      Priority                  = 100
)
```

1.10. Создайте структуру `IKERule2`, которая описывает правило создания контекста соединения с Мобильными клиентами для протокола IKEv2:

```
IKERule2 IKERule2:VPN:3:DMAP:2
(
      Proposals                 = IKE2-PROPOSAL
      LocalAuth                 = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                = IKEV2-AUTH-RSA-SIG-REMOTE
      IKECFGPool                = IKECFG_POOL_CLIENTS
      DPDMode                   = PERIODIC
      DPDIdleDuration           = 14
      Priority                  = 110
)
```

1.11. Задайте комбинированный алгоритм шифрования для трафика:

```
IPsecProposal2 IKE2-IPSEC-PROPOSAL (
      ESPEncryptionAlg          = ENCR_AES_CBC_256
      ESPIntegrityAlg           = AUTH_HMAC_SHA2_256_128
      PFSAlg                    = MODP_2048
)
```

1.12. Задайте комбинированный алгоритм шифрования для трафика без использования PFS, так как нет поддержки работы с PFS у Cisco и AnyConnect:

```
IPsecProposal2 IKE2-IPSEC-PROPOSAL-CISCO (
      ESPEncryptionAlg          = ENCR_AES_CBC_256
```

```
      ESPIntegrityAlg           = AUTH_HMAC_SHA2_256_128
)
```

1.13. В структуре `IPsecAction2` задайте правило создания контекста соединения с Cisco для протоколов семейства IPsec:

```
IPsecAction2 IPsecAction2:VPN:1
(
      TunnelingParameters       = TunnelEntry(
            PeerAddress         = 10.0.108.195
            Assemble            = TRUE
      )
      ContainedProposals        = IKE2-IPSEC-PROPOSAL-CISCO
      IKERule                   = IKERule2:VPN:1
)
```

1.14. В структуре `IPsecAction2` задайте правило создания контекста соединения с StrongSwan для протоколов семейства IPsec:

```
IPsecAction2 IPsecAction2:VPN:2(
      TunnelingParameters       = TunnelEntry(
            PeerAddress         = 10.0.108.237
            Assemble            = TRUE
      )
      ContainedProposals        = IKE2-IPSEC-PROPOSAL
      IKERule                   = IKERule2:VPN:2
)
```

1.15. В структуре `IPsecAction2` задайте правило создания контекста соединения с Мобильным клиентом AnyConnect для протоколов семейства IPsec:

```
IPsecAction2 IPsecAction2:VPN:3:DMAP:1(
      TunnelingParameters       = TunnelEntry(
             Assemble           = TRUE
      )
      ContainedProposals        = IKE2-IPSEC-PROPOSAL-CISCO
      IKERule                   = IKERule2:VPN:3:DMAP:1
)
```

1.16. В структуре `IPsecAction2` задайте правило создания контекста соединения с Мобильными клиентами для протоколов семейства IPsec:

```
IPsecAction2 IPsecAction2:VPN:3:DMAP:2(
      TunnelingParameters       = TunnelEntry(
            Assemble            = TRUE
      )
      ContainedProposals        = IKE2-IPSEC-PROPOSAL
      IKERule                   = IKERule2:VPN:3:DMAP:2
)
```

Структура `IPsecAction2` без указания параметра `PeerAddress`, является динамической крипто-картой. При использовании динамической крипто-карты криптошлюз может выступать не только в роли ответчика, но и инициировать IPsec-соединения, что может привести к превышению допустимого числа одновременных IKEv2-сессий. В рамках данного сценария корректно применять динамические крипто-карты только для подключения мобильных клиентов, где криптошлюз выступает в роли ответчика.

1.17. В структуре `FilterChain` задайте список правил пакетной и контекстной фильтрации:

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

```
            PacketType          = LOCAL_UNICAST, LOCAL_MISDIRECTED
      ),
      Filter (
            SourceIP            = 192.168.100.0/24
            DestinationIP       = 192.168.1.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:1 >
            LogEventID          = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_CISCO"
      ),
      Filter (
            SourceIP            = 192.168.100.0/24
            DestinationIP       = 192.168.2.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:2 >
            LogEventID          =
"IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_AND_STRONGSWAN"
      ),
      Filter (
            DestinationIP       = 192.168.255.0/24
            Action              = PASS
            ExtendedAction      =    ipsec<    sa    =    IPsecAction2:VPN:3:DMAP:1,
IPsecAction2:VPN:3:DMAP:2 >
            LogEventID          =
"IPsec:Protect:VPN:3:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS,                VPN:3:DMAP:2:
IPSEC_ACL_HUB1_AND_CLIENTS"
      )
)
```

В последнем фильтре параметр `SourceIP` не указывается, чтобы клиенты имели доступ не только к одной заданной защищаемой сети, а ко всем доступным для криптошлюза.

Если в подсеть, указанную в параметре `DestinationIP` последнего фильтра динамической крипто-карты, поступает трафик, криптошлюз станет инициатором IPsec. Не допускайте этого, так как это может привести к превышению допустимого количества одновременных IKEv2-сессий.

1.18. В структуре `NetworkInterface` выберите внешний интерфейс `GigabitEthernet0/0` для подключения созданной крипто-карты VPN к внешнему WAN-интерфейсу:

```
NetworkInterface (
            LogicalName         = "GigabitEthernet0/0"
            IPsecPolicy         = IPsecPolicy:VPN
)

NetworkInterface (
            LogicalName         = "GigabitEthernet0/1"
)
```

В Приложении представлен текст конфигураций LSP для криптошлюза.

2. Загрузите конфигурационный файл LSP в базу продукта:

```
root@Hub1:~/ca# lsp_mgr load -f lsp.txt
```

```
LSP successfully loaded from file "lsp.txt"
```

Настройка Cisco

Настройка сетевых параметров

1. Войдите в привилегированный режим и введите пароль, если он настроен:

```
csr1000v>enable
Password:
```

2. Войдите в режим глобальной конфигурации:

```
csr1000v#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

3. Задайте имя устройства:

```
csr1000v(config)#hostname Cisco
```

4. Включите внешний `GigabitEthernet1` и внутренний `GigabitEthernet2` интерфейсы:

```
Cisco(config)#interface range GigabitEthernet 1 - 2
Cisco(config-if-range)#no shutdown
Cisco(config-if-range)#exit
*May 16 10:32:36.456: %LINK-3-UPDOWN: Interface GigabitEthernet1, changed state to up
*May 16 10:32:36.465: %LINK-3-UPDOWN: Interface GigabitEthernet2, changed state to up
*May 16 10:32:37.456: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1,
changed state to up
*May 16 10:32:37.464: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2,
changed state to up
```

5. Задайте IP адреса в соответствии со схемой стенда на внешнем `GigabitEthernet1` и внутреннем `GigabitEthernet2` интерфейсах:

```
Cisco(config)#interface GigabitEthernet 1
Cisco(config-if)#ip address dhcp
*May 16 10:33:26.367: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet1 assigned DHCP
address 10.0.108.195, mask 255.255.0.0, hostname csr1000v
Cisco(config-if)#exit
Cisco(config)#interface GigabitEthernet 2
Cisco(config-if)#ip address 192.168.1.1 255.255.255.0
Cisco(config-if)#exit
```

6. Убедитесь, что интерфейсы настроены и находятся в состоянии up:

```
Cisco(config)#do sh ip int brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.0.108.195    YES DHCP   up                    up
GigabitEthernet2       192.168.1.1     YES manual up                    up
GigabitEthernet3       unassigned      YES NVRAM  administratively down down
GigabitEthernet4       unassigned      YES NVRAM  administratively down down
```

Настройки PKI (запросы и сертификаты)

1. Сгенерируйте пару ключей RSA размера 2048 бит под именем VPN с помощью команды ниже:

```
Cisco(config)#crypto key generate rsa label VPN modulus 2048
The name for the keys will be: VPN

% The key modulus size is 2048 bits

% Generating crypto RSA keys in background ...
```

2. Создайте точку доверия с именем `CA` для использования в процессе получения сертификатов от центра сертификации:

```
Cisco(config)#crypto pki trustpoint CA
```

3. Укажите, что запрос на получение сертификата будет введен непосредственно с устройства через терминал:

```
Cisco(ca-trustpoint)#enrollment terminal
```

4. Установите информацию о субъекте сертификата, такую как общее имя (cn), подразделение (ou), организация (o), местоположение (l), страна (c). А также выключите CRL и установите размер RSA для сертификата равным 2048:

```
Cisco(ca-trustpoint)#subject-name cn=Cisco-CSR100v, o=S-Terra CSP, l=Zelenograd, c=RU
Cisco(ca-trustpoint)#revocation-check none
Cisco(ca-trustpoint)#rsakeypair VPN
Cisco(ca-trustpoint)#exit
```

5. Запустите процесс аутентификации `CA` и вставьте сертификат УЦ в кодировке BASE64. В ходе этого процесса устройство может проверить подлинность сертификата центра сертификации, чтобы убедиться в его доверительности перед использованием сертификатов, полученных от этого CA:

```
Cisco(config)#crypto pki authenticate CA
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIDtzCCAp+gAwIBAgIUAX7UbyRms1TINoKSIMbF0wqJME4wDQYJKoZIhvcNAQEL
BQAwYzEpMCcGA1UEAwwgUy1UZXJyYSBDU1AgVGVzdCBSb290IENBLU9wZW5TU0wx
CzAJBgNVBAYTAlJVMRMwEQYDVQQHDApaZWxlbm9ncmFkMRQwEgYDVQQKDAtTLVRl
cnJhIENTUDAeFw0yNDAyMDIwOTQzMTVaFw00NDAxMjgwOTQzMTVaMGMxKTAnBgNV
BAMMIFMtVGVycmEgQ1NQIFRlc3QgUm9vdCBDQS1PcGVuU1NMMQswCQYDVQQGEwJS
VTETMBEGA1UEBwwKWmVsZW5vZ3JhZDEUMBIGA1UECgwLUy1UZXJyYSBDU1AwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDck4jT5lbNE3kPUQl8biukjRi2
p3KsFLUCsYgAJh7+ujifH+3ld0x9qBXTGjuU9teDTpQIRqMMuvfHfGp0WVhJvVzt
1qDXgC+BTKSbmZ9VhThHWyWoUvFP1nr1MvTNDNdOfK5GmJl0XXYMO9mjdPh7+GRF
wbgc0mkoujXKzaO8srFc9YJxv60up7L9nM+Vx7/NVKIyLM3W/JC02qARL0S9CRrk
o3UPXMyn939/8eZOEAmqLMILVVeRyc6y5YVb/+P6HQQSJESRDU1kZLP/cznFh9Xl
jZ+YzEzo9Er9XXvuGSrq7a63ULtybI65gSnxYb88yAil9N9vbs+JpPK1BrYRAgMB
AAGjYzBhMB0GA1UdDgQWBBSPHKdi7Z4T0MlrmraElH+mblnEgzAfBgNVHSMEGDAW
gBSPHKdi7Z4T0MlrmraElH+mblnEgzAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB
/wQEAwIBhjANBgkqhkiG9w0BAQsFAAOCAQEAO1e7zFcABfo0JxZ2sA1LJAye1S1i
dpVGWFc5Iwl8mWcbyqTPIj0/NYWO1iFB1M/i5Y4qd+exFxMjC/gfDDRk4GrMvkPA
t+VGyQ+4pM7t8nQt0gDEN/Kwq9h8stW273QwwFpT+3jrrCKzQyPkEFf0MuV+lwPf
KtbbBRH4ssuoij5kpqQwY2f0nP5sfwLg9nE2VlP9gROmRv8gjo5mxo6tIF/Nd2DP
GxKmem+ivR4MfS8eZGvdoACpRDHFouW3RxZf538mQE9mzFdMiG4ErLNuyYH0pDgs
gCtMmfx7lcjW+2Ft1v6ePdVq3xPqQbTjyqyPjW5O7FIBhyEV0tz+yTN5Bg==
-----END CERTIFICATE-----

Certificate has the following attributes:
       Fingerprint MD5: B48D3CB3 11144F33 179AE0F0 ECB06A54
      Fingerprint SHA1: FDC3CA24 E21BCF89 FF403FE1 AF892EEE E42FB02A

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

6. Инициируйте процесс генерации запроса на сертификат командой `crypto pki enroll CA`. Чтобы отобразить сгенерированный запрос на сертификат ответьте положительно `yes` на вопрос `Display Certificate Request to terminal? [yes/no]`.

   Примечание – Сертификат можно использовать без дополнительных элементов: серийного номера или IP адреса в subject name для этого ответьте отрицательно `no`. При необходимости включить эти дополнительные элементы в subject name ответьте положительно `yes`:

```
Cisco(config)#crypto pki enroll CA
% Start certificate enrollment ..

% The subject name in the certificate will include: cn=Cisco-CSR100v, o=S-Terra CSP,
l=Zelenograd, c=RU
% The subject name in the certificate will include: Cisco.ikev2.cisco.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Display Certificate Request to terminal? [yes/no]: yes
Certificate Request follows:

MIIC4TCCAckCAQAwezELMAkGA1UEBhMCUlUxEzARBgNVBAcTClplbGVub2dyYWQx
FDASBgNVBAoTC1MtVGVycmEgQ1NQMRYwFAYDVQQDEw1DaXNjby1DU1IxMDB2MSkw
JwYJKoZIhvcNAQkCFhpDaXNjby1DZXJ0LmlrZXYyLmNpc2NvLmNvbTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAJj1jpcbLmAoSDsb3bXmvugEECZeHdDH
HBrvIrTzzhYmrnPyOGu0zowyls2RXsmDOdgXD+AQyzLnIBt9faoN2sZhi4V2/5Pq
SFZ6gDZyoBMoY3H7n/gBTICfNuVGNBdPqWSRgsXoyzGEtHHioIm+jKPfdZI6TDHJ
frm8N2NCJh5itK5CW5G1xrOhS+HqUfUI/DLYVE2sM5FDsxfeqXoGS617UgJ26q+H
bgxTj3gSFtXeAvB8cIVUKn73wrCIwrmCkTv4jXDtApm9TJSYdy3YSZUosgsDRwYR
9VwJC+JG7q1y2S3fwZjXVgOKI4V87ySf+dgBWwVVoHTgZAM5lXF94vsCAwEAAaAh
MB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUA
A4IBAQAu70Kqtey8tsi2/v0X6UGGrO3FzhXwUO7gj+GY9Ta2KpuZMgM02YdqaGX4
wc53AS7btpiAnB7JwbfWlunGiL5IMmIdAoL1vTy5NjyL4ps+0JhAIYzgv9pxu6N1
c8X6YpF8fZgh+Y/q8bl/MLUghoFcytBKvx1CBeDaXJrxi4PYijr7ya1cIs0U3ydg
Y+8XkltfaC/UuYaxt6kxZtKvgW3ItfZU6GBM2LLFLQ9WBkqifHbTnPqv2QRtKxoC
o9jvUEZqH7BBwdDN2l3RnrwS5Yms1JnWUeG3ARll5ALXyt8zPtD5XFWh5390RY6M
FvW6jcrKrU/xHP7JvaNhtCu7zhQv

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: no
```

Cisco не добавляет к выводу запроса сертификата строки `-----BEGIN CERTIFICATE REQUEST-----` и `-----END CERTIFICATE REQUEST-----`.
Для корректной обработки запроса на УЦ необходимо вручную добавить эти строки перед передачей файла на подпись.

После выполнения этой команды скопируйте запрос на сертификат в кодировке BASE64 на УЦ.

7. Создайте файл с запросом Cisco на криптошлюзе под именем `cisco.pem` по пути `/root/ca/csr/cisco.pem` и обязательно добавьте заголовки:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC4TCCAckCAQAwezELMAkGA1UEBhMCUlUxEzARBgNVBAcTClplbGVub2dyYWQx
FDASBgNVBAoTC1MtVGVycmEgQ1NQMRYwFAYDVQQDEw1DaXNjby1DU1IxMDB2MSkw
JwYJKoZIhvcNAQkCFhpDaXNjby1DZXJ0LmlrZXYyLmNpc2NvLmNvbTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAJj1jpcbLmAoSDsb3bXmvugEECZeHdDH
HBrvIrTzzhYmrnPyOGu0zowyls2RXsmDOdgXD+AQyzLnIBt9faoN2sZhi4V2/5Pq
SFZ6gDZyoBMoY3H7n/gBTICfNuVGNBdPqWSRgsXoyzGEtHHioIm+jKPfdZI6TDHJ
frm8N2NCJh5itK5CW5G1xrOhS+HqUfUI/DLYVE2sM5FDsxfeqXoGS617UgJ26q+H
bgxTj3gSFtXeAvB8cIVUKn73wrCIwrmCkTv4jXDtApm9TJSYdy3YSZUosgsDRwYR
9VwJC+JG7q1y2S3fwZjXVgOKI4V87ySf+dgBWwVVoHTgZAM5lXF94vsCAwEAAaAh
MB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUA
A4IBAQAu70Kqtey8tsi2/v0X6UGGrO3FzhXwUO7gj+GY9Ta2KpuZMgM02YdqaGX4
wc53AS7btpiAnB7JwbfWlunGiL5IMmIdAoL1vTy5NjyL4ps+0JhAIYzgv9pxu6N1
c8X6YpF8fZgh+Y/q8bl/MLUghoFcytBKvx1CBeDaXJrxi4PYijr7ya1cIs0U3ydg
Y+8XkltfaC/UuYaxt6kxZtKvgW3ItfZU6GBM2LLFLQ9WBkqifHbTnPqv2QRtKxoC
o9jvUEZqH7BBwdDN2l3RnrwS5Yms1JnWUeG3ARll5ALXyt8zPtD5XFWh5390RY6M
FvW6jcrKrU/xHP7JvaNhtCu7zhQv
-----END CERTIFICATE REQUEST-----
```

8. Подпишите запрос на сертификат для Cisco:

```
root@Hub1:~/ca# openssl ca -config openssl.cnf -extensions server_cert -days 375 -
notext -md sha256 -in csr/cisco.pem -out certs/cisco.cer
```

В команде необходимо указать раздел `server_cert`, чтобы сертификат был выпущен с расширениями для удаленной аутентификации устройства.

```
Using configuration from openssl.cnf
Enter pass phrase for /root/ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4103 (0x1007)
        Validity
            Not Before: Jul 15 14:23:39 2024 GMT
            Not After : Jul 25 14:23:39 2025 GMT
        Subject:
            countryName               = RU
            localityName              = Zelenograd
            organizationName          = S-Terra CSP
            commonName                = Cisco-CSR100v
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                4B:DE:D4:DB:41:01:D9:C9:CE:FC:87:6F:8B:C1:A1:35:29:65:35:47
            X509v3 Authority Key Identifier:
                keyid:3C:6E:22:57:B8:50:A5:95:A5:0E:CC:95:78:02:04:49:7C:5C:28:CD
                DirName:/CN=S-Terra CSP Test Root CA-OpenSSL/C=RU/L=Zelenograd/O=S-
Terra CSP
                serial:77:0C:FD:38:EF:6F:E6:9B:B1:69:CE:48:FE:D6:F4:50:FF:01:36:A6
            X509v3 Key Usage:
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
Certificate is to be certified until Jul 25 14:23:39 2025 GMT (375 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
```

9. Импортируйте полученный сертификат от УЦ в устройство.

```
Cisco(config)#crypto pki import CA certificate
```

```
Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIEIjCCAwqgAwIBAgICEAcwDQYJKoZIhvcNAQELBQAwYzEpMCcGA1UEAwwgUy1U
ZXJyYSBDU1AgVGVzdCBSb290IENBLU9wZW5TU0wxCzAJBgNVBAYTAlJVMRMwEQYD
VQQHDApaZWxlbm9ncmFkMRQwEgYDVQQKDAtTLVRlcnJhIENTUDAeFw0yNDAyMDIx
MjU4MjVaFw0yNTAyMTExMjU4MjVaMFAxCzAJBgNVBAYTAlJVMRMwEQYDVQQHEwpa
ZWxlbm9ncmFkMRQwEgYDVQQKEwtTLVRlcnJhIENTUDEWMBQGA1UEAxMNQ2lzY28t
Q1NSMTAwdjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJj1jpcbLmAo
SDsb3bXmvugEECZeHdDHHBrvIrTzzhYmrnPyOGu0zowyls2RXsmDOdgXD+AQyzLn
IBt9faoN2sZhi4V2/5PqSFZ6gDZyoBMoY3H7n/gBTICfNuVGNBdPqWSRgsXoyzGE
tHHioIm+jKPfdZI6TDHJfrm8N2NCJh5itK5CW5G1xrOhS+HqUfUI/DLYVE2sM5FD
sxfeqXoGS617UgJ26q+HbgxTj3gSFtXeAvB8cIVUKn73wrCIwrmCkTv4jXDtApm9
TJSYdy3YSZUosgsDRwYR9VwJC+JG7q1y2S3fwZjXVgOKI4V87ySf+dgBWwVVoHTg
```

```
ZAM5lXF94vsCAwEAAaOB8jCB7zAJBgNVHRMEAjAAMB0GA1UdDgQWBBRL3tTbQQHZ
yc78h2+LwaE1KWU1RzCBoAYDVR0jBIGYMIGVgBSPHKdi7Z4T0MlrmraElH+mblnE
g6FnpGUwYzEpMCcGA1UEAwwgUy1UZXJyYSBDU1AgVGVzdCBSb290IENBLU9wZW5T
U0wxCzAJBgNVBAYTAlJVMRMwEQYDVQQHDApaZWxlbm9ncmFkMRQwEgYDVQQKDAtT
LVRlcnJhIENTUIIUAX7UbyRms1TINoKSIMbF0wqJME4wCwYDVR0PBAQDAgeAMBMG
A1UdJQQMMAoGCCsGAQUFBwMCMA0GCSqGSIb3DQEBCwUAA4IBAQDJK22qKaFpG5lJ
6vx1ZrPXNLdDXo7doVmcGasqPLMjB6nbzGkDsxzqJZhjDoLyLZC2wyX9HyiYaHPm
Bk9I+NYtPXVoR97onqUV1e0GJsmMhTUtspJEz6WshG5b+o3CdKr4fg5tYlxGFHX1
Ca/o1qQftw+Alo0Pezs8aWKwhK4v80YiQf2Z3bbkUkaF/Mw9dVvWnYlJ/W2O4yTg
zj8N4oV+h2DkFYCRVHs8vHyCQ9kI9bklDefxcJwNaiaHrnY/Ut2wPrgMce7VbGke
oDc7E45bT18IVs7JL2XFWZhRghPHPxXRBvQc2esExVA813KWrawjJYyb+vHLB5rQ
tXFHZif3
-----END CERTIFICATE-----

% Router Certificate successfully imported
```

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

```
Cisco#show crypto pki certificates
```

```
Certificate
  Status: Available
  Certificate Serial Number (hex): 1001
  Certificate Usage: General Purpose
  Issuer:
    ou=RnD
    o=S-Terra CSP
    l=Zelenograd
    c=RU
    cn=S-Terra CSP Test Root CA-OpenSSL
  Subject:
    Name: Cisco-CSR100v
    cn=Cisco-CSR100v
    o=S-Terra CSP
    l=Zelenograd
    c=RU
  Validity Date:
    start date: 14:23:39 CST Jul 15 2024
    end   date: 14:23:39 CST Jul 25 2025
  Associated Trustpoints: CA

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 770CFD38EF6FE69BB169CE48FED6F450FF0136A6
  Certificate Usage: Signature
  Issuer:
    ou=RnD
    o=S-Terra CSP
    l=Zelenograd
    c=RU
    cn=S-Terra CSP Test Root CA-OpenSSL
  Subject:
    ou=RnD
    o=S-Terra CSP
    l=Zelenograd
    c=RU
    cn=S-Terra CSP Test Root CA-OpenSSL
  Validity Date:
    start date: 17:19:52 CST Feb 7 2024
    end   date: 17:19:52 CST Feb 2 2044
  Associated Trustpoints: CA
```

Настройка шифрования:

1. Задайте алгоритмы, которые будут задействованы для установления/защиты обмена сообщениями IKE_SA_INIT:

```
Cisco(config)#crypto ikev2 proposal 1
```

```
IKEv2 proposal MUST either have a set of an encryption algorithm other than aes-gcm,
an integrity algorithm and a DH group configured or
 encryption algorithm aes-gcm, a prf algorithm and a DH group configured
```

```
Cisco(config-ikev2-proposal)#encryption aes-cbc-256
Cisco(config-ikev2-proposal)#integrity sha256
Cisco(config-ikev2-proposal)#group 14
Cisco(config-ikev2-proposal)#exit
```

2. Создайте политику IKEv2, которая содержит уже созданный `proposal`:

```
Cisco(config)#crypto ikev2 policy 1
```

```
IKEv2 policy MUST have atleast one complete proposal attached
```

```
Cisco(config-ikev2-policy)#proposal 1
```

3. Определите методы аутентификации в `profile`:

```
Cisco(config)#crypto ikev2 profile CISCO_ALGS
```

```
IKEv2 profile MUST have:
   1. A local and a remote authentication method.
   2. A match identity or a match certificate or match any statement.
```

```
Cisco(config-ikev2-profile)#description IKEv2 Cisco profile
Cisco(config-ikev2-profile)#match identity remote any
Cisco(config-ikev2-profile)#authentication remote rsa-sig
Cisco(config-ikev2-profile)#authentication local rsa-sig
Cisco(config-ikev2-profile)#pki trustpoint CA
Cisco(config-ikev2-profile)#dpd 10 5 periodic
Cisco(config-ikev2-profile)#exit
```

4. Задайте комбинированный алгоритм шифрования и имитозащиты (набор преобразований) для трафика:

```
Cisco(config)#crypto ipsec transform-set ENCRYPT_AND_INTEGRITY esp-aes 256 esp-sha256-
hmac
Cisco(config)#exit
```

5. Укажите фрагментацию для IKEv2 соединения:

```
Cisco(config)#crypto ikev2 fragmentation mtu 1024
```

6. Создайте список доступа (ACL) для трафика, который нужно защищать:

```
Cisco(config)#ip access-list extended IPSEC_ACL_CISCO_AND_HUB1
Cisco(config-ext-nacl)#10 permit ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
```

7. Создайте крипто-карту (имя VPN раздел 1):

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

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

8. Укажите список доступа для защищаемого трафика:

```
Cisco(config-crypto-map)#match address IPSEC_ACL_CISCO_AND_HUB1
```

9. Укажите при помощи какого набора алгоритмов нужно защищать трафик:

```
Cisco(config-crypto-map)#set transform-set ENCRYPT_AND_INTEGRITY
```

10. Укажите IP адрес партнера по IPsec, в данном сценарии это внешний IP адрес криптошлюза:

```
Cisco(config-crypto-map)#set peer 10.0.102.25
```

11. Укажите методы аутентификации:

```
Cisco(config-crypto-map)# set ikev2-profile CISCO_ALGS
Cisco(config-crypto-map)# exit
```

12. Прикрепите созданную крипто-карту VPN к внешнему интерфейсу `GigabitEthernet1`:

```
Cisco(config)#interface GigabitEthernet1
Cisco(config-if)#crypto map VPN
```

В Приложении представлен текст конфигураций для Cisco.

Настройка StrongSwan

Настройка сетевых параметров

1. Задайте IP адреса в соответствии со схемой стенда на внешнем `ens33` и внутреннем `ens34` интерфейсах. Для этого напишите следующее содержимое в файл `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet dhcp

allow-hotplug ens34
iface ens34 inet static
address 192.168.2.1
netmask 255.255.255.0
```

2. Перезагрузите сетевые настройки при помощи команды:

```
root@StrongSwan:~#systemctl restart networking.service
```

3. Включите маршрутизацию на устройстве со StrongSwan. Для этого раскомментируйте следующую строку в файле `/etc/sysctl.conf`:

```
net.ipv4.ip_forward=1
```

4. Выйдите из редактора с сохранением файла, затем выполните в командной строке команду:

```
root@StrongSwan:~#sysctl -p /etc/sysctl.conf
```

Установка StrongSwan

Установите StrongSwan и дополнительные пакеты при помощи команд:

```
root@StrongSwan:~# apt update
root@StrongSwan:~# apt install strongswan strongswan-pki libcharon-extra-plugins libcharon-extauth-plugins
```

Настройки PKI (запросы и сертификаты)

1. Создайте пару RSA-ключей (открытый и закрытый) с размером 2048 бит и сохраните их в файл с именем `strongSwanKey.pem` в формате PEM:

```
root@StrongSwan:~#pki --gen --type rsa --size 2048 --outform pem > /etc/ipsec.d/private/strongSwanKey.pem
```

```
plugin 'test-vectors': failed to load - test_vectors_plugin_create not found and no plugin file available
plugin 'pkcs11': failed to load - pkcs11_plugin_create not found and no plugin file available
plugin 'tpm': failed to load - tpm_plugin_create not found and no plugin file available
plugin 'rdrand': failed to load - rdrand_plugin_create not found and no plugin file available
plugin 'gcrypt': failed to load - gcrypt_plugin_create not found and no plugin file available
plugin 'af-alg': failed to load - af_alg_plugin_create not found and no plugin file available
```

```
plugin 'curve25519': failed to load - curve25519_plugin_create not found and no plugin
file available
plugin 'curl': failed to load - curl_plugin_create not found and no plugin file
available
```

Примечание – После выполнения команды появятся сообщения об отсутствии плагинов, на генерацию ключа это не повлияет.

2. Используйте закрытый ключ из файла `strongSwanKey.pem` для создания запроса на сертификат с указанными данными о субъекте (DN - Distinguished Name) и сохраните результат в файл `strongSwanReq.pem` в формате PEM:

```
root@StrongSwan:~#pki --req --type priv --in /etc/ipsec.d/private/strongSwanKey.pem --dn "C=RU,L=Zelenograd,O=S-Terra CSP,CN=StrongSwan" --outform pem > /etc/ipsec.d/reqs/strongSwanReq.pem
```

```
plugin 'test-vectors': failed to load - test_vectors_plugin_create not found and no
plugin file available
plugin 'pkcs11': failed to load - pkcs11_plugin_create not found and no plugin file
available
plugin 'tpm': failed to load - tpm_plugin_create not found and no plugin file available
plugin 'rdrand': failed to load - rdrand_plugin_create not found and no plugin file
available
plugin 'gcrypt': failed to load - gcrypt_plugin_create not found and no plugin file
available
plugin 'af-alg': failed to load - af_alg_plugin_create not found and no plugin file
available
plugin 'curve25519': failed to load - curve25519_plugin_create not found and no plugin
file available
plugin 'curl': failed to load - curl_plugin_create not found and no plugin file
available
```

3. После выполнения этой команды перенесите запрос на сертификат в кодировке BASE64 со StrongSwan под именем `strongswan.pem` на криптошлюз по пути `/root/ca/csr/` и подпишите запрос на сертификат для StrongSwan:

```
root@Hub1:~/ca#openssl ca -config openssl.cnf -extensions server_cert -days 375 -notext -md sha256 -in csr/strongswan.pem -out certs/strongswan.cer
```

В команде необходимо указать раздел `server_cert`, чтобы сертификат был выпущен с расширениями для удаленной аутентификации устройства.

```
Using configuration from openssl.cnf
Enter pass phrase for /root/ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Jul 29 01:28:05 2024 GMT
            Not After : Aug  8 01:28:05 2025 GMT
        Subject:
            countryName               = RU
            localityName              = Zelenograd
            organizationName          = S-Terra CSP
            organizationalUnitName    = RnD
            commonName                = StrongSwan
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                CB:7F:B6:A8:03:D3:85:BB:31:B8:8A:8B:FE:A7:0D:8B:68:3F:21:99
            X509v3 Authority Key Identifier:
                keyid:3C:6E:22:57:B8:50:A5:95:A5:0E:CC:95:78:02:04:49:7C:5C:28:CD
```

```
                DirName:/CN=S-Terra CSP Test Root CA-OpenSSL/C=RU/L=Zelenograd/O=S-
Terra CSP/OU=RnD
                serial:77:0C:FD:38:EF:6F:E6:9B:B1:69:CE:48:FE:D6:F4:50:FF:01:36:A6
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Aug  8 01:28:05 2025 GMT (375 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
```

4. Перенесите выпущенный сертификат под именем `strongswan.cer` в директорию `/etc/ipsec.d/certs/` и сертификат УЦ под именем `ca.cert.cer` в директорию `/etc/ipsec.d/cacerts/`.

Чтобы импортировать выпущенный сертификат в базу StrongSwan, необходимо, чтобы сертификат `strongswan.cer` находился в директории `/etc/ipsec.d/certs`, сертификат УЦ находился в директории `/etc/ipsec.d/cacerts/` и закрытый ключ `strongSwanKey.pem` в директории `/etc/ipsec.d/private/`.

5. Укажите имя закрытого ключа `strongSwanKey.pem` для сертификата `strongswan.cer` в файле `/etc/ipsec.secrets` для того, чтобы добавить закрытый ключ для локального сертификата:

```
# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.
: RSA "strongSwanKey.pem"
```

В Приложении представлен текст конфигураций для StrongSwan.

6. Убедитесь, что сертификат УЦ успешно импортирован в базу Продукта, сертификат устройства можно будет посмотреть только после перезагрузки конфигурации StrongSwan:

```
root@StrongSwan:~#ipsec restart
root@StrongSwan:~#ipsec listcacerts
```

```
 List of X.509 CA Certificates

  subject:  "CN=S-Terra CSP Test Root CA-OpenSSL, C=RU, L=Zelenograd, O=S-Terra CSP,
OU=RnD"
  issuer:   "CN=S-Terra CSP Test Root CA-OpenSSL, C=RU, L=Zelenograd, O=S-Terra CSP,
OU=RnD"
  validity:  not before Feb 07 17:19:52 2024, ok
             not after  Feb 02 17:19:52 2044, ok (expires in 7127 days)
  serial:    77:0c:fd:38:ef:6f:e6:9b:b1:69:ce:48:fe:d6:f4:50:ff:01:36:a6
  flags:     CA CRLSign self-signed
  authkeyId: 3c:6e:22:57:b8:50:a5:95:a5:0e:cc:95:78:02:04:49:7c:5c:28:cd
  subjkeyId: 3c:6e:22:57:b8:50:a5:95:a5:0e:cc:95:78:02:04:49:7c:5c:28:cd
  pubkey:    RSA 2048 bits
  keyid:     56:cc:56:be:bd:2c:e6:42:84:e6:42:d7:5b:c0:87:92:fd:d1:94:76
  subjkey:   3c:6e:22:57:b8:50:a5:95:a5:0e:cc:95:78:02:04:49:7c:5c:28:cd
```

Примечание – Сертификат устройства можно будет посмотреть с помощью команды ниже только после настройки конфигурационного файла `/etc/ipsec.conf`, до этого сертификат отображаться не будет.

```
root@StrongSwan:~# ipsec listcerts
```

Настройка шифрования:

1. Отредактируйте файл `/etc/ipsec.conf`, который отвечает за настройку StronSwan:

   1.1. Задайте общие параметры защищенного соединения:

```
config setup
        charondebug="all"
        uniqueids=no
conn %default
        type=tunnel
        auto=route
        keyexchange=ikev2
        keyingtries=%forever
        ikelifetime=86400s
        lifetime=3600s
        dpddelay=30s
        dpdtimeout=120s
        dpdaction=restart
        fragmentation=yes
```

   1.2. Необходимо указать метод аутентификации при помощи цифровых сертификатов:

```
        authby=rsasig
```

   1.3. Добавьте IP адрес внешних интерфейсов партнеров по IPsec и их защищаемые подсети (`left` - IP адрес StrongSwan), а также укажите имя локального сертификата. Для партнера задайте прием любого ID:

```
        left=10.0.108.237
        leftsubnet=192.168.2.0/24
        leftcert=strongswan.cer
```

   1.4. Укажите параметры IKE:

```
        ike=aes256-sha256-prfsha256-modp2048!
```

   1.5. Укажите параметры IPsec:

```
        esp=aes256-sha256-modp2048!
```

   1.6. Отключите протокол по умолчанию MOBIKE:

```
        mobike=no
```

   Примечание – Отключение протокола по умолчанию MOBIKE связано с особенностью реализации MOBIKE в StrongSwan. Поведение StrongSwan отличается от стандарта. В криптошлюзе нет поддержки реализации StrongSwan протокола MOBIKE.

   1.7. Создайте отдельное соединение с Hub1 с подсетью, которую нужно защищать:

```
conn StronSwan-Hub1-Cert
        rightid=%any
        right=10.0.102.25
        rightsubnet=192.168.100.0/24
```

   В Приложении представлен текст конфигураций `/etc/ipsec.conf` для StrongSwan.

2. Перезапустите StongSwan демон:

```
root@StrongSwan:~#ipsec restart
```

```
Stopping strongSwan IPsec...
Starting strongSwan 5.9.8 IPsec [starter]...
```

Созданные в конфигурационном файле соединения будут активны при наличии трафика между защищенными подсетями.

Настройка Android

Настройки PKI (запросы и сертификаты)

Создание и экспорт сертификата для мобильного клиента с криптошлюза

Создание запроса, выпуск и экспорт сертификата в формате pkcs12 для мобильного клиента происходит на криптошлюзе. Сертификат будет подписан с использованием УЦ на базе OpenSSL, настроенного ранее на криптошлюзе в разделе «Настройки PKI (запросы и сертификаты) на Hub1».

1. Запрос для выпуска сертификата для Android будет сгенерирован при помощи УЦ на базе OpenSSL на криптошлюзе. Запустите процесс генерации закрытого ключа, введите пароль:

```
root@Hub1:~/ca#openssl genrsa -aes256 -out private/android.key.pem 2048
```

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

2. Создайте запрос на сертификат Android с сохранением файла запроса:

```
root@Hub1:~/ca#openssl req -config  openssl.cnf -key  private/android.key.pem -new -sha256 -out csr/android.csr.pem
```

```
Enter pass phrase for private/android.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name []:Android
Country Name (2 letter code) []:RU
State or Province Name []:
Locality Name []:Zelenograd
Organization Name []:S-Terra CSP
Organizational Unit Name []:
Email Address []:
```

3. Выпустите сертификат для Android по сгенерированному запросу на сертификат на УЦ OpenSSL:

В команде необходимо указать раздел `usr_cert`, чтобы сертификат был выпущен с необходимыми расширениями. В сертификате должно быть расширение с полем «Улучшенный ключ» и значением «Проверка подлинности клиента», необходимое для установления защищенного соединения IPsec с «С-Терра Шлюз». Все необходимые расширения для сертификата клиента указаны в файле конфигурации `openssl.cnf` в разделе `usr_cert`.

```
root@Hub1:~/ca#openssl ca -config openssl.cnf -extensions usr_cert -days 375 -notext -md sha256 -in csr/android.csr.pem -out certs/android.cert.pem
```

```
 Using configuration from openssl.cnf
Enter pass phrase for /root/ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4099 (0x1003)
        Validity
            Not Before: Jul 29 01:30:59 2024 GMT
            Not After : Aug  8 01:30:59 2025 GMT
        Subject:
            countryName               = RU
            localityName              = Zelenograd
            organizationName          = S-Terra CSP
            commonName                = Android
```

```
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                9D:67:C7:5C:09:87:C8:E8:CA:06:1B:37:C0:03:D4:2B:A6:9C:5B:C5
            X509v3 Authority Key Identifier:
                keyid:3C:6E:22:57:B8:50:A5:95:A5:0E:CC:95:78:02:04:49:7C:5C:28:CD
                DirName:/CN=S-Terra  CSP  Test  Root  CA-OpenSSL/C=RU/L=Zelenograd/O=S-
Terra CSP/OU=RnD
                serial:77:0C:FD:38:EF:6F:E6:9B:B1:69:CE:48:FE:D6:F4:50:FF:01:36:A6
            X509v3 Key Usage:
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
Certificate is to be certified until Aug  8 01:30:59 2025 GMT (375 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
```

4. Создайте сертификат в формате pkcs12 для Android и введите пароль от ключа и пароль для экспорта сертификата:

Параметр `-legacy` необходим для импорта контейнера на мобильных клиентов, без него не будет принят пароль. Из-за особенности обновления безопасности необходимо пользоваться устаревшими алгоритмами, чтобы пароль от контейнера сработал. На Android версии 14 и новее параметр `-legacy` можно опустить. Также можно не использовать данный параметр для Windows.

```
root@Hub1:~/ca#openssl pkcs12 -export -out certs/Android.pfx -inkey
private/android.key.pem -in certs/android.cert.pem -legacy
```

Из-за особенностей импорта пароль для экспорта сертификата должен содержать латинские буквы или цифры. Пароль, в котором содержится кириллица, может вызвать ошибку.

```
Enter pass phrase for private/android.key.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

Импорт сертификатов на Android

Для импорта сертификатов на Android выполните описанные ниже шаги:

1. Перенесите сертификаты на Android.
2. Для импорта сертификата мобильного клиента на Android с закрытым ключем выполните следующие пункты:

   2.1. Перейдите в настройки Android и в поиске найдите «Сертификаты безопасности» (см. Рисунок 2):

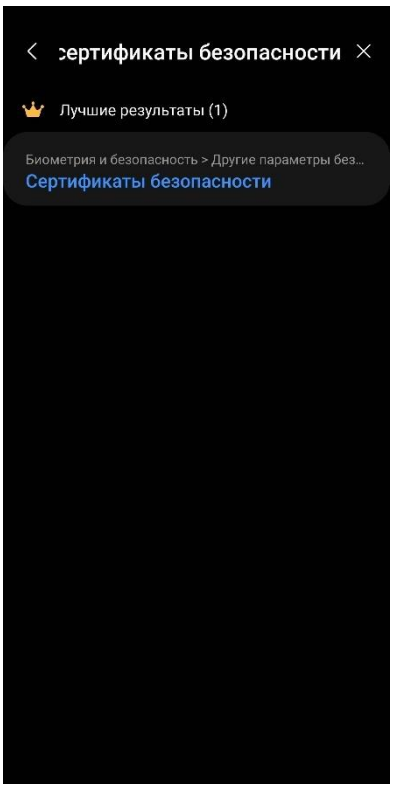


Рисунок 2

2.2. Найдите в настройках в списке параметров безопасности «Установка из памяти» и нажмите на этот пункт (см. Рисунок 3):

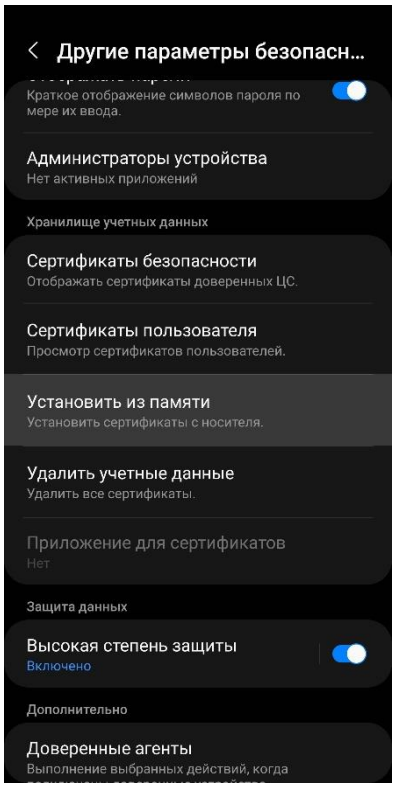


Рисунок 3

2.3. Выберите пункт «Сертификат VPN и приложения» и выберите скаченный сертификат в формате pkcs12 (см. Рисунок 4).

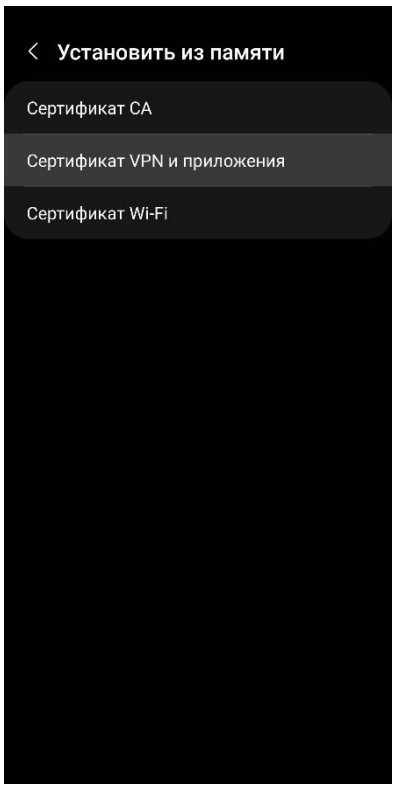


Рисунок 4

2.4. Введите пароль, который был указан на криптошлюзе при соответсвующих настройках для импорта сертификата и нажмите «OK» (см. Рисунок 5).

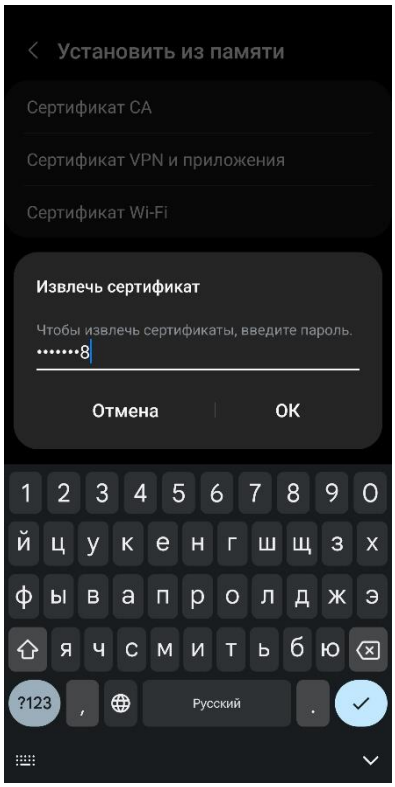


Рисунок 5

2.5. Задайте имя импортируемого сертификата и нажмите «OK» (см. Рисунок 6):

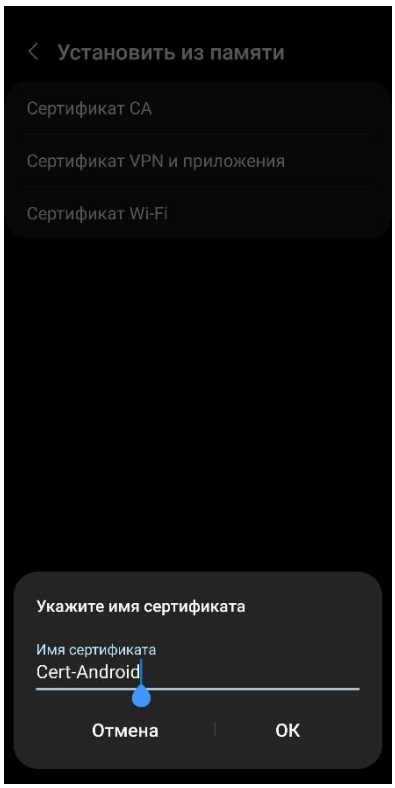


Рисунок 6

2.6. Обратите внимание, что в результате должно появиться сообщение об успешном импорте сертификата (см. Рисунок 7):

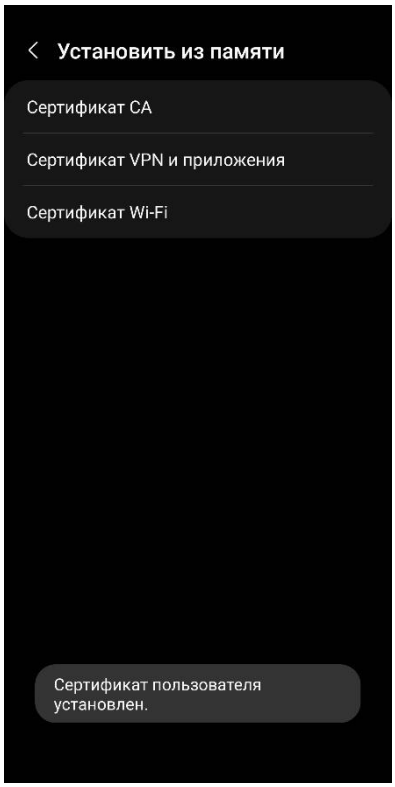


Рисунок 7

3. Для импорта сертификата УЦ на Android выполните следующие пункты:

3.1. Чтобы установить сертификат УЦ еще раз перейдите в пункт «Установить из памяти» (см. Рисунок 8):

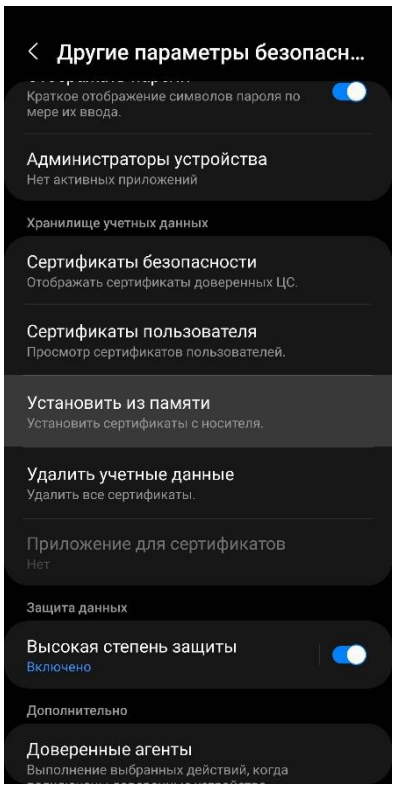


Рисунок 8

3.2. Выберите из списка «Сертификат CA» (см. Рисунок 9):

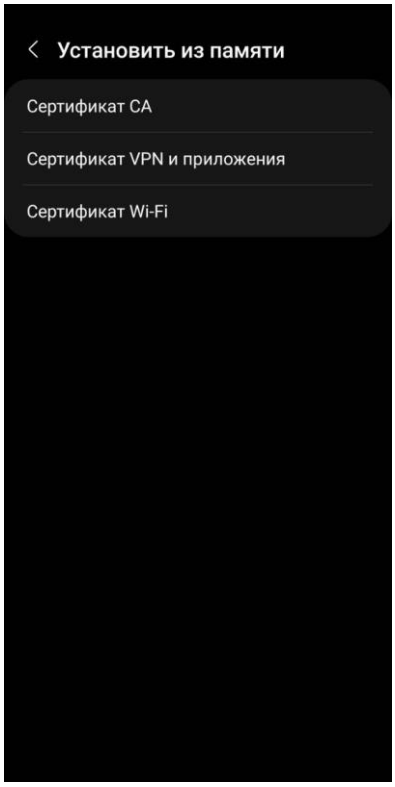


Рисунок 9

3.3. Ознакомтесь с информацией и выберите из списка «Установить в любом случае» (см. Рисунок 10):

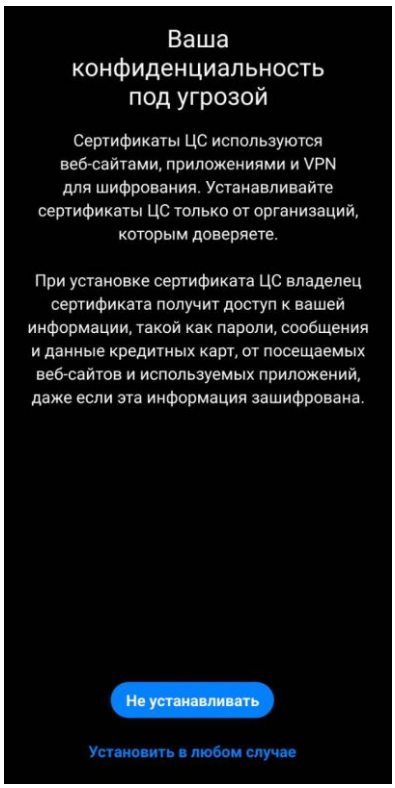


Рисунок 10

3.4. Выберите папку, в которой находится сертификат УЦ, и выберите его (см. Рисунок 11):

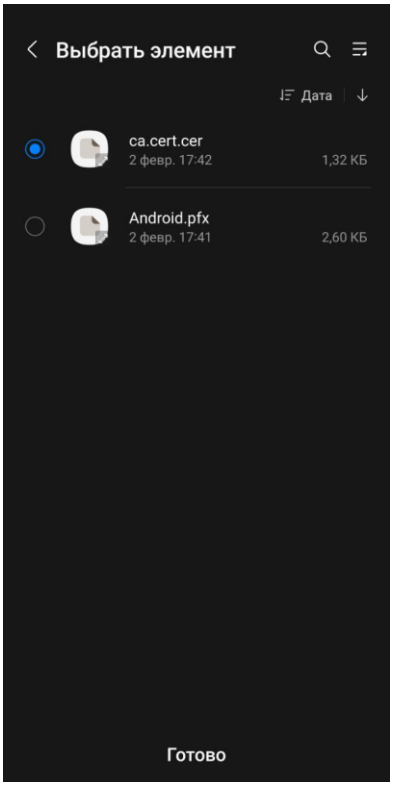


Рисунок 11

3.5. При успешном импорте выйдет сообщение «Сертификат СА установлен» (см. Рисунок 12):

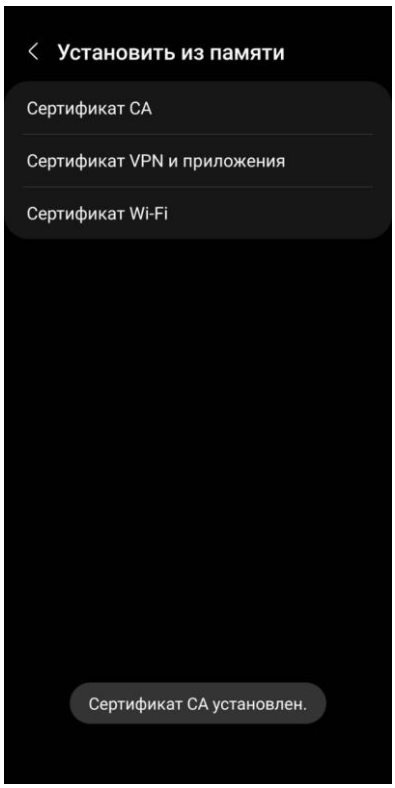


Рисунок 12

Создание VPN профиля на Android

Предлагаемый метод использования VPN-профиля на Android реализован с помощью встроенных профилей устройства.

1. Найдите в настройках «VPN» при помощи поиска и перейдите по результату (см. Рисунок 13):

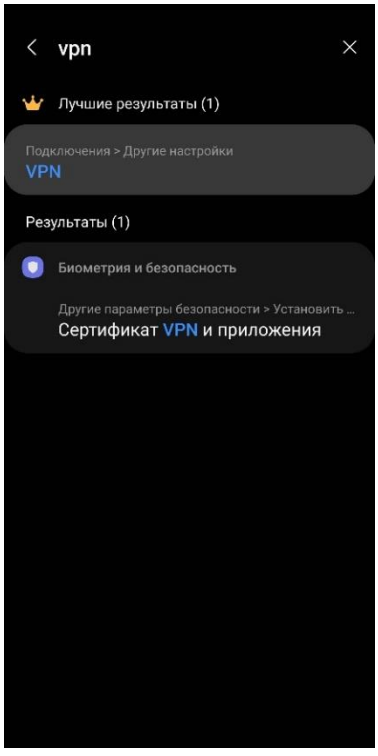


Рисунок 13

2. Перейдите во вкладку «VPN» (см. Рисунок 14):

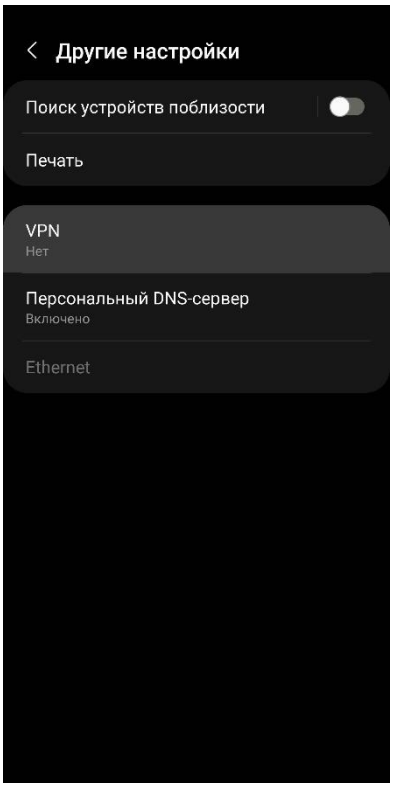


Рисунок 14

3. Нажмите в правом верхнем углу на троеточие (см. Рисунок 15):

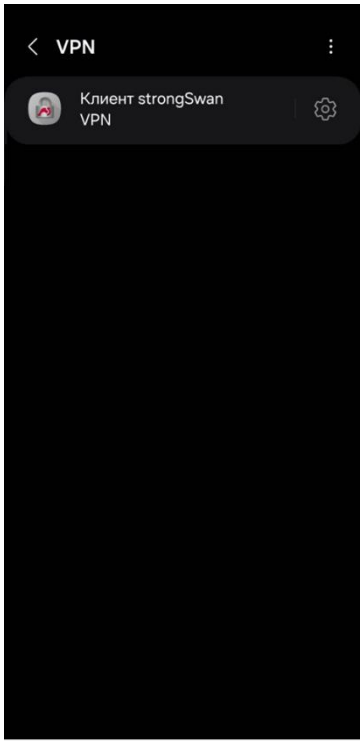


Рисунок 15

4. Нажмите на кнопку «Добавить профиль VPN» (см. Рисунок 16):

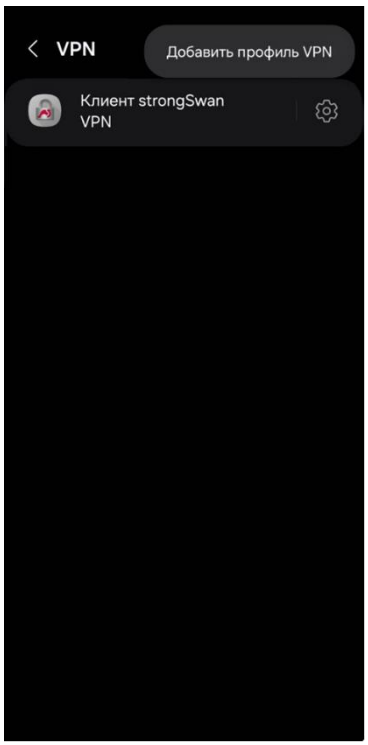


Рисунок 16

5. Выполните следующие действия:
   5.1 Введите «Имя».
   5.2 Укажите тип соединения «IKEv2/IPSec RSA».
   5.3 Введите IP адрес криптошлюза в поле «Адрес сервера»: 10.0.102.25.
   5.4 Укажите импортированный «Сертификат пользователя IPSec».
   5.5 Укажите импортированный сертификат УЦ в выборе меню «Сертификат ЦС IPSec».
   5.6 Остальное оставьте по умолчанию (см. Рисунок 17 и Рисунок 18).

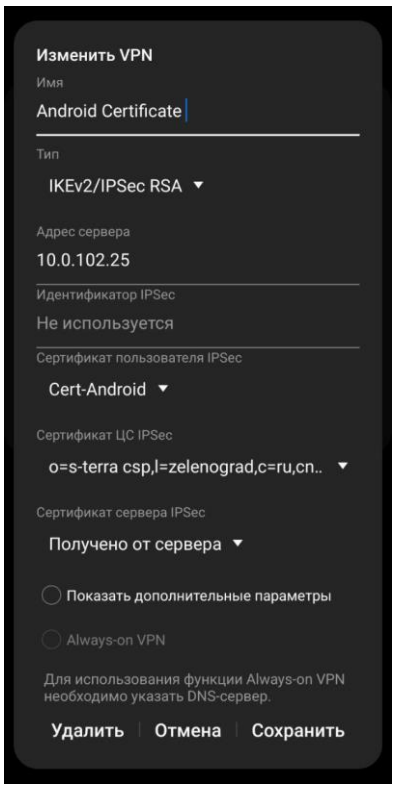


Рисунок 17

Если вам необходимо настроить split tunneling, чтобы только определенный трафик проходил через защищенное соединение IPsec, то необходимо прописать это, например, для данного стенда строка перенаправления маршрутов будет выглядеть следующим образом:

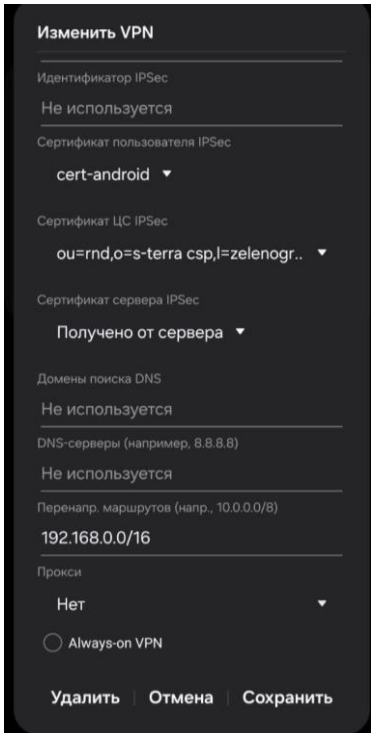


Рисунок 18

Создание VPN профиля на Android при помощи «StrongSwan VPN Client»

Предлагаемый метод использования VPN-профиля на Android реализован с помощью стороннего VPN-клиента «StrongSwan VPN Client» для случаев, когда невозможно подключиться через VPN-профиль на Android, реализованный с помощью встроенных профилей устройства.

1. Скачайте и войдите в приложение «StrongSwan VPN Client». Нажмите кнопку «Добавьте VPN-профиль» и настройте следующее:
   1.1. Введите IP адрес криптошлюза в поле «Сервер»: 10.0.102.25.
   1.2. Укажите тип соединения «IKEv2 Сертификат».
   1.3. Укажите импортированный «Сертификат пользователя».
   1.4. Уберите галочку «Выбрать автоматически» в меню «Сертификат CA». Укажите импортированный сертификат УЦ в выборе меню «Сертификат ЦС IPSec».
   1.5. Укажите название профиля (см. Рисунок 19 и Рисунок 20).

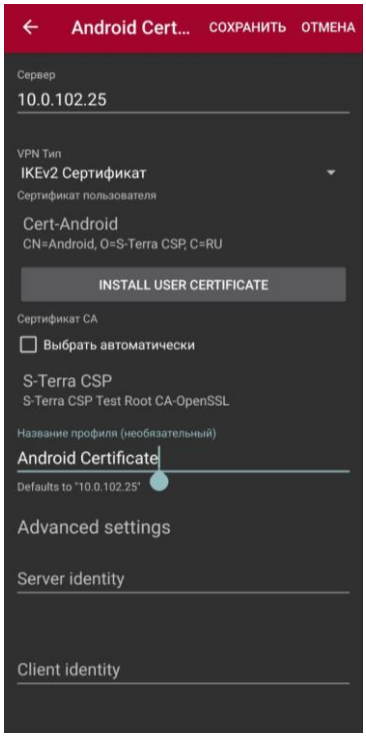


Рисунок 19

Если необходимо настроить split tunneling, чтобы только определенный трафик был направлен через защищенное соединение IPsec, нужно будет указать это явно. Например, для данного стенда строка «Custom subnets» будет иметь следующий вид:



Рисунок 20

1.6. Сохраните созданный VPN профиль. Для этого нажмите кнопку «Сохранить», расположенную вверху справа (см. Рисунок 21):

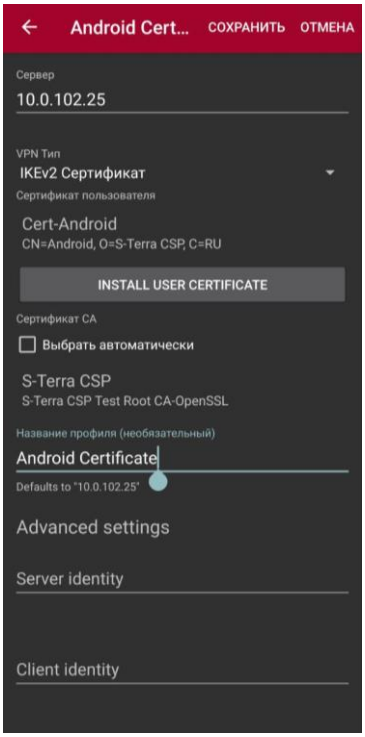


Рисунок 21

Настройка Windows

Настройки PKI (запросы и сертификаты)

В Приложении описан более частный способ настройки PKI с использованием УЦ на базе MSCA. Однако его использование не рекомендуется, так как в этом случае сертификат и закрытый ключ генерируются на клиенте, а не импортируются в формате pkcs12.

Создание и экспорт сертификата для мобильного клиента с криптошлюза

Настройка PKI (создание и экспорт сертификата для мобильного клиента) на Windows происходит аналогично настройке в разделе «Создание и экспорт сертификата для мобильного клиента с криптошлюза».

Импорт сертификатов на Windows

Для импорта сертификатов на Windows выполните описанные ниже шаги:

1. Перенесите сертификаты на Windows.
2. Для импорта сертификата мобильного клиента на Windows с закрытым ключем выполните следующие пункты:

   2.1. Перейдите в каталог, куда были перенесены сертификаты (см. Рисунок 22).

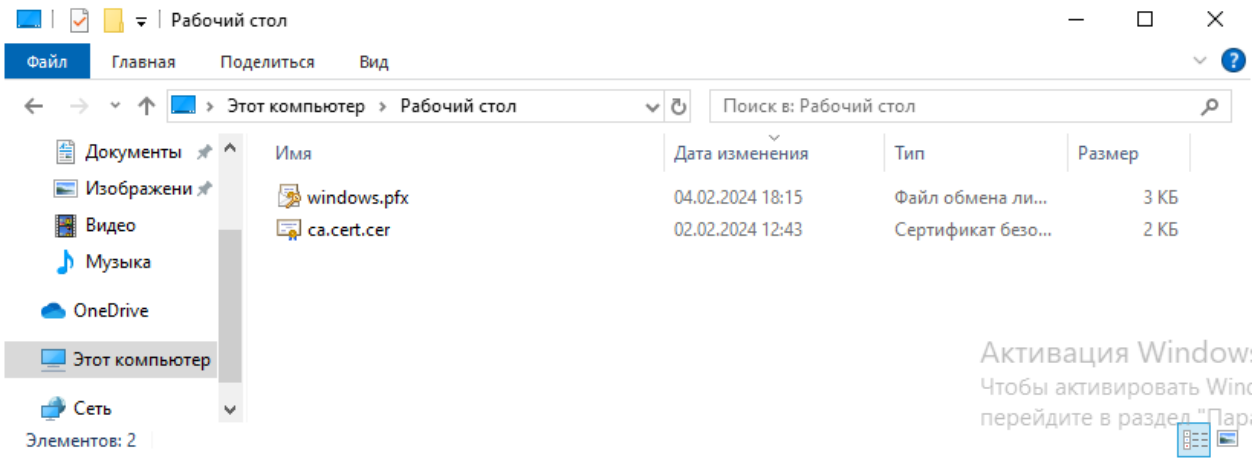


Рисунок 22

   2.2. Для импорта щелкните левой кнопкой мыши по файлу `windows`.

   2.3. Установите флажок на поле «Локальный компьютер» и нажмите «Далее» (см. Рисунок 23):

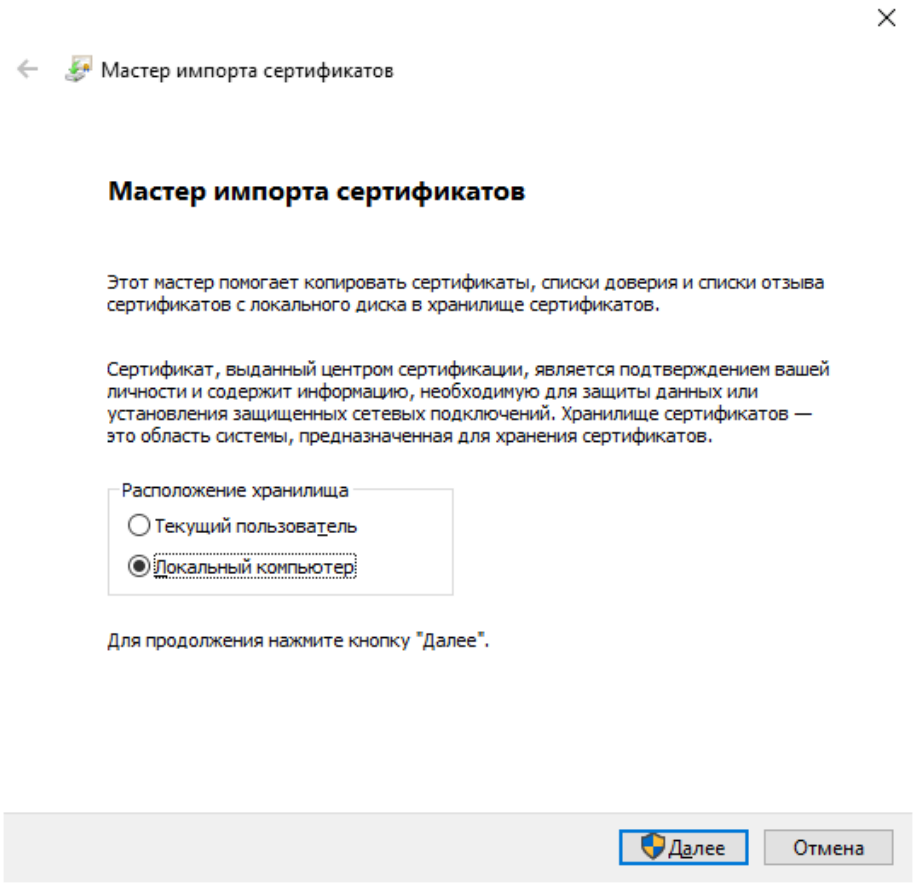


Рисунок 23

2.4. Разрешите этому приложению вносить изменения на вашем устройстве, для этого нажмите кнопку «Да» (см. Рисунок 24):

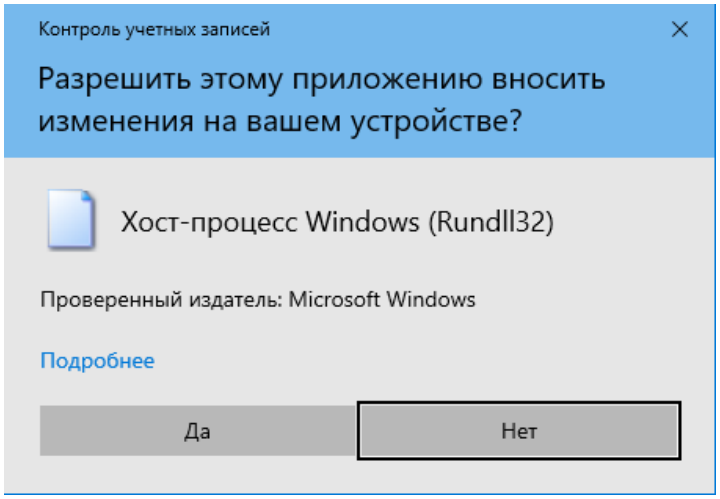


Рисунок 24

2.5. Подтвердите, что хотите импортировать данный файл, и нажмите кнопку «Далее» (см. Рисунок 25):

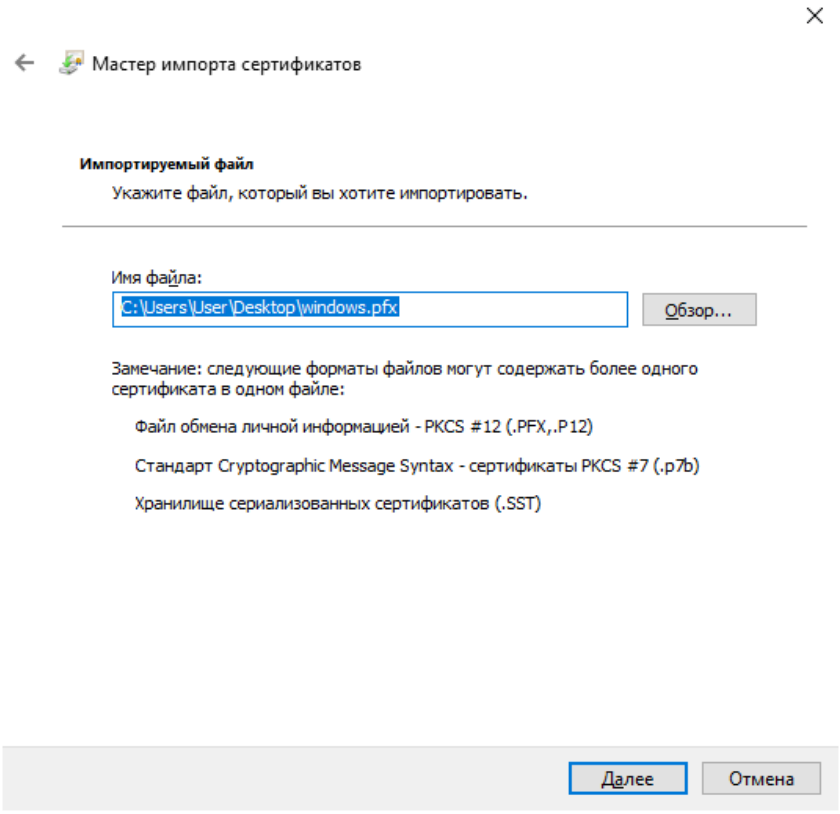


Рисунок 25

2.6. Введите пароль для закрытого ключа и оставьте все флаги по умолчанию. Нажмите кнопку «Далее» (см. Рисунок 26):

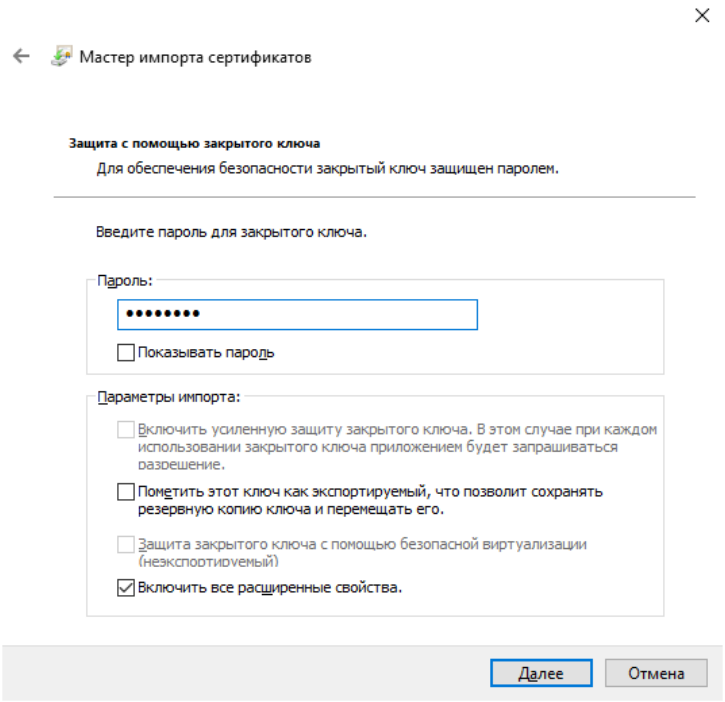


Рисунок 26

2.7. Оставьте выбор по умолчанию и нажмите «Далее» (см. Рисунок 27):

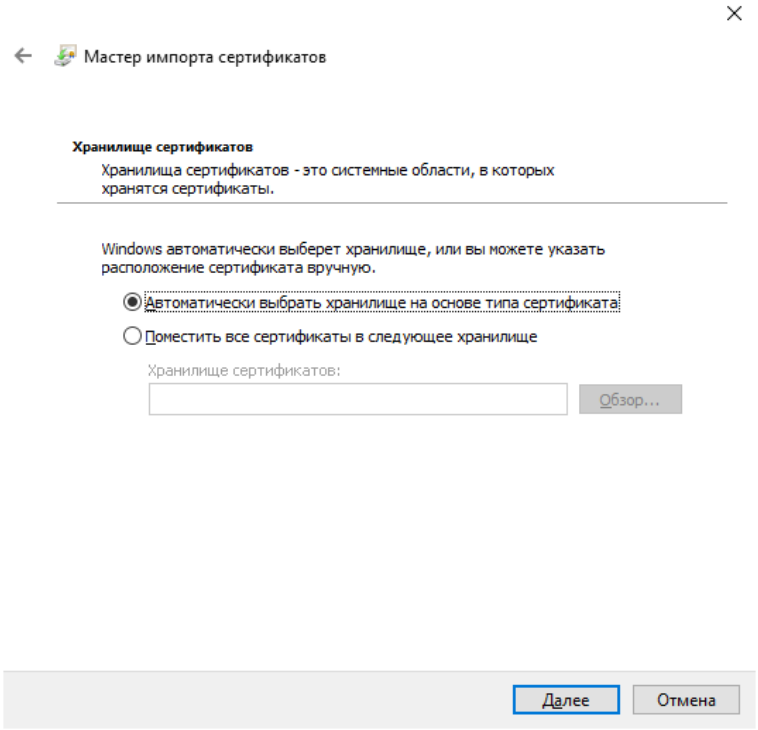


Рисунок 27

2.8. Чтобы завершить импорт, нажмите «Готово» (см. Рисунок 28):

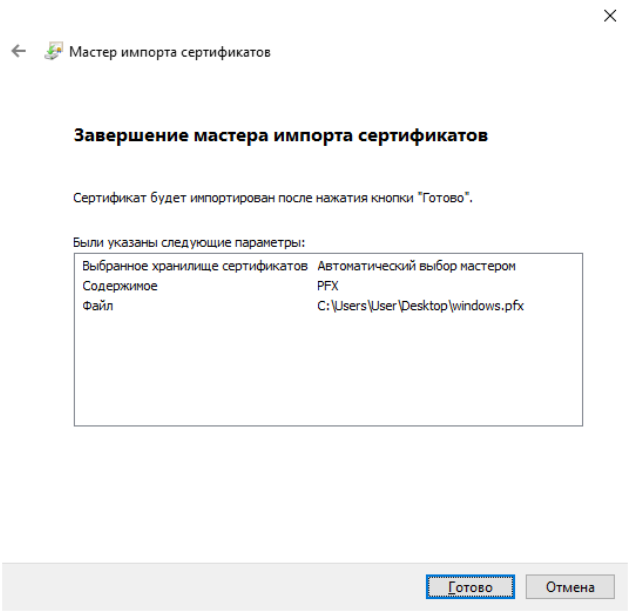


Рисунок 28

2.9. После успешного импорта сертификата появится окно об успешном импорте сертификата (см. Рисунок 29):

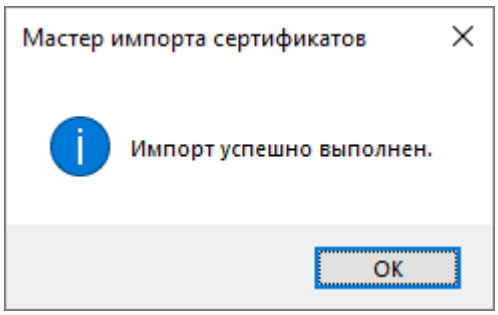


Рисунок 29

3. Для импорта сертификата УЦ на Windows выполните следующие пункты:

   3.1. Перейдите в каталог, куда были перенесены сертификаты (см. Рисунок 22).

   3.2. Импортируйте сертификат УЦ OpenSSL, для этого дважды щелкните левой кнопкой мыши по файлу `ca.cert`. После нажмите кнопку «Установить сертификат…» (см. Рисунок 30):

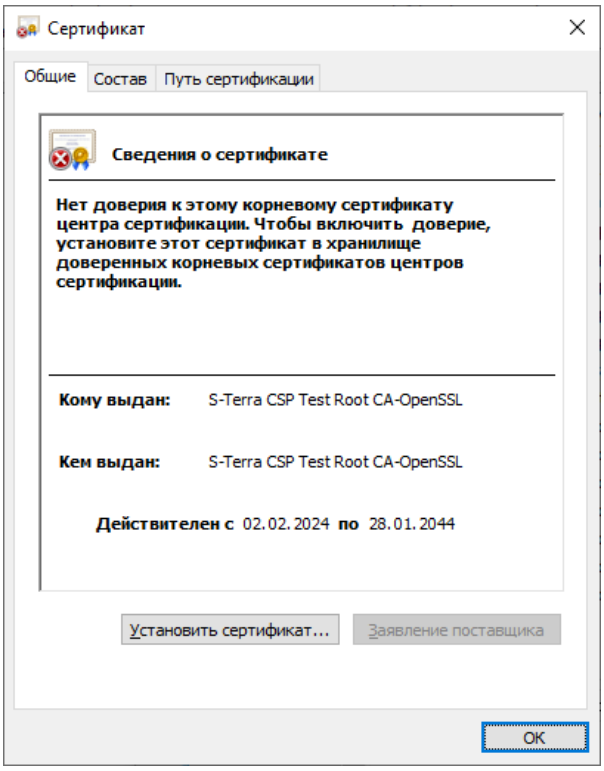


Рисунок 30

   3.3. Установите флажок на поле «Локальный компьютер» и нажмите «Далее» (см. Рисунок 31):

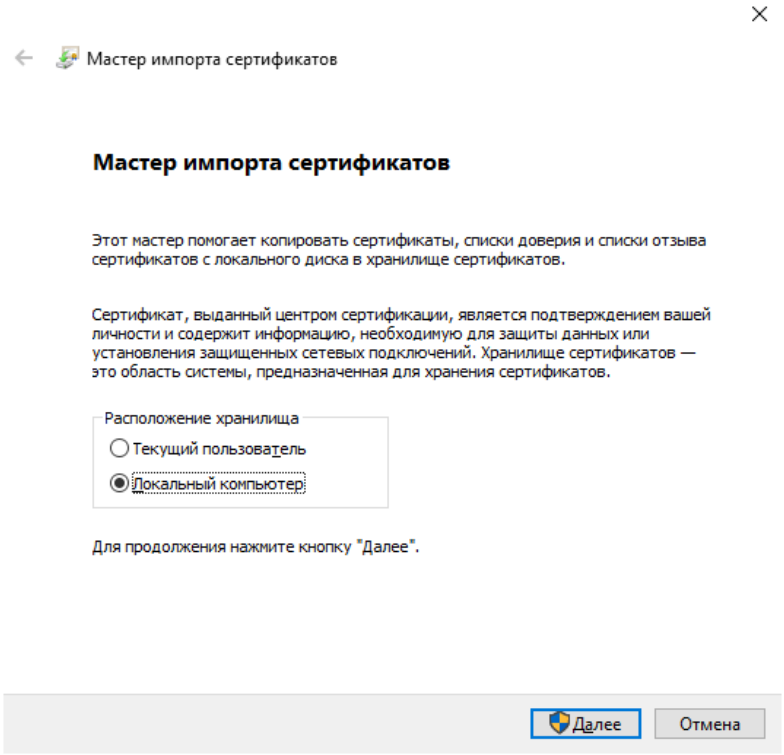


Рисунок 31

3.4. Разрешите этому приложению вносить изменения на вашем устройстве, для этого нажмите кнопку «Да» (см. Рисунок 32):

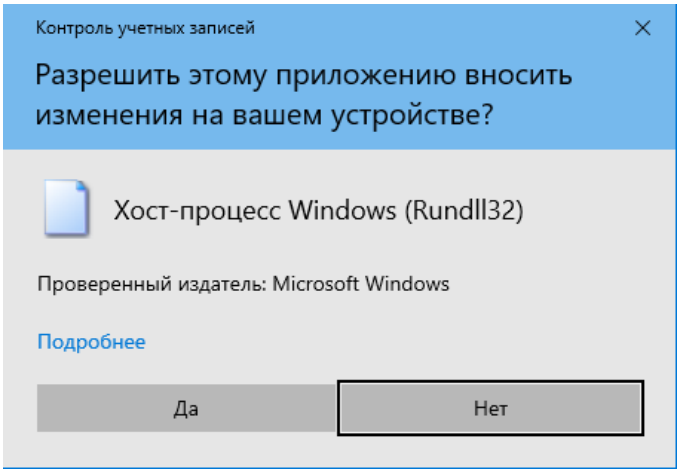


Рисунок 32

3.5. Установите флажок на поле «Поместить все сертификаты в следующее хранилище» и нажмите кнопку «Обзор…» (см. Рисунок 33):

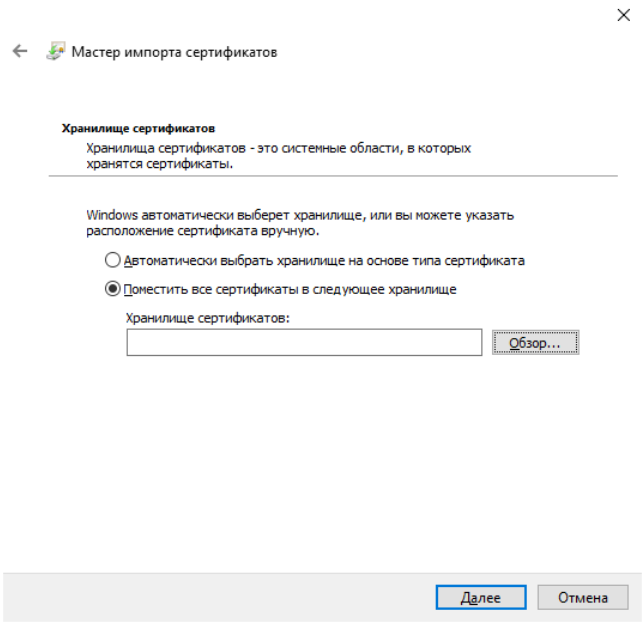


Рисунок 33

3.6. В окне «Выбор хранилища сертификата» нажмите на поле «Доверенные корневые центры сертификации» и нажмите кнопку «ОК» (см. Рисунок 34):

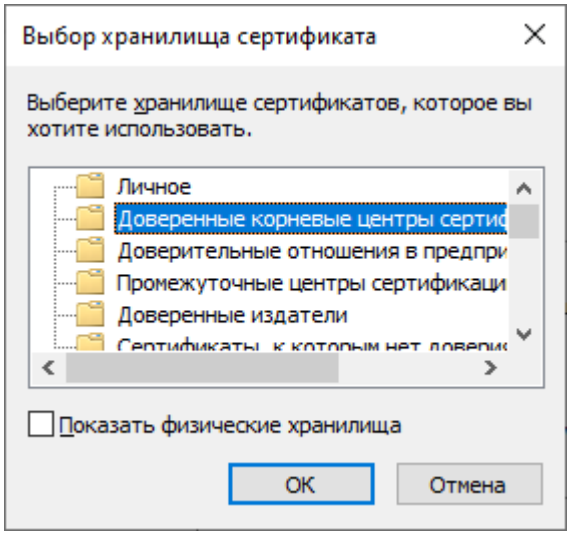


Рисунок 34

3.7. Нажмите кнопку «Далее» (см. Рисунок 35):

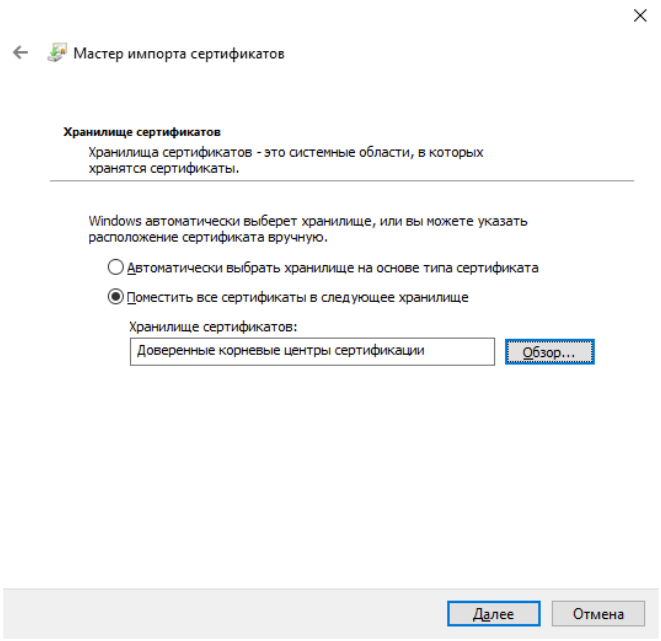


Рисунок 35

3.8. Нажмите кнопку «Готово» для завершения импорта сертификата УЦ (см. Рисунок 36):

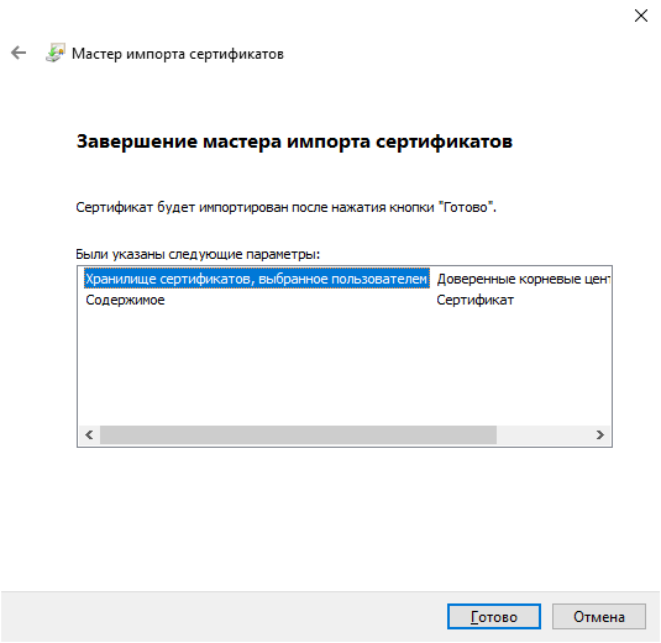


Рисунок 36

3.9. После успешного импорта сертификата появится окно «Мастер импорта сертификатов» с сообщением «Импорт успешно выполнен.» (см. Рисунок 37):

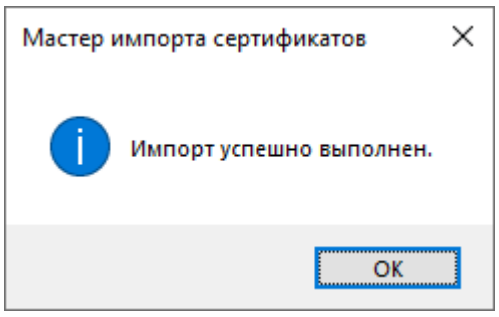


Рисунок 37

Создание VPN-подключения на Windows

1. На панели задач откройте поиск и введите «Параметры». После появления параметров, нажмите кнопку «Открыть» (см. Рисунок 38):

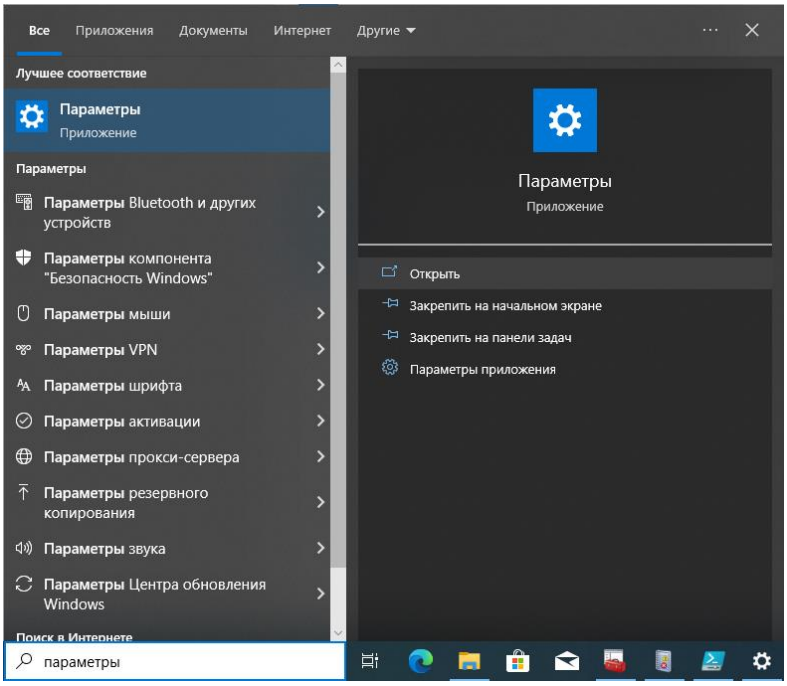


Рисунок 38

2. В панели поиска в параметрах введите «Добавить VPN-подключение» (см. Рисунок 39):

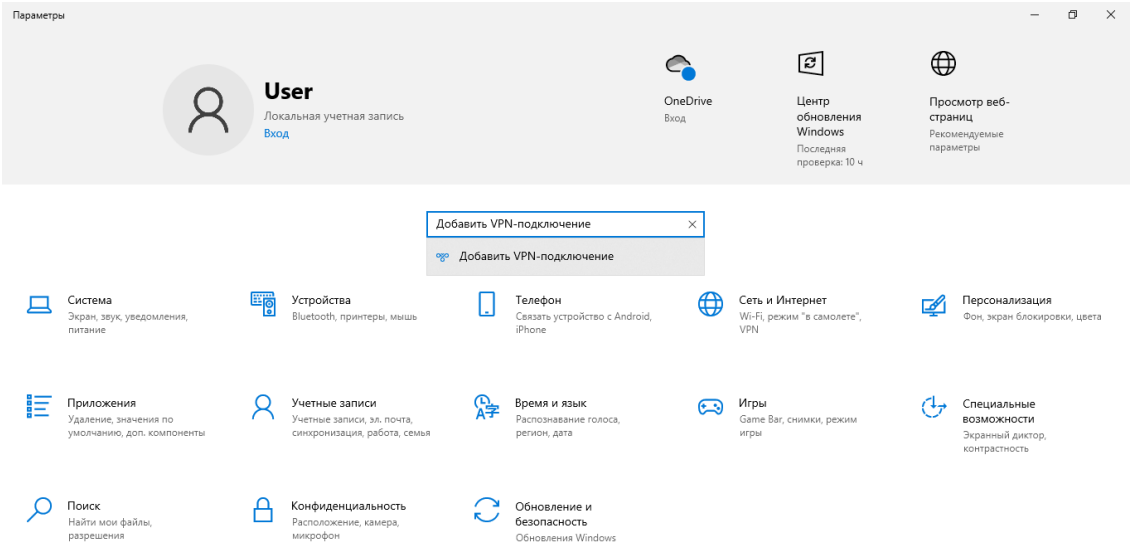


Рисунок 39

3. Нажмите кнопку «Добавить VPN-подключение» (см. Рисунок 40):

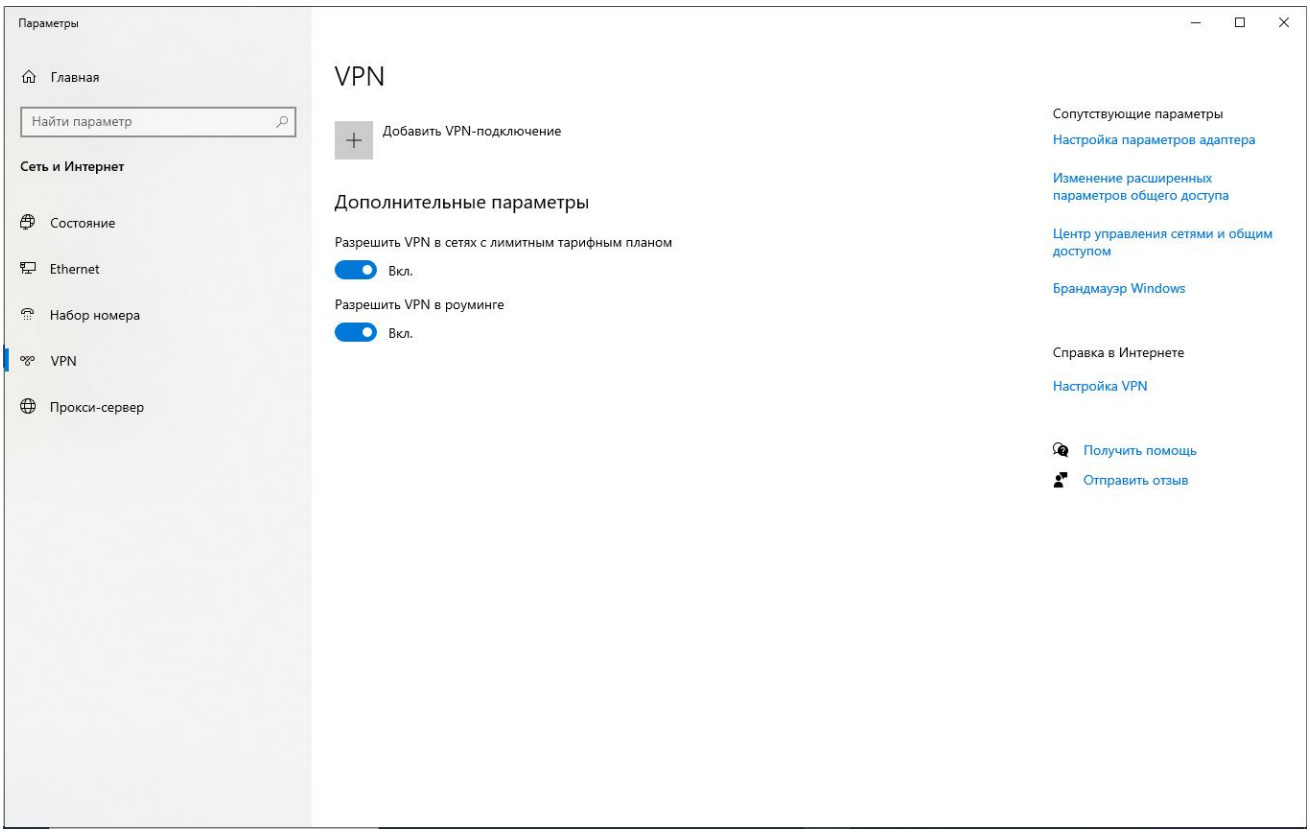


Рисунок 40

4. Введите и выберите следующее:
   4.1. В окне выбора «Поставщик услуг VPN» выберите «Windows (встроенные)».
   4.2. В окне «Имя подключения» введите имя: «IKEv2-IPsec-RSA».
   4.3. В окне «Имя или адрес сервера» введите IP адрес криптошлюза: 10.0.102.25.
   4.4. В окне «Тип VPN» выберите «IKEv2».
   4.5. В окне «Тип данных для входа» выберите «Сертификат».
   4.6. Оставьте по умолчанию галочку «Запомнить мои данные для входа» (см. Рисунок 41).

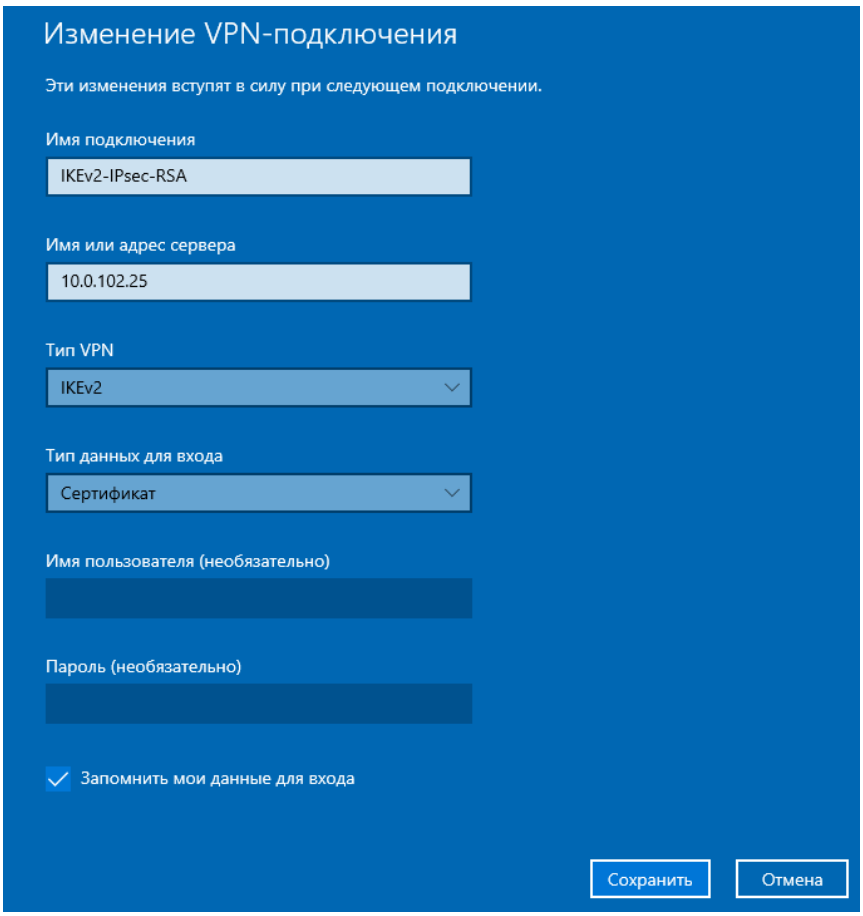


Рисунок 41

5. На панели задач откройте поиск и введите «Windows PowerShell». После появления нажмите кнопку «Открыть» (см. Рисунок 42):

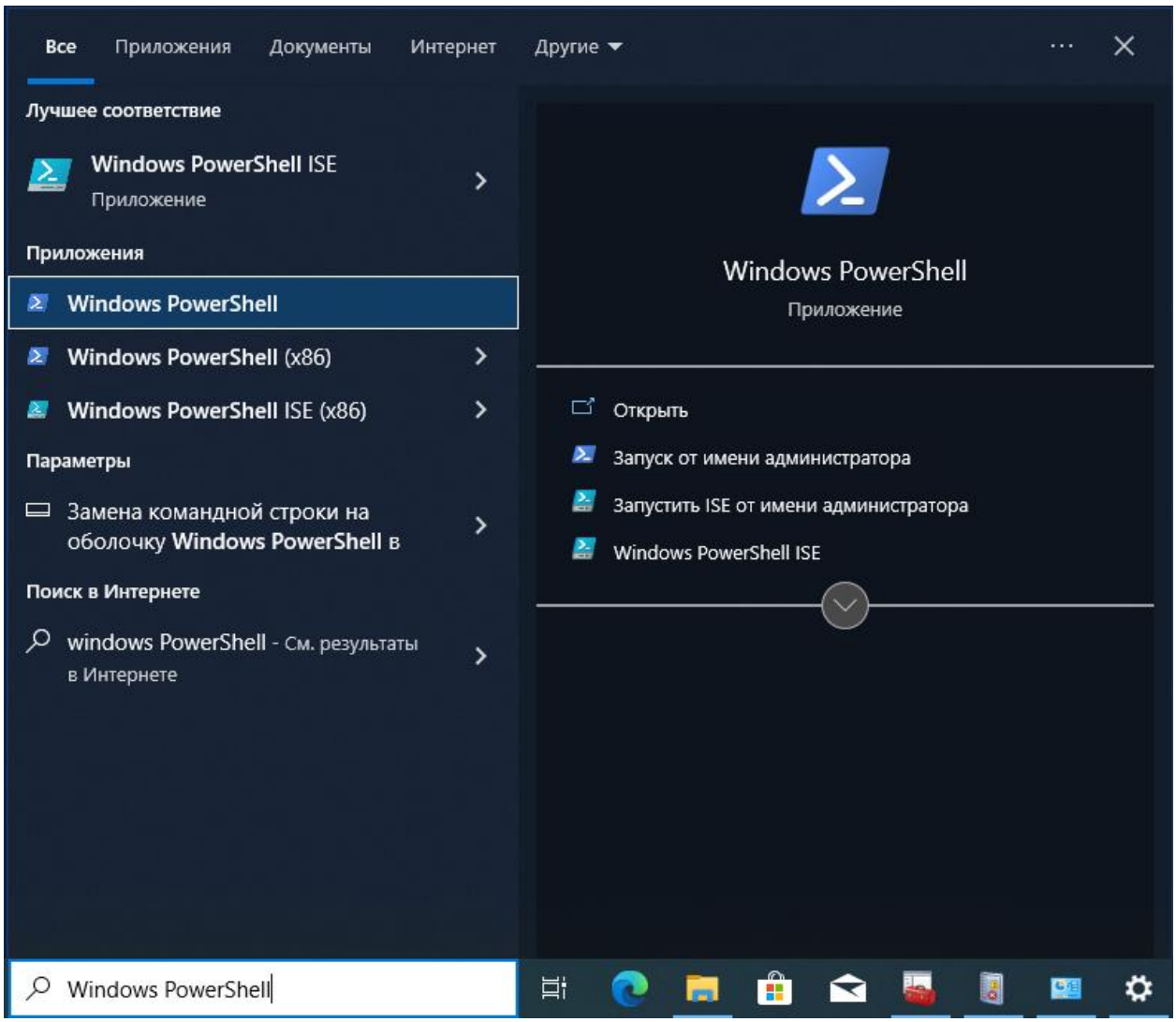


Рисунок 42

6. В консоль «Windows PowerShell» необходимо ввести следующую команду и подтвердить изменения криптографии:

При создании VPN-подключения были применены максимально упрощенные настройки, для изменения настроек необходимо прописать команду в «Windows PowerShell». Данная команда изменяет параметры IPsec и IKEv2 для существующего VPN-подключения. В данной команде определяются параметры протокола IKEv2: алгоритм шифрования, проверки целостности пакета, выработки общего ключа, а также псевдослучайная функция. Также определяются параметры протокола IPsec: алгоритм проверки целостности пакета, выработки общего ключа, а также комбинированный алгоритм шифрования.

```
PS C:\Windows\system32> Set-VpnConnectionIPsecConfiguration -ConnectionName IKEv2-IPsec-RSA -AuthenticationTransformConstants SHA256128 -CipherTransformConstants AES256 -DHGroup Group14 -EncryptionMethod AES256 -IntegrityCheckMethod SHA256 -PfsGroup PFS2048
```

```
Подтверждение
Changing the Cryptography Settings. Do you want to continue?
[Y] Да - Y  [N] Нет - N  [S] Приостановить - S  [?] Справка (значением по умолчанию является "Y"): Y
```

Если необходимо настроить split tunneling, чтобы только определенный трафик был направлен через защищенное соединение IPsec, нужно будет указать это явно. Например, для данного стенда команда будет иметь следующий вид:

```
PS C:\Windows\system32> Add-VpnConnectionRoute -DestinationPrefix 192.168.0.0/16
IKEv2-IPsec-RSA
```

7. Завершите настройку VPN-профиля, для этого на панели задач найдите поиск и введите «Панель управления». После появления параметров нажмите кнопку «Открыть» (см. Рисунок 43):

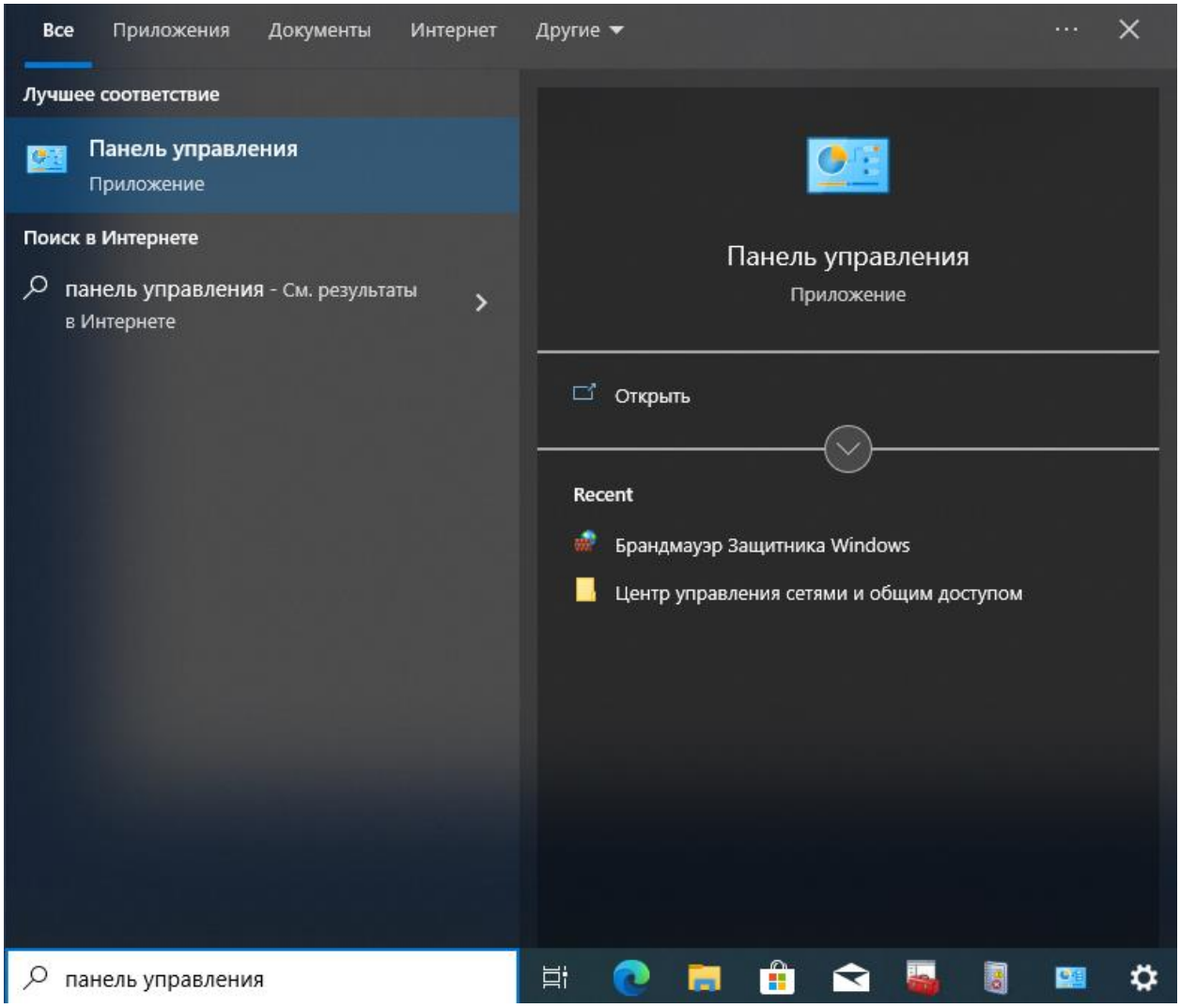


Рисунок 43

8. В разделе «Сеть и интернет» нажмите «Просмотр состояния сети и задач» (см. Рисунок 44):

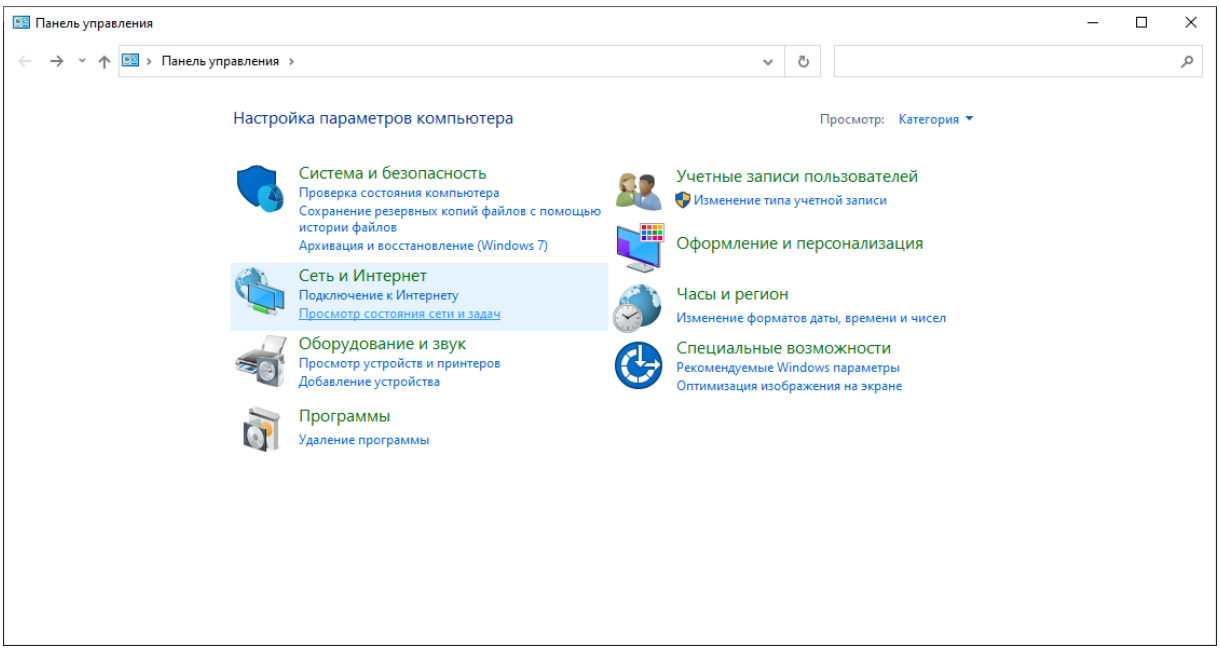


Рисунок 44

9. В окне «Центр управления сетями и общим доступом» в меню слева выберите «Изменение параметров адаптера», нажмите левой кнопкой мыши дважды (см. Рисунок 45):

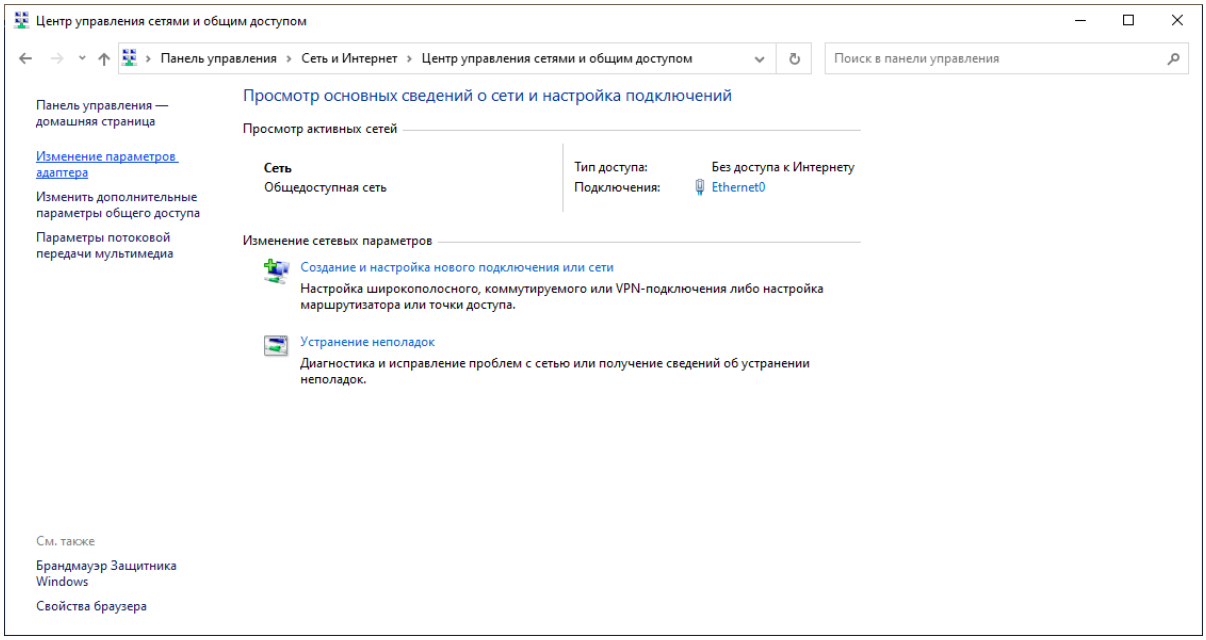


Рисунок 45

10. Выберите VPN-подключение «IKEv2-IPsec-RSA» (см. Рисунок 46) и нажмите правой кнопкой мыши:

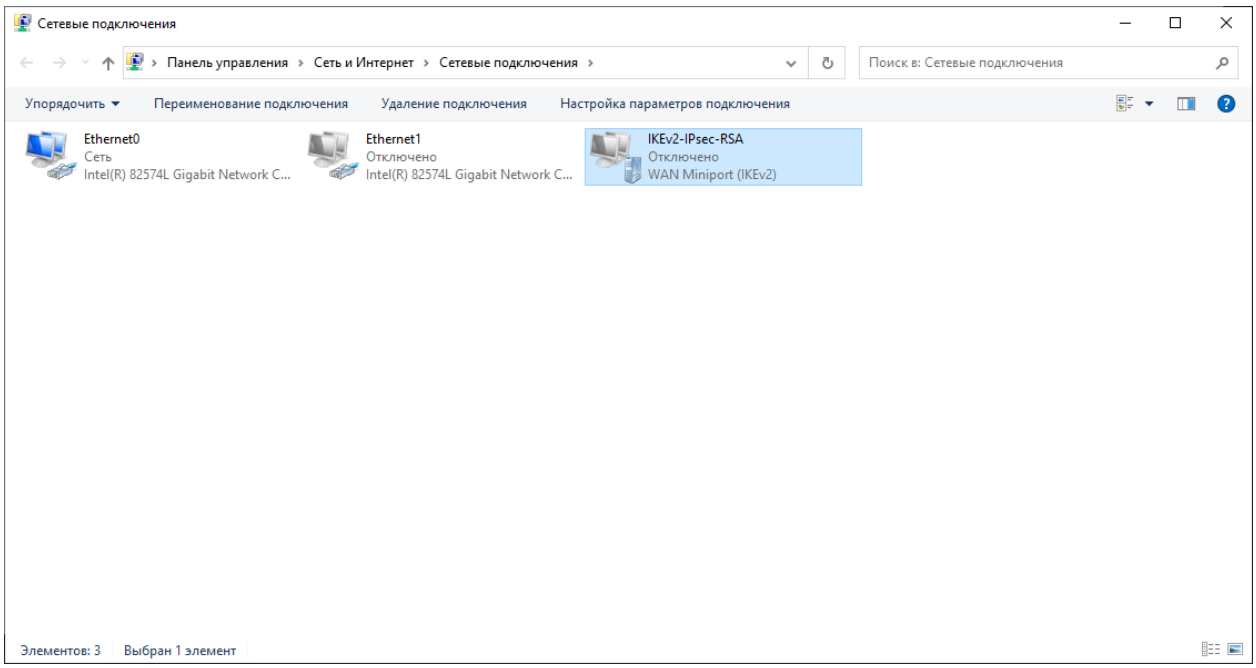


Рисунок 46

11. Перейдите в «Свойства» для настройки VPN-подключения (см. Рисунок 47):

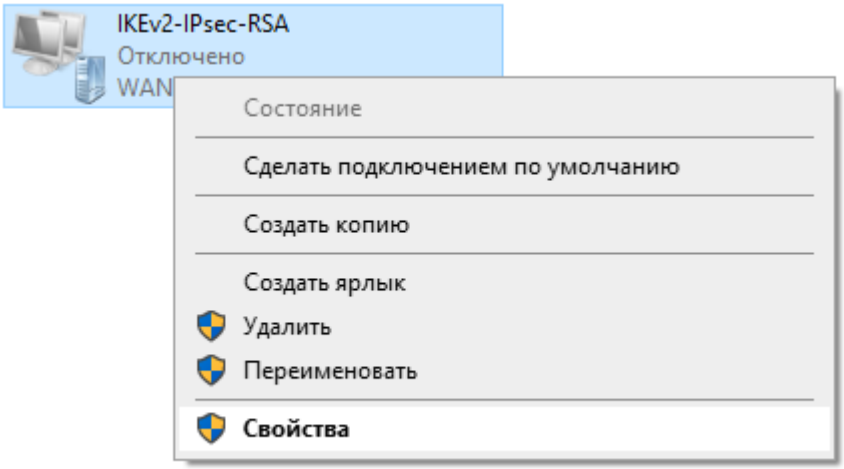


Рисунок 47

12. Перейдите во вкладку «Безопасность» в окне «IKEv2-IPsec-RSA: свойства», поставьте флажок «Использовать сертификаты компьютеров» и нажмите на кнопку «Дополнительные параметры» (см. Рисунок 48):

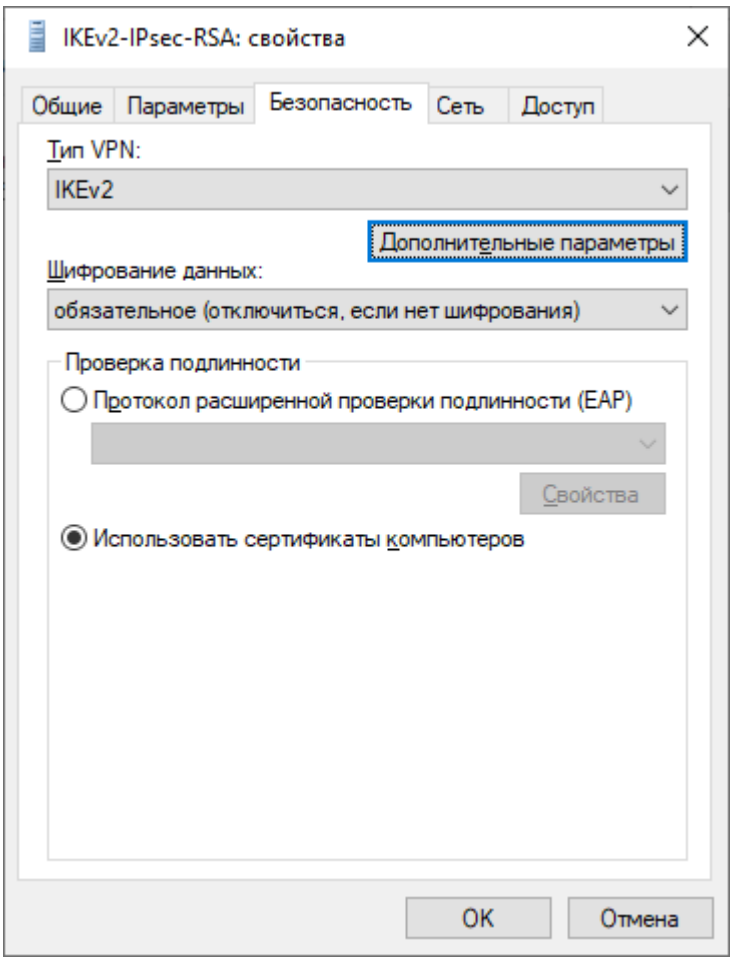


Рисунок 48

13. В окне «Дополнительные свойства» уберите выбор «Мобильность» и нажмите «ОК» (см. Рисунок 49):

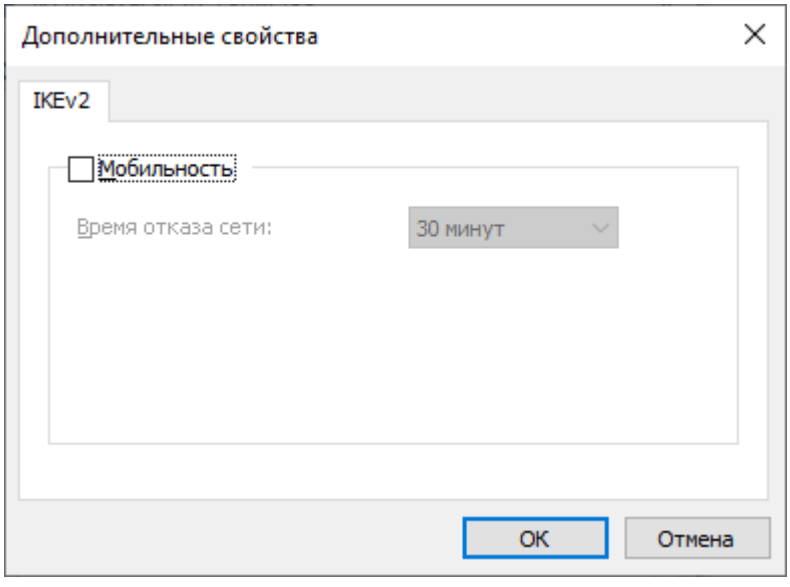


Рисунок 49

14. Перейдите во вкладку «Сеть» выберите галочку с полем «IP версии 4 (TCP/IPv4)» и нажмите на кнопку «Свойства» (см. Рисунок 50):

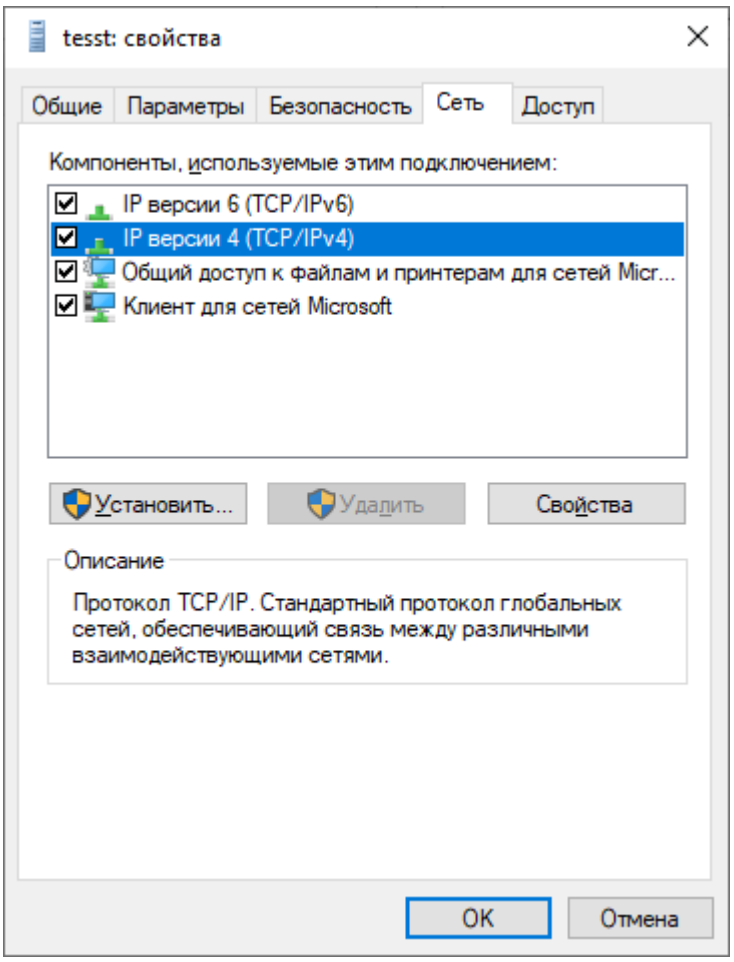


Рисунок 50

15. В окне «Свойства: IP версии 4 (TCP/IPv4)» нажмите кнопку «Дополнительно…» (см. Рисунок 51):

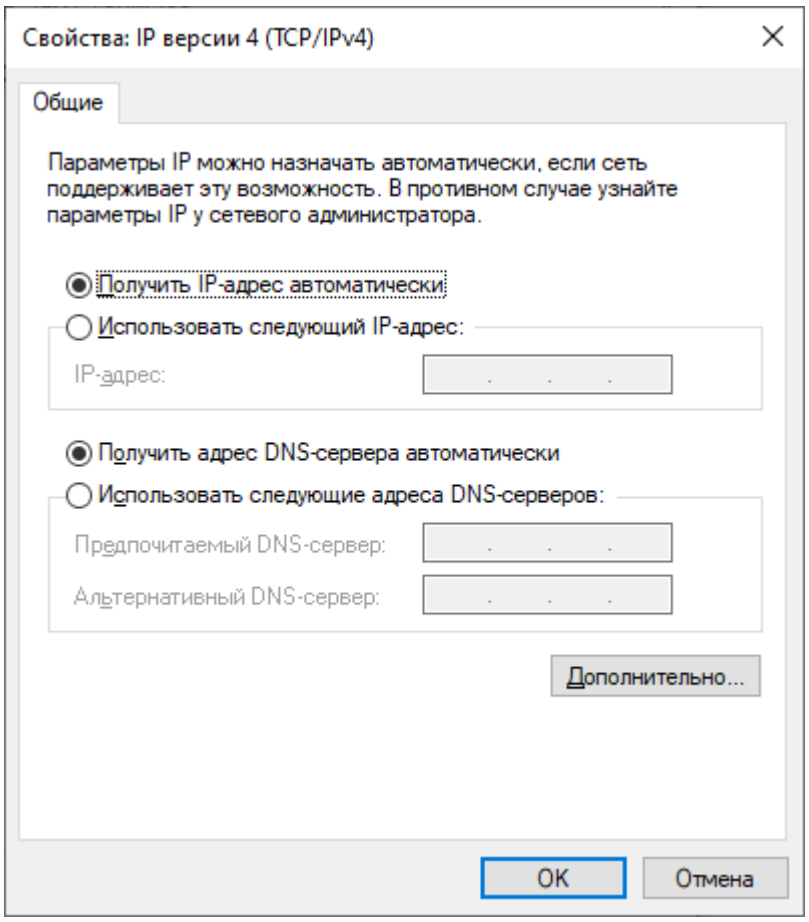


Рисунок 51

16. В окне «Дополнительные параметры TCP/IP» уберите флаг «Использовать основной шлюз в удаленной сети» и нажмите кнопку «ОК». Данная настройка split tunneling поможет использовать локальную сеть для выхода в интернет, не направляя весь трафик в защищенное соединение IPsec с криптошлюзом (см. Рисунок 52).

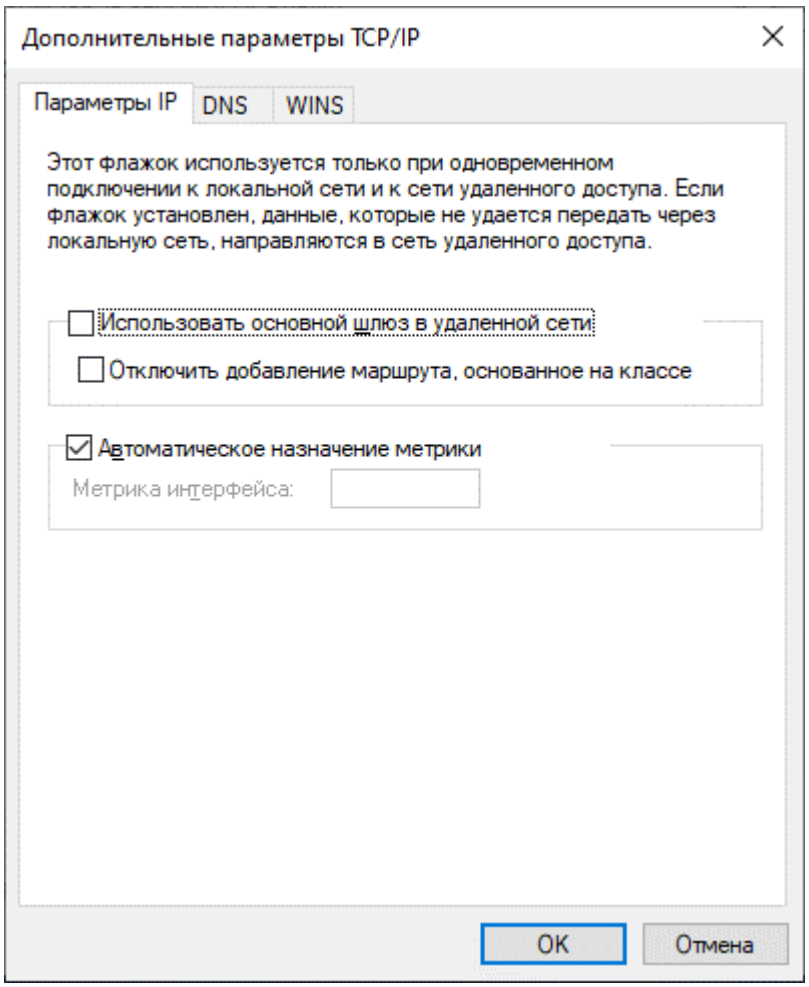


Рисунок 52

17. После нажмите «ОК» в окне «Свойства: IP версии 4 (TCP/IPv4)» и в окне «IKEv2-IPsec-RSA: свойства» для завершения настроек.

Настройка iOS

Настройки PKI (запросы и сертификаты)

Создание и экспорт сертификата для мобильного клиента с криптошлюза

Настройка PKI (создание и экспорт сертификата для мобильного клиента) на iOS происходит аналогично настройке в разделе «Создание и экспорт сертификата для мобильного клиента с криптошлюза».

Импорт сертификатов на iOS

Для импорта сертификатов на iOS выполните описанные ниже шаги:

1. Перенесите сертификаты на iOS.
2. Для импорта сертификата мобильного клиента на iOS с закрытым ключем выполните следующие пункты:

   2.1. Перейдите в «Файлы», куда были перенесены сертификаты. Выберите сертификат в формате pkcs12 (см. Рисунок 53).



Рисунок 53

   2.2. Загрузите профиль скаченного сертификата в формате pkcs12 на iOS (см. Рисунок 54 и Рисунок 55):

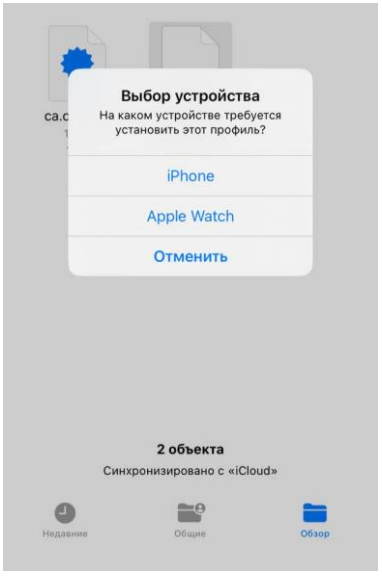


Рисунок 54

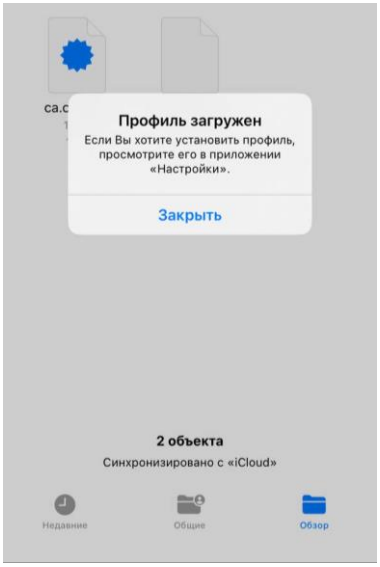


Рисунок 55

2.3. Установите профили для VPN. Для этого зайдите в настройки и перейдите во вкладку «Профиль загружен» (см. Рисунок 56):

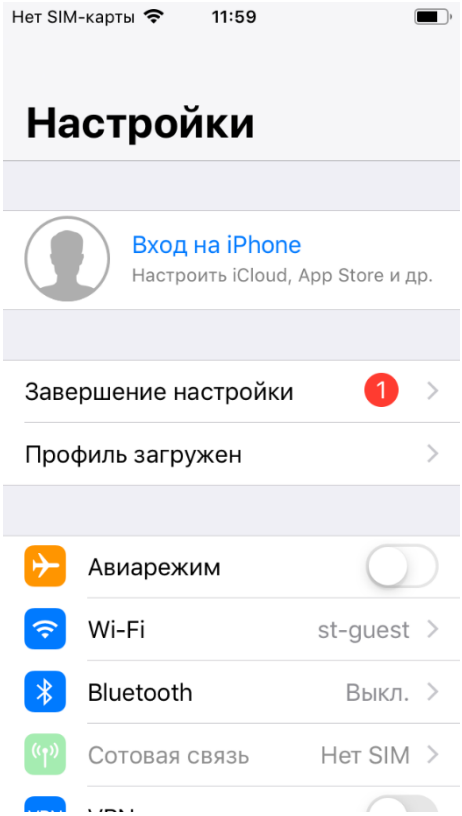


Рисунок 56

2.4. В следующем окне нажмите кнопку «Установить» (см. Рисунок 57):

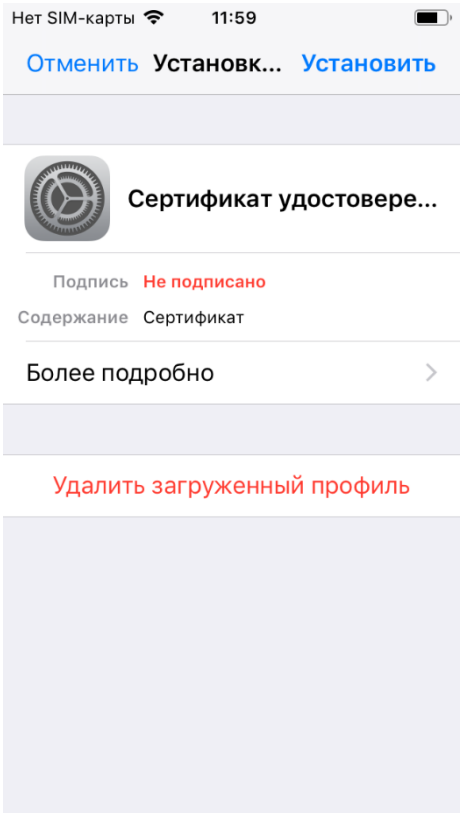


Рисунок 57

2.5. Введите пароль от устройства, чтобы дальше подтвердить установку.

2.6. Подтвердите установку, нажав на кнопку «Установить» (см. Рисунок 58 и Рисунок 59):

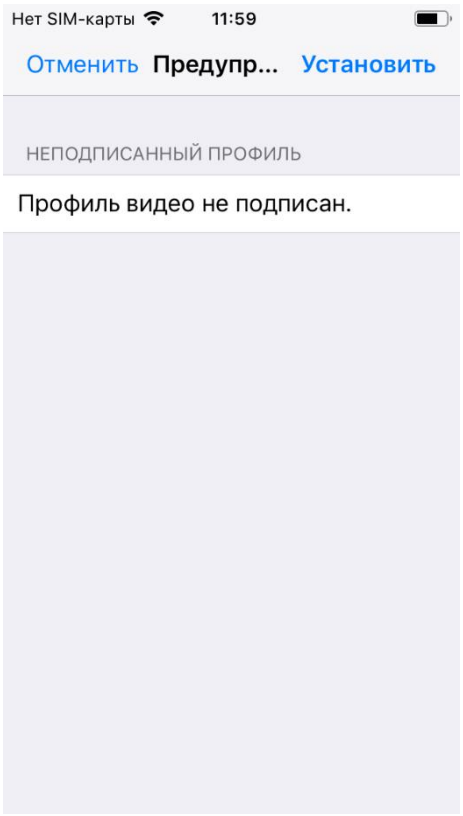


Рисунок 58

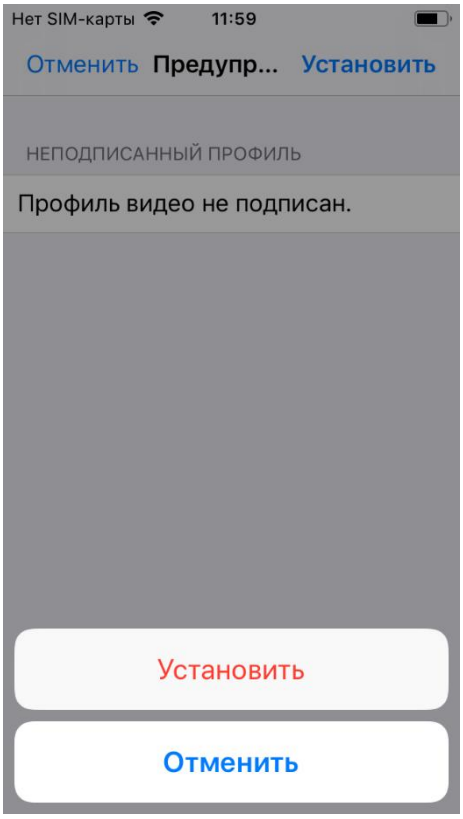


Рисунок 59

2.7. Введите пароль от сертификата в формате pkcs12 и нажмите «Далее» (см. Рисунок 60):

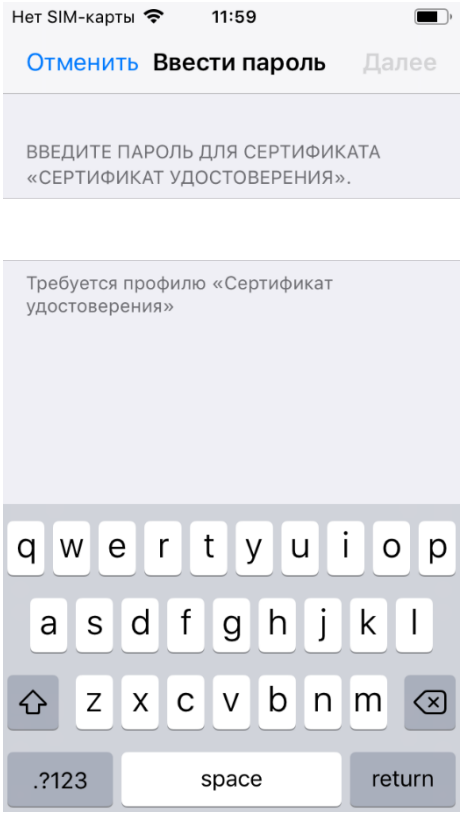


Рисунок 60

2.8. После установки загруженного профиля сертификат мобильного клиента iOS будет доступен, но будет в статусе «Не проверен», для этого необходимо импортировать сертификат УЦ. Для этого повторите п.п. 2.1-2.6.

Создание VPN-подключения на iOS

1. Далее необходимо добавить конфигурацию VPN.
2. Для этого необходимо зайти в «Настройки» > «Основные» > «VPN».
3. Прейдите в настройки «VPN» и нажмите на кнопку «Добавить конфигурацию VPN» и выберите «IKEv2…» (см. Рисунок 61):

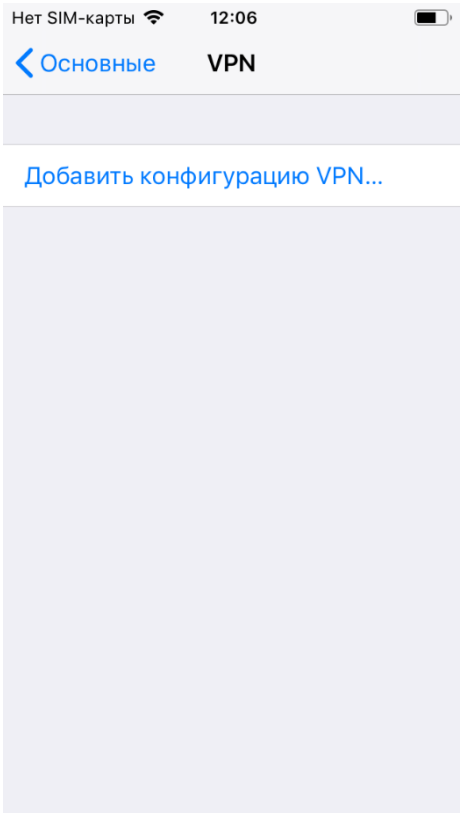


Рисунок 61

3. Выполните следующие действия:
   3.1. Выберите тип «IKEv2».
   3.2. В поле «Описание» запишите название «IKEv2».
   3.3. В поле «Сервер» введите IP адрес криптошлюза: 10.0.102.25.
   3.4. В поле «Удаленный ID» введите повторно IP адрес криптошлюза.
   3.5. Переместите ползунок поля «Сертификат» в режим «ON» (см. Рисунок 62).

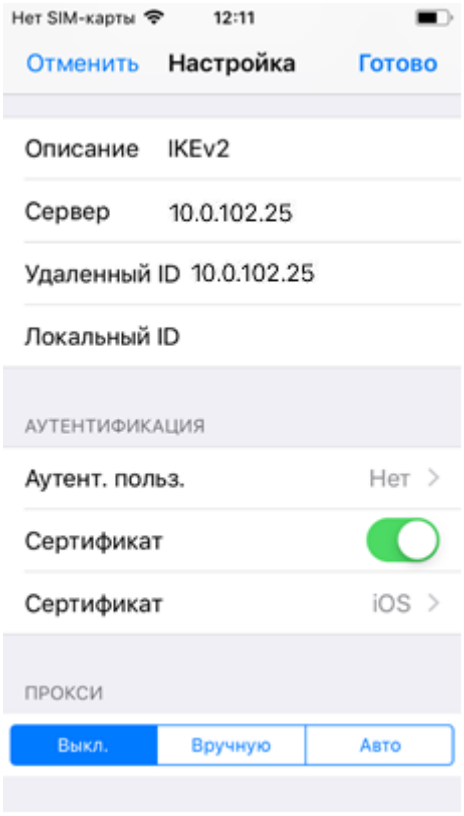


Рисунок 62

3.6. В поле «Сертификат» выберите импортированный сертификат и нажмите кнопку «Назад» (см. Рисунок 63).

3.7. Отключите аутентификацию пользователя для этого в поле «Аутентификации пользователя» выберите тип «Нет».

Примечание – В этом случае будет строиться защищенное соединение IPsec с IKEv2 на сертификатах RSA, если оставить предыдущий выбор, то будет строиться защищенное соединение IPsec с аутентификацией в IKEv2 на сертификатах RSA с EAP (см. Рисунок 62).

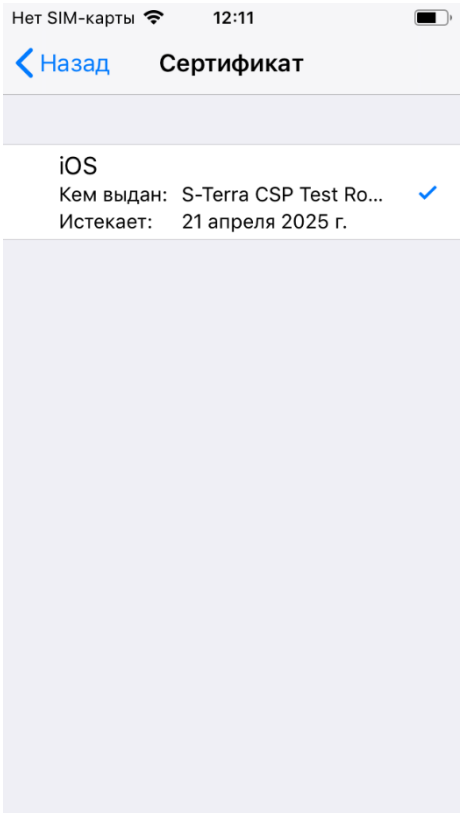


Рисунок 63

4. Сохраните конфигурацию, для этого нажмите кнопку «Готово» (см. Рисунок 64):

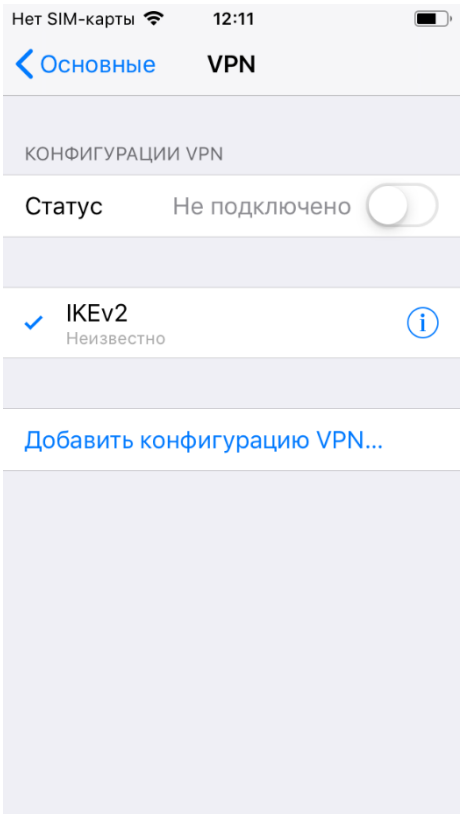


Рисунок 64

Настройка macOS

Настройки PKI (запросы и сертификаты)

Создание и экспорт сертификата для мобильного клиента с криптошлюза

Настройка PKI (создание и экспорт сертификата для мобильного клиента) на macOS происходит аналогично настройке в разделе «Создание и экспорт сертификата для мобильного клиента с криптошлюза».

Импорт сертификатов на macOS

Для импорта сертификатов на Android выполните описанные ниже шаги:

1. Перенесите сертификаты на macOS.
2. Для импорта сертификата мобильного клиента на macOS с закрытым ключем выполните следующие пункты:

2.1. Перейдите в каталог, куда были перенесены сертификаты (см. Рисунок 65).

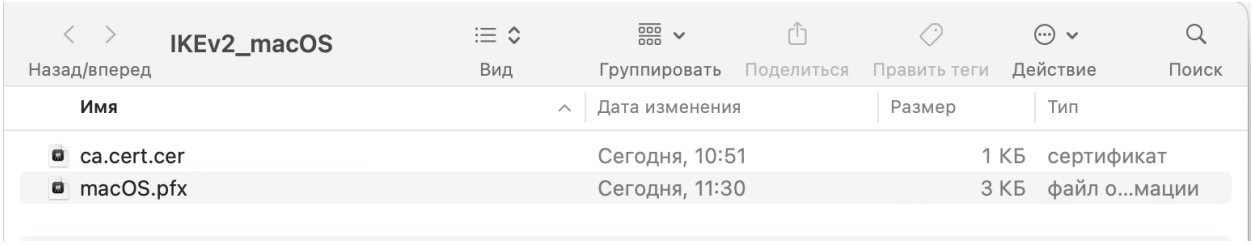


Рисунок 65

2.2. Импортируйте сертификат УЦ на macOS (см. Рисунок 66):

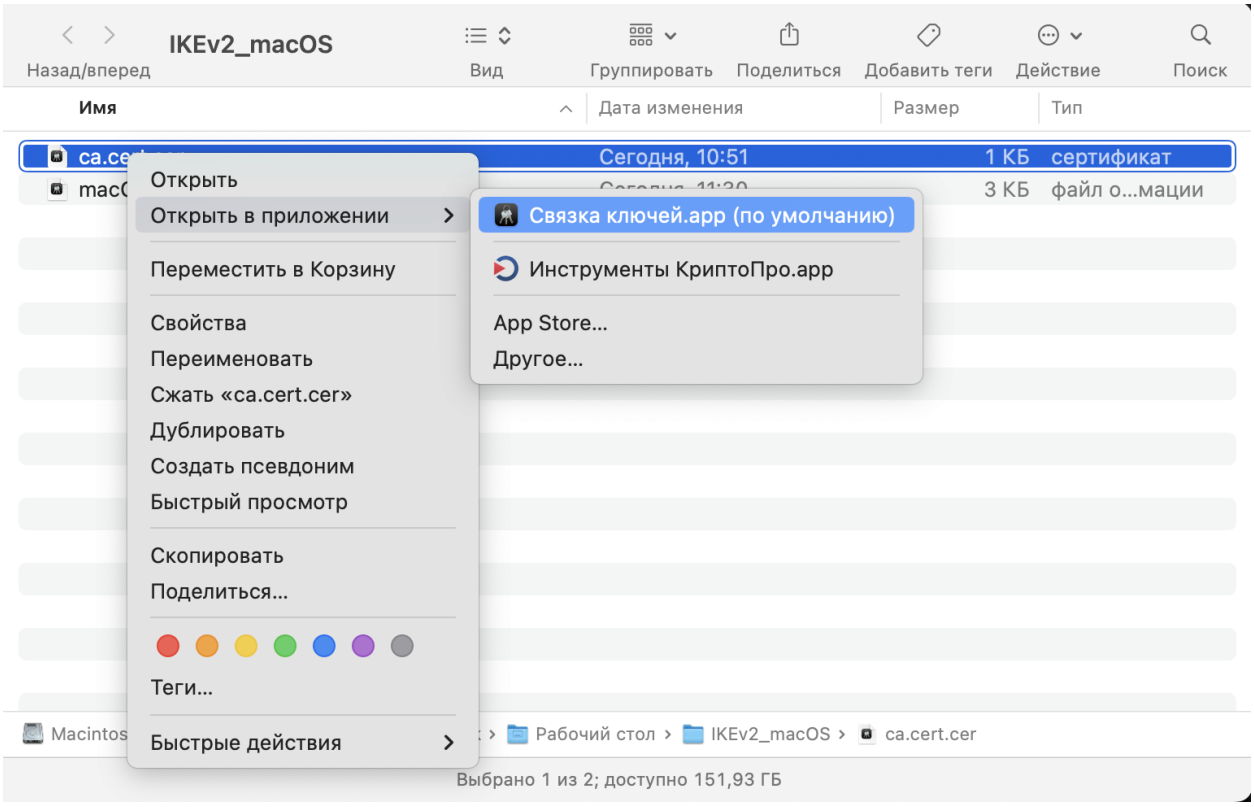


Рисунок 66

2.3. Зайдите в «Связку ключей» и найдите в поиске сертификат УЦ (см. Рисунок 67):

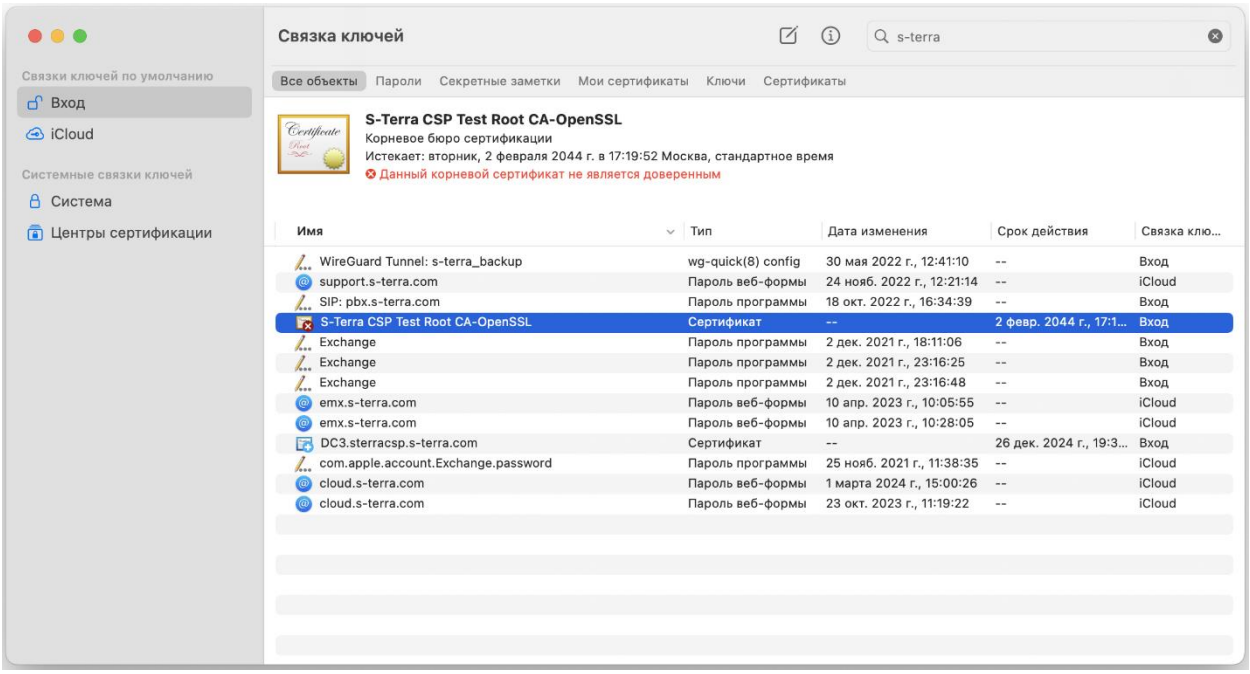


Рисунок 67

2.4. Зайдите в «Свойства» сертификата (см. Рисунок 68).

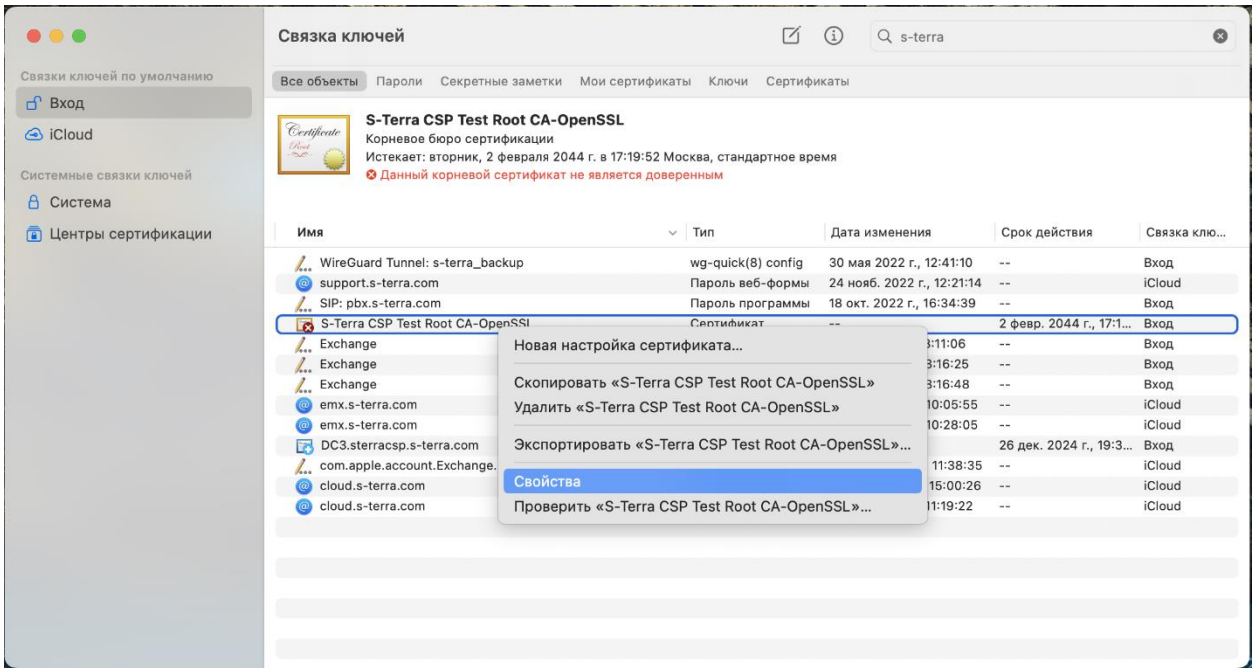


Рисунок 68

2.5. В свойствах в разделе «Доверие» измените параметр «Параметры использования сертификата» на «Всегда доверять» и разрешите изменения в настройках. Это необходимо, чтобы операционная система доверяла всем сертификатам, которые подписаны этим УЦ (см. Рисунок 69 и Рисунок 70):

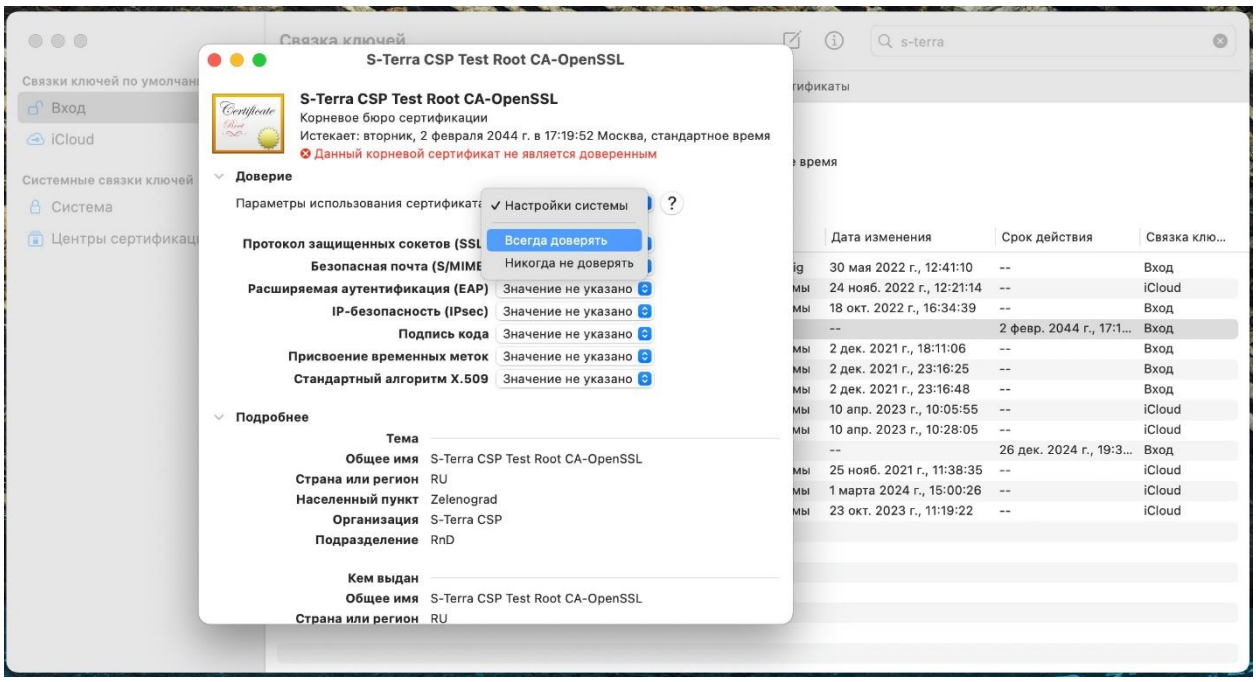


Рисунок 69

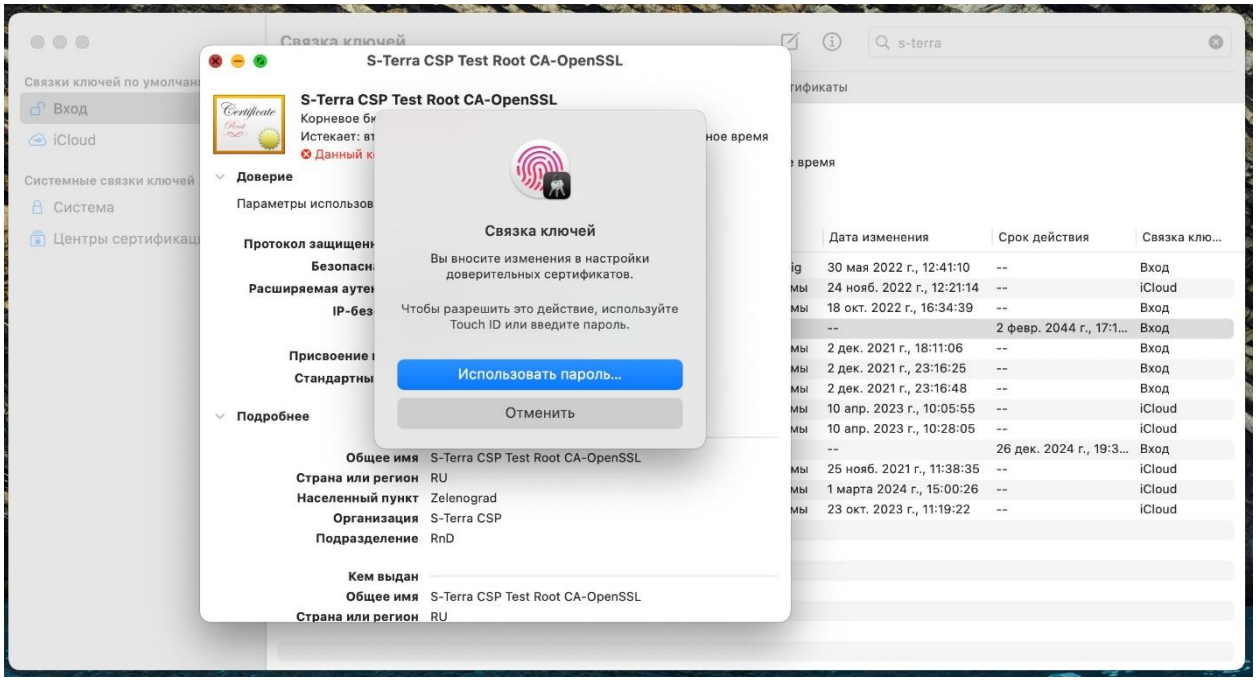


Рисунок 70

2.6. Аналогично импортируйте сертификат для macOS и введите пароль от контейнера (см. Рисунок 71 и Рисунок 72):

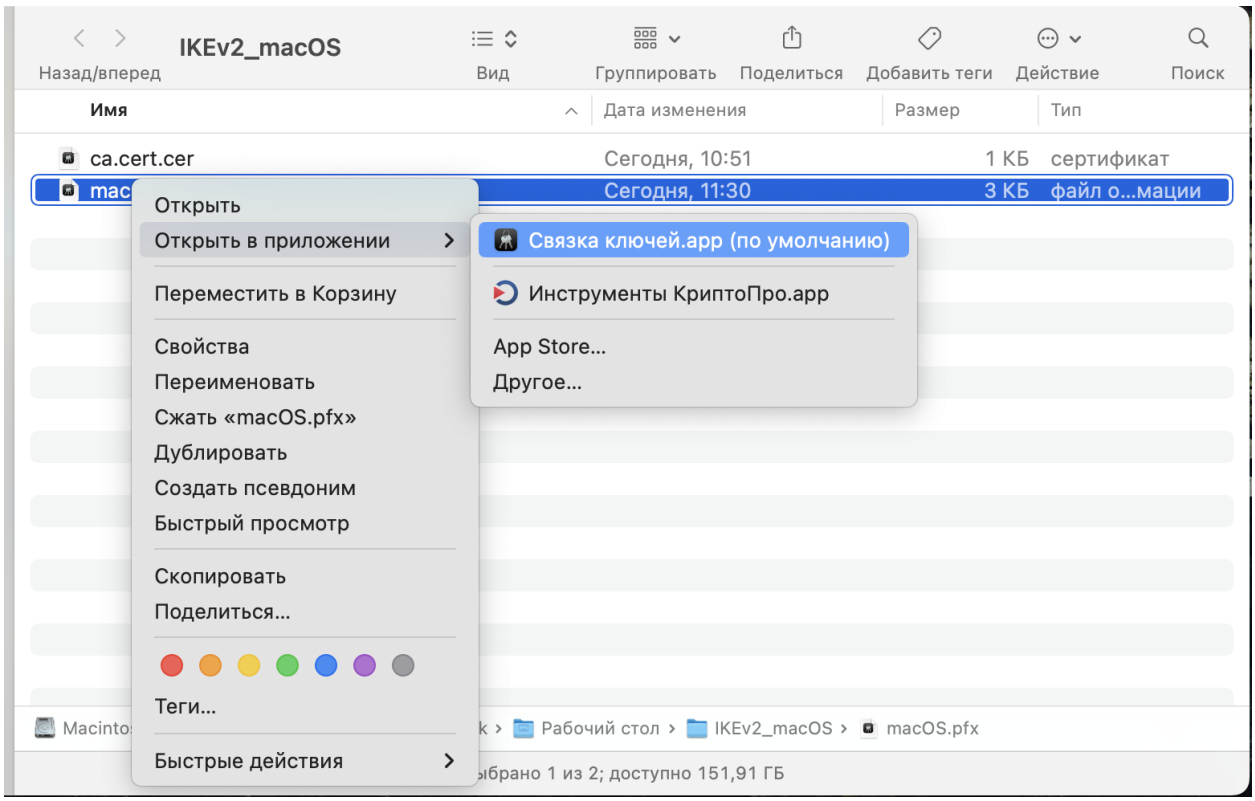


Рисунок 71

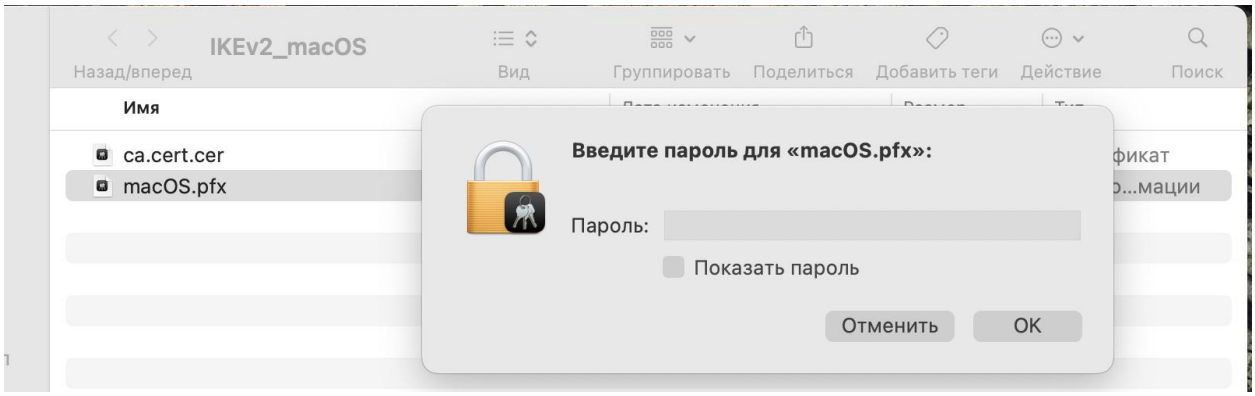


Рисунок 72

2.7. Зайдите в свойства данного сертификата в связках ключей и убедитесь, что сертификат действителен (см. Рисунок 73):

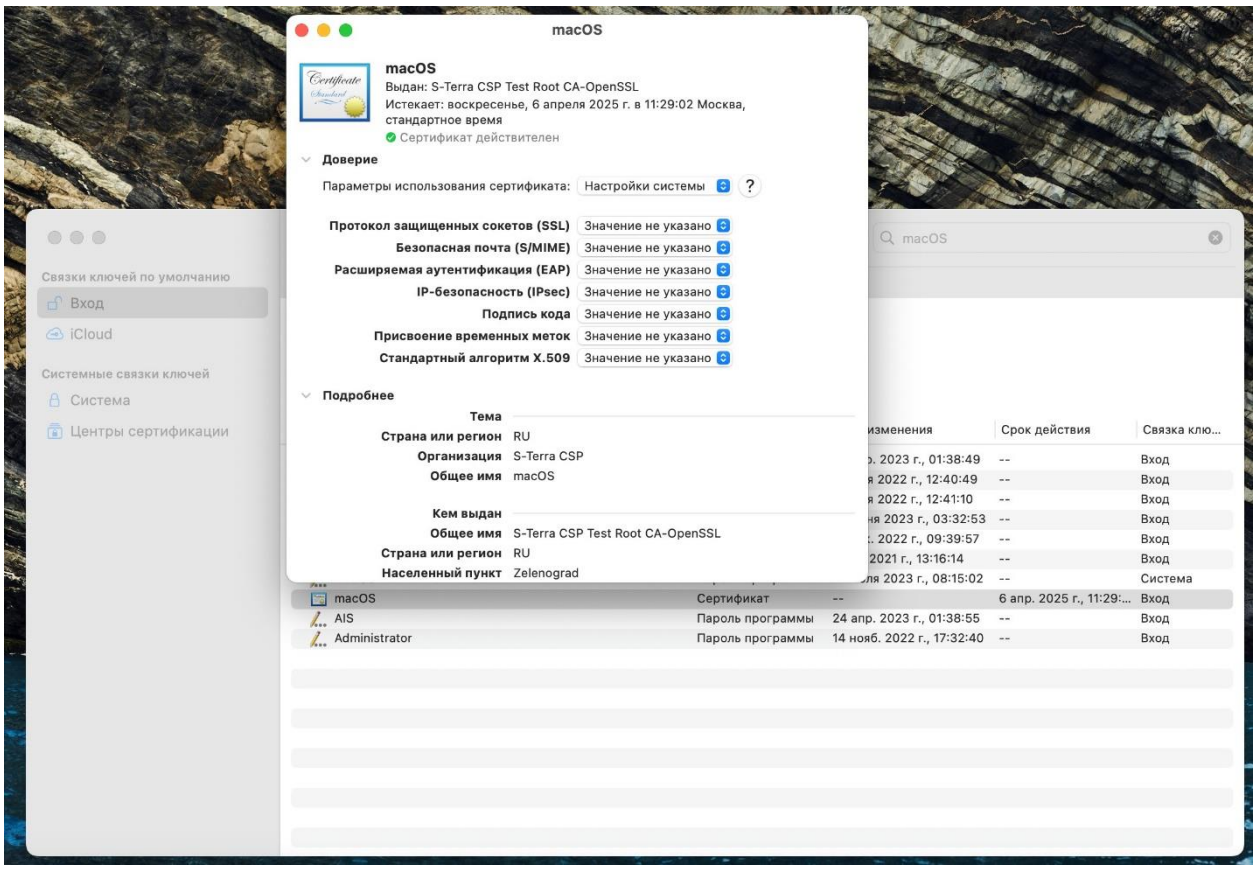


Рисунок 73

Создание VPN-подключения на macOS

Далее необходимо добавить конфигурацию VPN.

1. Для этого необходимо зайти в «Системные настройки» (см. Рисунок 74):

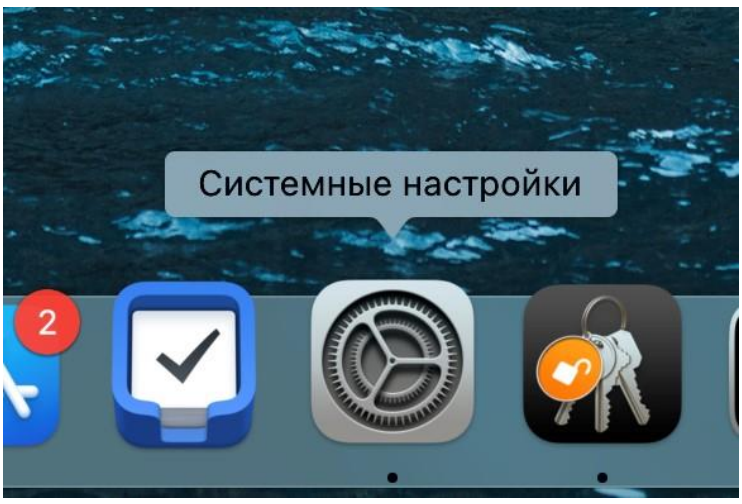


Рисунок 74

2. Перейдите в настройки «VPN» и нажмите на кнопку «Добавить конфигурацию VPN» и выберите в раскрывающемся списке «IKEv2…» (см. Рисунок 75):

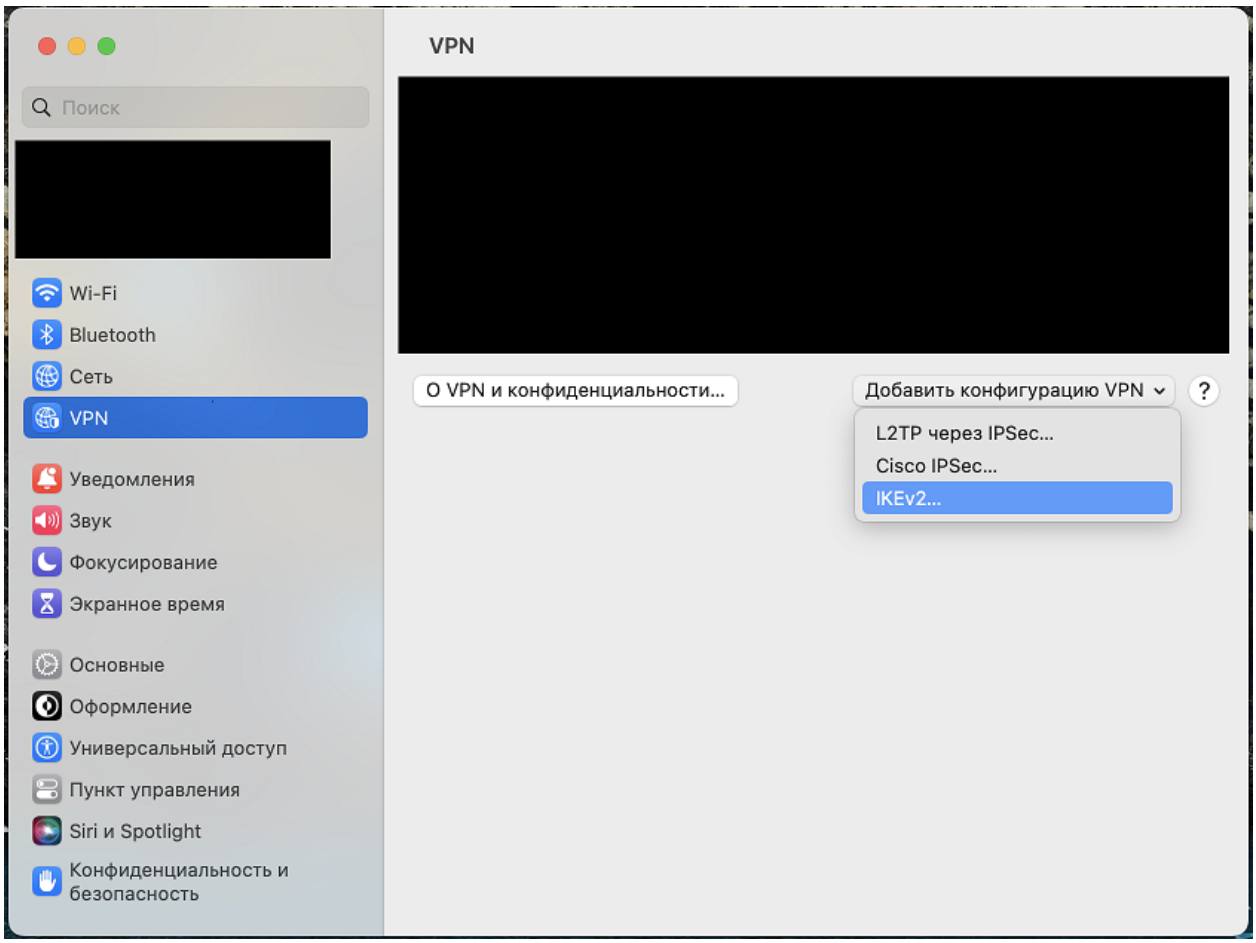


Рисунок 75

3. В окне «IKEv2» выполните следующие действия:
   3.1. В поле «Отображаемое имя» запишите имя «VPN»;
   3.2. Введите IP адрес криптошлюза в поле «Адрес сервера»: 10.0.102.25;
   3.3. Введите повторно IP адрес криптошлюза в поле «Удаленный ID»;
   3.4. Выберите вариант аутентификации пользователя для этого в поле «Аутентификации пользователя» выберите тип «Сертификат» (см. Рисунок 76):

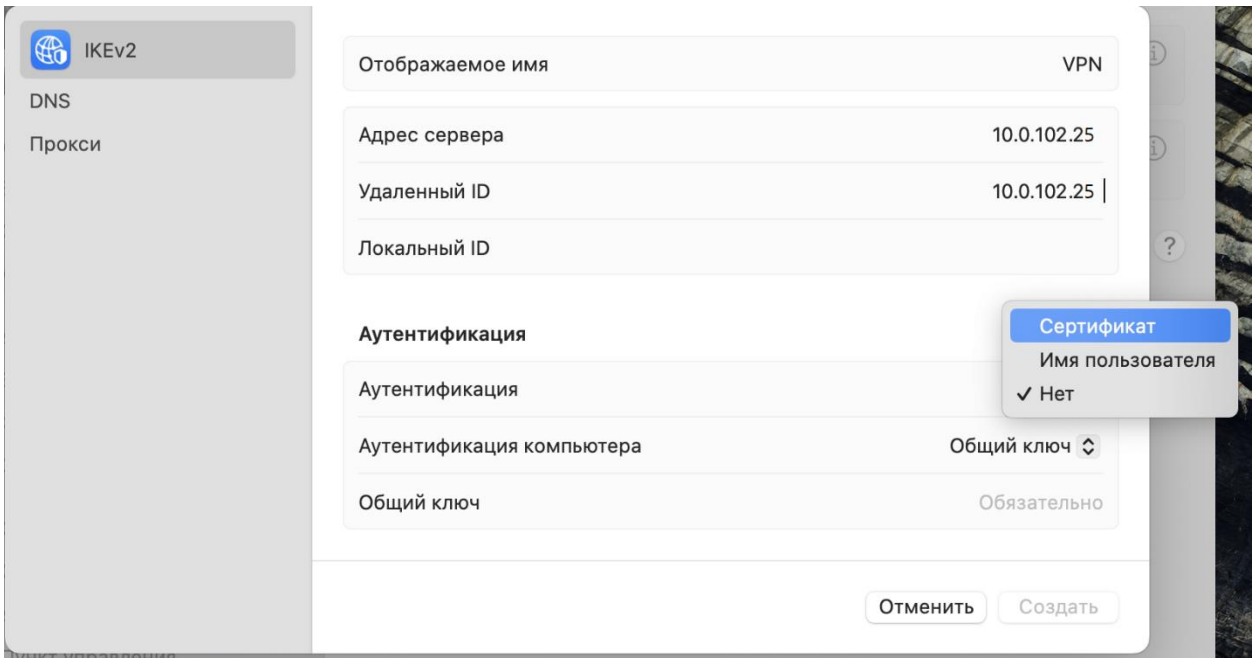


Рисунок 76

3.5. Выберите сертификат macOS, который импортировали выше (см. Рисунок 77):

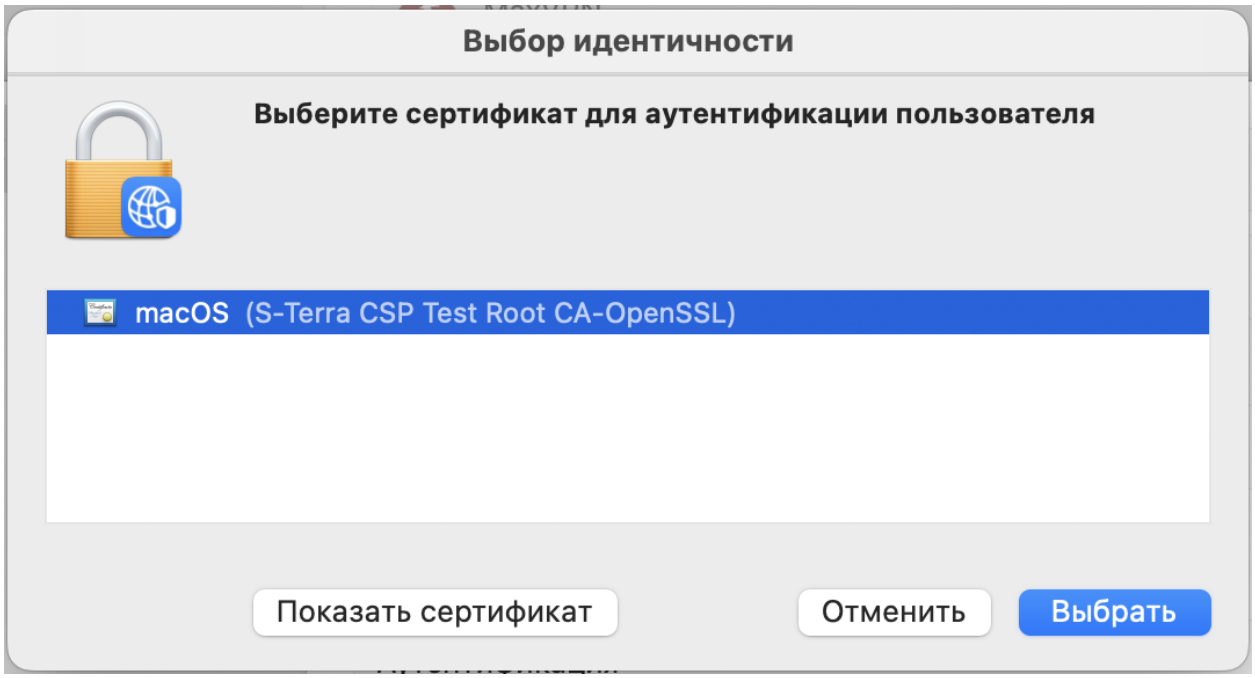


Рисунок 77

3.6. После выбора сертификата выберите в поле «Аутентификации» выберите «Нет» (см. Рисунок 78).

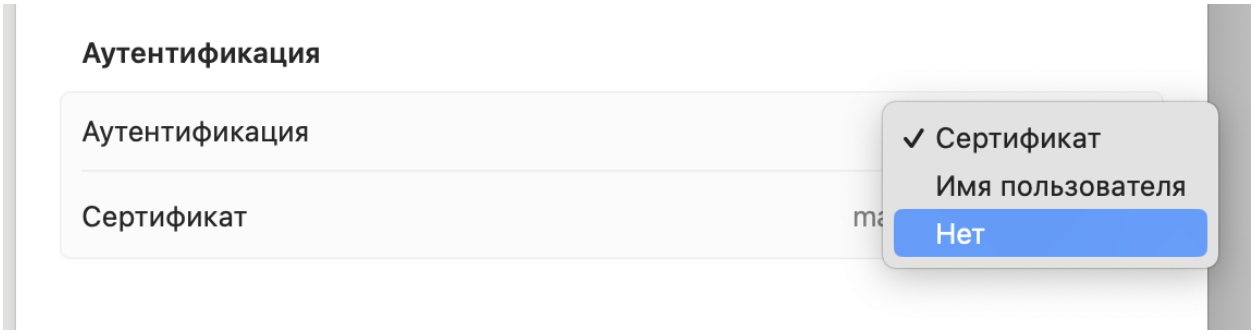


Рисунок 78

Примечание – В этом случае будет строиться защищенное соединение IPsec с IKEv2 на сертификатах RSA, если оставить предыдущий выбор, то будет строиться защищенное соединение IPsec с аутентификацией в IKEv2 на сертификатах RSA с EAP.

3.7. Сохраните созданную конфигурацию, для этого нажмите кнопку «Создать» (см. Рисунок 79).

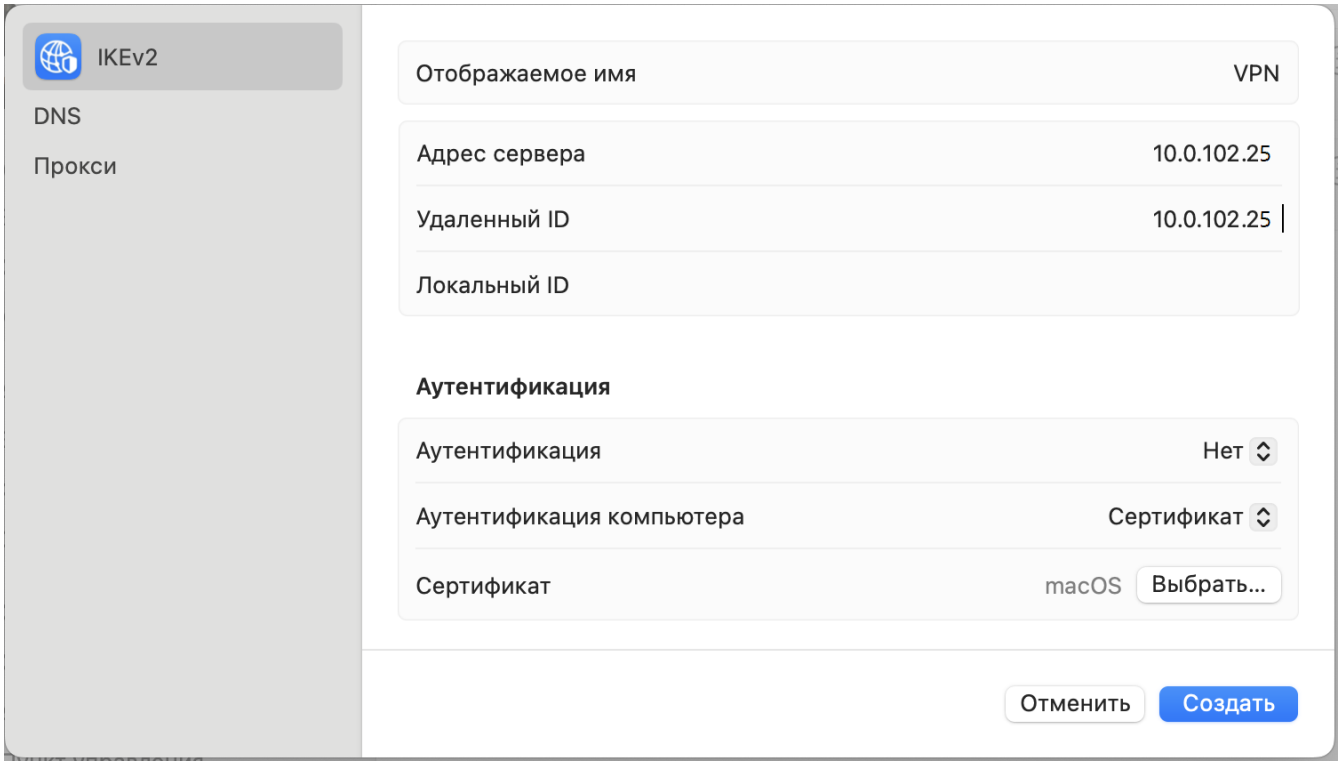


Рисунок 79

Настройка AnyConnect

Настройки PKI (запросы и сертификаты)

Создание и экспорт сертификата для мобильного клиента с криптошлюза

Действия по созданию запроса, выпуску и экспорту сертификата в формате pkcs12 для мобильного клиента происходят на криптошлюзе. Сертификат будет подписан с использованием УЦ на базе OpenSSL, настроенного ранее на криптошлюзе в разделе «Настройки PKI (запросы и сертификаты)».

1. Запрос для выпуска сертификата для AnyConnect будет сгенерирован при помощи УЦ на базе OpenSSL на криптошлюзе. Запустите процесс генерации закрытого ключа, введите пароль:

```
root@Hub1:~/ca#openssl genrsa -aes256 -out private/anyconnect.key.pem 2048
```

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

2. Создайте запрос на сертификат AnyConnect с сохранением файла запроса:

```
root@Hub1:~/ca#openssl req -config  openssl.cnf -key  private/anyconnect.key.pem -new
-sha256 -out csr/anyconnect.csr.pem
```

При создании сертификата для мобильного клиента на криптошлюзе `Subject` сертификата должен совпадать с `Subject` указаным в структуре `IdentityEntry` под именем `AnyConnect` в LSP в разделе «Настройка шифрования с помощью LSP».

```
Enter pass phrase for private/anyconnect.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name []:AnyConnect
Country Name (2 letter code) []:RU
State or Province Name []:
Locality Name []:Zelenograd
Organization Name []:S-Terra CSP
Organizational Unit Name []:
Email Address []:
```

3. Выпустите сертификат для AnyConnect по сгенерированному запросу на сертификат на УЦ OpenSSL:

```
root@Hub1:~/ca#openssl ca -config openssl.cnf -extensions usr_cert -days 375 -notext -
md sha256 -in csr/anyconnect.csr.pem -out certs/anyconnect.cert.pem
```

```
Using configuration from openssl.cnf
Enter pass phrase for /root/ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Jul 19 08:47:52 2024 GMT
            Not After : Jul 29 08:47:52 2025 GMT
        Subject:
            countryName               = RU
            localityName              = Zelenograd
            organizationName          = S-Terra CSP
            commonName                = AnyConnect
        X509v3 extensions:
            X509v3 Basic Constraints:
```

```
                CA:FALSE
            X509v3 Subject Key Identifier:
                96:CC:34:C3:43:C4:BF:41:69:70:EC:1E:29:93:50:7E:75:5B:A4:76
            X509v3 Authority Key Identifier:
                keyid:3C:6E:22:57:B8:50:A5:95:A5:0E:CC:95:78:02:04:49:7C:5C:28:CD
                DirName:/CN=S-Terra CSP Test Root CA-OpenSSL/C=RU/L=Zelenograd/O=S-
Terra CSP/OU=RnD
                serial:77:0C:FD:38:EF:6F:E6:9B:B1:69:CE:48:FE:D6:F4:50:FF:01:36:A6
            X509v3 Key Usage:
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
Certificate is to be certified until Jul 29 08:47:52 2025 GMT (375 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
```

4. Создайте сертификат в формате pkcs12 для AnyConnect и введите пароль от ключа и пароль для экспорта сертификата:

```
root@Hub1:~/ca# openssl pkcs12 -export -out certs/anyconnect.pfx -inkey
private/anyconnect.key.pem -in certs/anyconnect.cert.pem -legacy
```

```
Enter pass phrase for private/anyconnect.key.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

Импорт сертификатов на AnyConnect

Импорт сертификатов на AnyConnect происходит аналогично настройке в разделе «Импорт сертификатов на Windows».

Установка AnyConnect на Windows

1. Распакуйте архив с ПО Cisco AnyConnect и запустите «Setup.exe» (см. Рисунок 80):

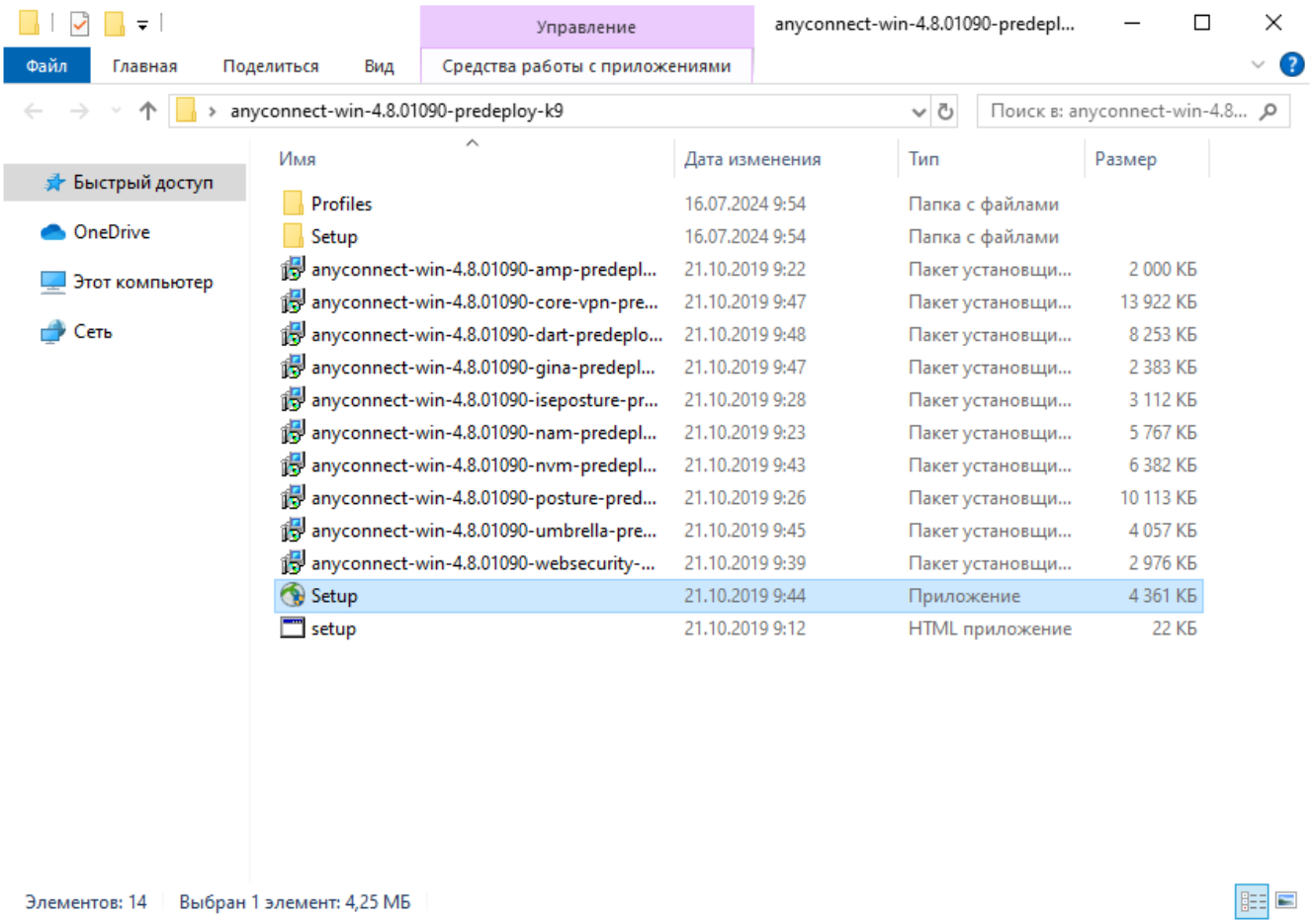


Рисунок 80

2. В открывшемся окне «Cisco AnyConnect Secure Mobility Client Install Selector» выберите следующие элементы и нажмите кнопку «Install Selected» (см. Рисунок 81):

   - Core & VPN;
   - Start Before Login;
   - Diagnostic And Reporting Tool.

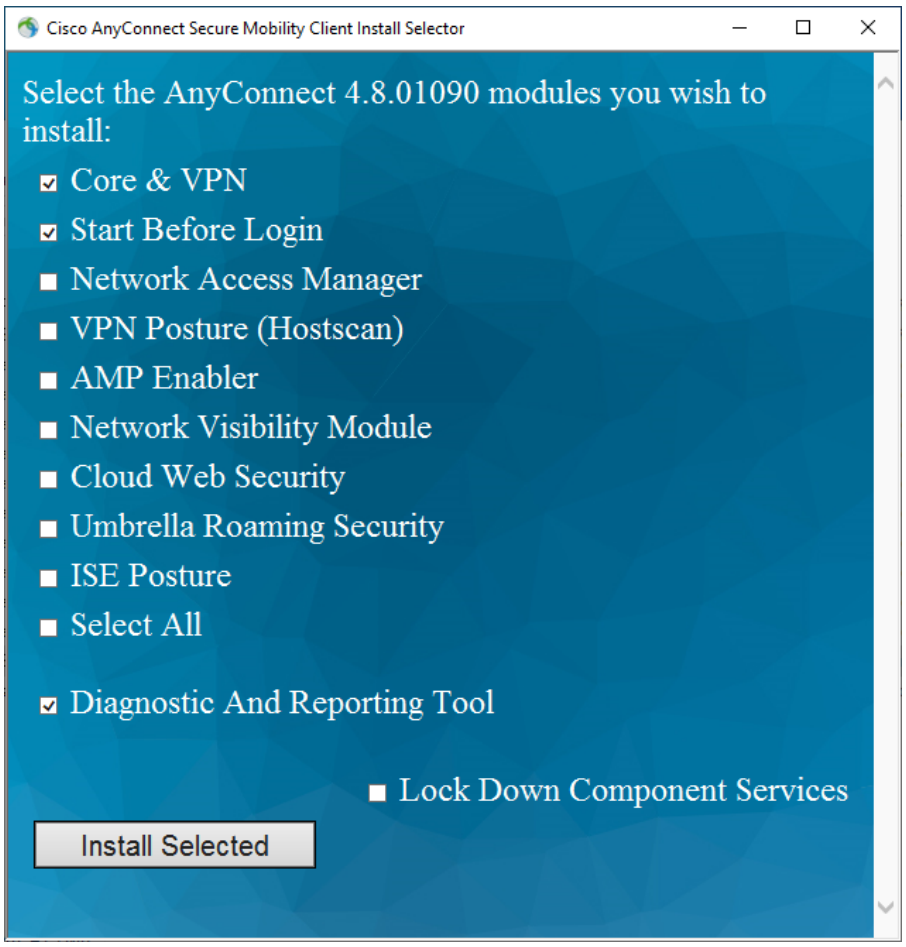


Рисунок 81

3. В открывшемся окне «Cisco AnyConnect Secure Mobility Client Install Selector» кнопку «OK» (см. Рисунок 82).

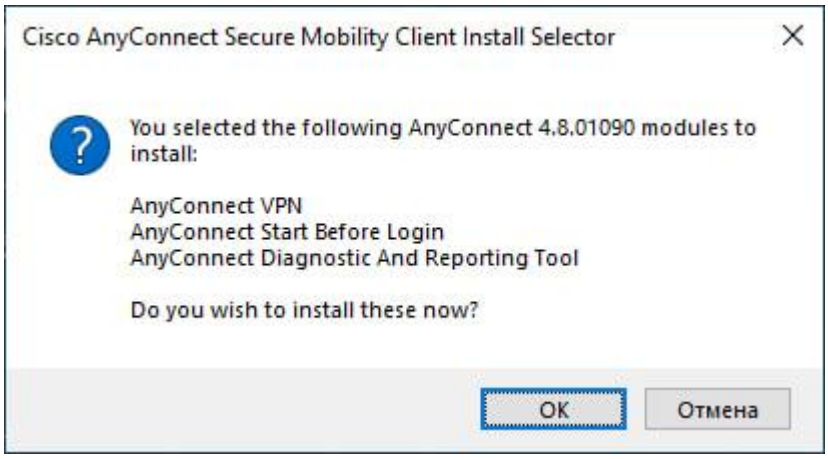


Рисунок 82

4. В открывшемся окне «Cisco AnyConnect Secure Mobility Client EULA» примите лицензионное соглашение. Для этого нажмите кнопку «Accept» (см. Рисунок 83):

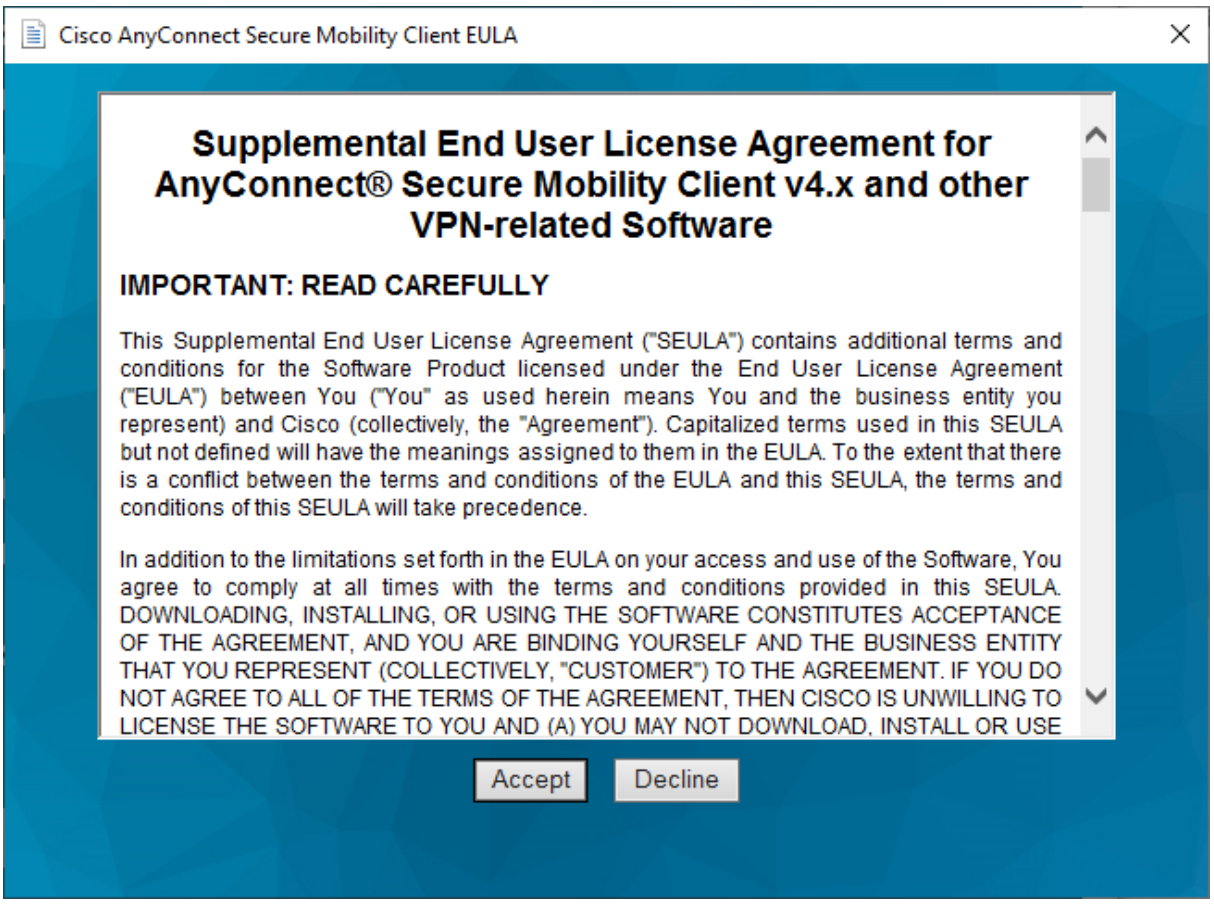


Рисунок 83

5. Завершите установку. Для этого нажмите кнопку «ОК» (см. Рисунок 84):

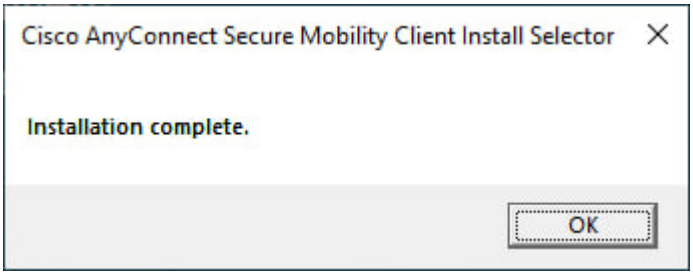


Рисунок 84

6. Проверьте, что ПО «Cisco AnyConnect Secure Mobility Client» успешно установилось (см. Рисунок 85):

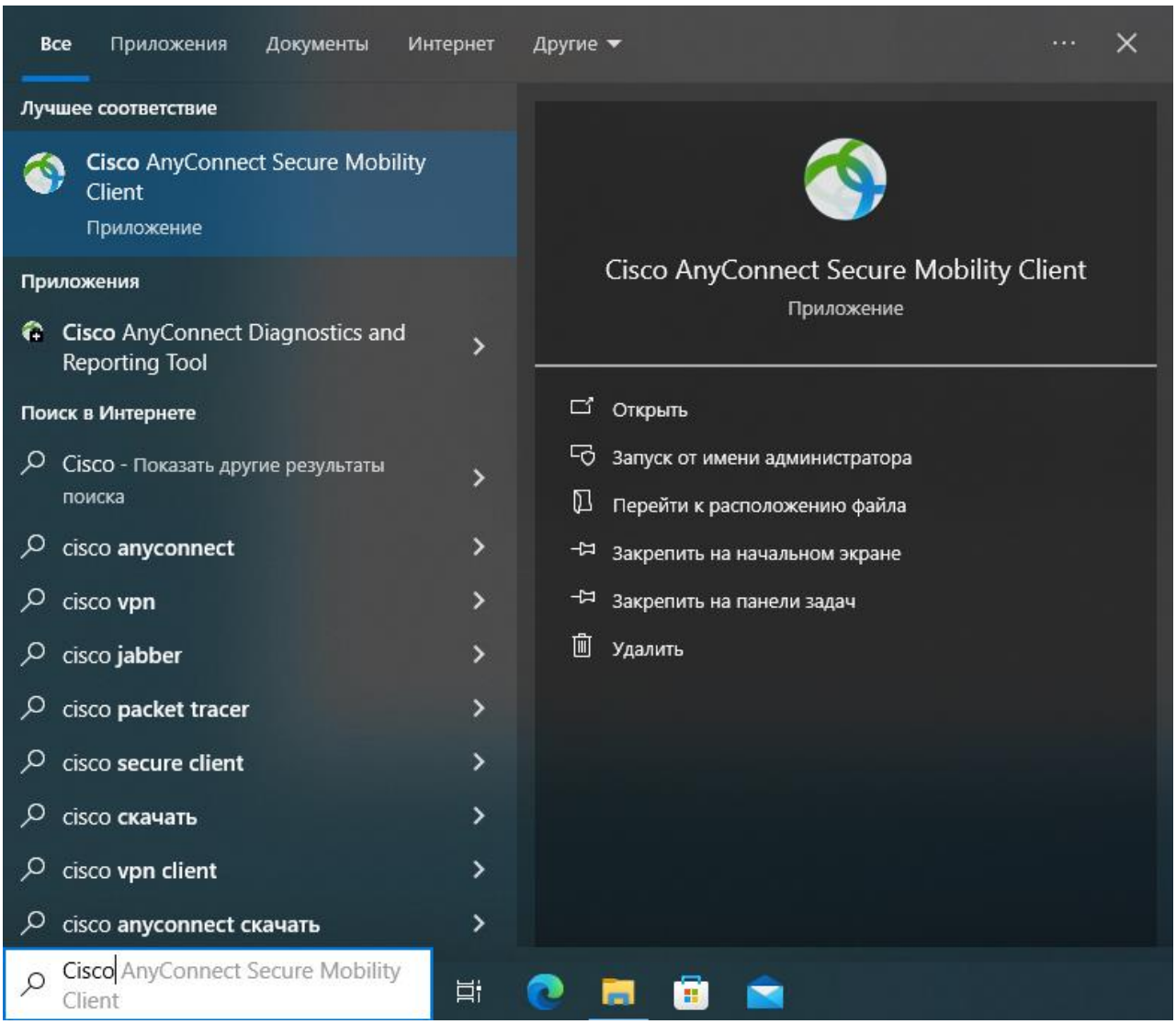


Рисунок 85

Создание VPN-подключения на AnyConnect

Для настройки AnyConnect достаточно создать файл конфигурации, где будут указаны основные параметры для построения защищенного соединения с криптошлюзом. Для этого выполните описанные ниже шаги:

1. Создайте файл конфигурации под именем `VPN.xml` по пути `C:\ProgramData\Cisco\ Cisco AnyConnect Secure Mobility Client\Profile\`:

```
<?xml version="1.0" encoding="UTF-8"?>
<AnyConnectProfile xmlns="http://schemas.xmlsoap.org/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/encoding/ AnyConnectProfile.xsd">
        <ClientInitialization>
                <UseStartBeforeLogon UserControllable="true">true</UseStartBeforeLogon>
                <AutomaticCertSelection UserControllable="false">true</AutomaticCertSelection>
                <ShowPreConnectMessage>false</ShowPreConnectMessage>
                <CertificateStore>All</CertificateStore>
                <CertificateStoreMac>All</CertificateStoreMac>
                <CertificateStoreLinux>All</CertificateStoreLinux>
                <CertificateStoreOverride>false</CertificateStoreOverride>
                <ProxySettings>Native</ProxySettings>
                <AllowLocalProxyConnections>true</AllowLocalProxyConnections>
                <AuthenticationTimeout>30</AuthenticationTimeout>
```

```
                <AutoConnectOnStart UserControllable="true">false</AutoConnectOnStart>
                <MinimizeOnConnect UserControllable="true">true</MinimizeOnConnect>
                <LocalLanAccess UserControllable="false">true</LocalLanAccess>
                <DisableCaptivePortalDetection UserControllable="true">false</DisableCaptivePortalDetection>
                <ClearSmartcardPin UserControllable="false">true</ClearSmartcardPin>
                <IPProtocolSupport>IPv4</IPProtocolSupport>
                <AutoReconnect UserControllable="false">true
                        <AutoReconnectBehavior
UserControllable="false">ReconnectAfterResume</AutoReconnectBehavior>
                </AutoReconnect>
                <SuspendOnConnectedStandby>false</SuspendOnConnectedStandby>
                <AutoUpdate UserControllable="false">true</AutoUpdate>
                <RSASecurIDIntegration UserControllable="false">Automatic</RSASecurIDIntegration>
                <WindowsLogonEnforcement>SingleLocalLogon</WindowsLogonEnforcement>
                <LinuxLogonEnforcement>SingleLocalLogon</LinuxLogonEnforcement>
                <WindowsVPNEstablishment>LocalUsersOnly</WindowsVPNEstablishment>
                <LinuxVPNEstablishment>LocalUsersOnly</LinuxVPNEstablishment>
                <AutomaticVPNPolicy>false</AutomaticVPNPolicy>
                <PPPExclusion UserControllable="false">Disable
                        <PPPExclusionServerIP UserControllable="false"></PPPExclusionServerIP>
                </PPPExclusion>
                <EnableScripting UserControllable="false">false</EnableScripting>
                <EnableAutomaticServerSelection UserControllable="false">false
                        <AutoServerSelectionImprovement>20</AutoServerSelectionImprovement>
                        <AutoServerSelectionSuspendTime>4</AutoServerSelectionSuspendTime>
                </EnableAutomaticServerSelection>
                <RetainVpnOnLogoff>false
                </RetainVpnOnLogoff>
                <CaptivePortalRemediationBrowserFailover>false</CaptivePortalRemediationBrowserFailover>
                <AllowManualHostInput>true</AllowManualHostInput>
        </ClientInitialization>
        <ServerList>
                <HostEntry>
                        <HostName>IKEv2-AnyConnect</HostName>
                        <HostAddress>10.0.102.25</HostAddress>
                        <PrimaryProtocol>IPsec
                                <StandardAuthenticationOnly>true
                                        <AuthMethodDuringIKENegotiation>IKE-
RSA</AuthMethodDuringIKENegotiation>
                                </StandardAuthenticationOnly>
                        </PrimaryProtocol>
                </HostEntry>
        </ServerList>
</AnyConnectProfile>
```

Где:

- `<HostName>IKEv2-AnyConnect</HostName>` – имя VPN соединения;
- `<HostAddress>10.0.102.25</HostAddress>` – IP адрес WAN интерфейса криптошлюза;
- `<PrimaryProtocol>IPsec…</PrimaryProtocol>` – протокол шифрования;
- `<StandardAuthenticationOnly>true…</StandardAuthenticationOnly>` – включает аутентификацию;
- `<AuthMethodDuringIKENegotiation>IKE-RSA</AuthMethodDuringIKENegotiation>` – выберите метод аутентификации.

В Приложении представлен текст конфигурационного файла для AnyConnect.

2. Проверьте, что AnyConnect создал VPN соединение под именем «IKEv2-AnyConnect» по конфигурационному файлу `VPN.xml`. Для это запустите AnyConnect (см. Рисунок 85) и найдите наличие созданного VPN соединения (см. Рисунок 86):

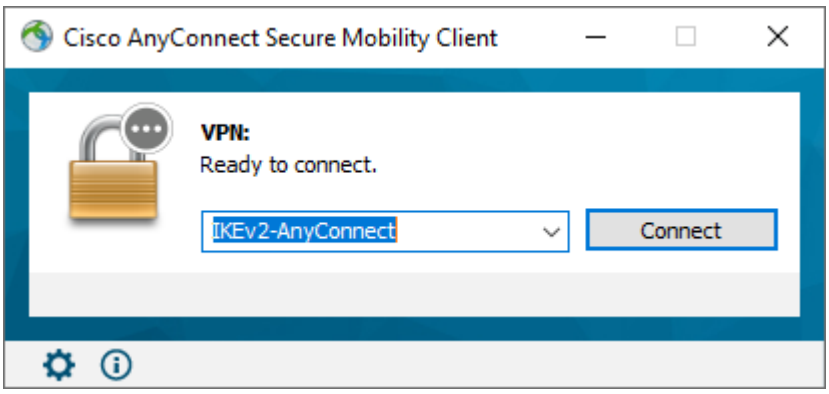


Рисунок 86

Создавать VPN-подключение на AnyConnect и на Windows одновременно на одном устройстве не рекомендуется, так как данный сценарий представляет настройку для каждого отдельного клиента или филиала.

Дополнение

Перешифрование в центре

Для настройки безопасного взаимодействия подсетей филиалов между собой с перешифрованием в центре требуется разрешить трафик между филиалами.

1. Измените конфигурационный файл LSP на криптошлюзе, для этого отредактируйте файл `lsp.txt`. Замените структуру `FilterChain` на следующую:

```
FilterChain IPsecPolicy:VPN (
      Filters                   = Filter (
            ProtocolID          = 17
            SourcePort          = 500, 4500
            Action              = PASS
            PacketType          = LOCAL_UNICAST, LOCAL_MISDIRECTED
      ),
      Filter (
            SourceIP            = 192.168.100.0/24
            DestinationIP       = 192.168.1.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:1 >
            LogEventID          = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_CISCO"
      ),
      Filter (
            SourceIP            = 192.168.2.0/24
            DestinationIP       = 192.168.1.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:1 >
            LogEventID          = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_CISCO"
      ),
      Filter (
            SourceIP            = 192.168.100.0/24
            DestinationIP       = 192.168.2.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:2 >
            LogEventID          =
"IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_AND_STRONGSWAN"
      ),
      Filter (
            SourceIP            = 192.168.1.0/24
            DestinationIP       = 192.168.2.0/24
```

```
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:2 >
            LogEventID          =
"IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_AND_STRONGSWAN"
      ),
      Filter (
            DestinationIP       = 192.168.255.0/24
            Action              = PASS
            ExtendedAction      =    ipsec<    sa    =    IPsecAction2:VPN:3:DMAP:1,
IPsecAction2:VPN:3:DMAP:2 >
            LogEventID          =
"IPsec:Protect:VPN:3:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS,                VPN:3:DMAP:2:
IPSEC_ACL_HUB1_AND_CLIENTS"
      )
)
```

В новой структуре `FilterChain` будет добавлен IPv4 трафик от филиала №1 к филиалу №2 и трафик от филиала №2 к филиалу №1, который нужно защищать с помощью списка правил пакетной и контекстной фильтрации.

2. Загрузите конфигурационный файл LSP в базу продукта:

```
root@Hub1:~#lsp_mgr load -f lsp.txt
```

```
LSP successfully loaded from file "lsp.txt"
```

3. Измените на устройстве Cisco филиала №1 список доступа (ACL). Добавьте IPv4 трафик от филиала №1 к филиалу №2, который нужно защищать:

```
Cisco(config)#ip access-list extended IPSEC_ACL_CISCO_AND_HUB1
Cisco(config-ext-nacl)#20 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Cisco(config-ext-nacl)#end
```

4. Измените конфигурационный файл `/etc/ipsec.conf` на устройстве StrongSwan филиала №2. Добавьте в конец файла отдельное соединение с подсетью Cisco, которую нужно защищать (`right` - IP адрес криптошлюза):

```
conn StronSwan-Cisco-Cert
        rightid=%any
        right=10.0.102.25
        rightsubnet=192.168.1.0/24
```

5. Перезапустите StongSwan демон:

```
root@StrongSwan:~#ipsec restart
```

```
Stopping strongSwan IPsec...
Starting strongSwan 5.9.8 IPsec [starter]...
```

Проверка работоспособности стенда

Проверка PKI

Проверку работоспособности нужно выполнять после завершения настройки всех устройств стенда.

Проверьте, что на криптошлюзе Hub1 статус всех сертификатов Active.

```
root@Hub1:~/ca#cert_mgr show
```

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

```
root@Hub1:~/ca#cert_mgr check
```

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

Видно, что на криптошлюзе статус всех сертификатов Active. Если статус Inactive, то проверьте правильны и одинаковы ли дата и время на криптошлюзе.

Проверка построения защищенного соединения между подсетями Cisco и Hub1

1. Инициируйте построение защищенного соединения между Cisco и Hub1 при помощи ICMP трафика, посылаемого с устройства Host-b-Cisco на Host-b-Hub1. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-Cisco.

```
root@Host-b-Cisco:~# 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=4 ttl=62 time=2.53 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=1.62 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 2 received, 60% packet loss, time 4056ms
```

2. Проверьте, что на Hub1 и Cisco установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюза и команду `show crypto session` на Cisco:

```
root@Hub1:~/ca# sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (10.0.102.25,500)-(10.0.108.195,500) active 2589 2890

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.100.0-192.168.100.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 192 192
```

```
Cisco#show crypto session
```

```
Crypto session current status

Interface: GigabitEthernet1
Profile: CISCO_ALGS
Session status: UP-ACTIVE
Peer: 10.0.102.25 port 500
  Session ID: 1
  IKEv2 SA: local 10.0.108.195/500 remote 10.0.102.25/500 Active
  IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 192.168.2.0/255.255.255.0
        Active SAs: 0, origin: crypto map
  IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 192.168.100.0/255.255.255.0
        Active SAs: 2, origin: crypto map
```

Видно, что на Hub1 и Cisco установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`) на Hub1. При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетями StrongSwan и Hub1

1. Инициируйте построение защищенного соединения между StrongSwan и Hub1 при помощи ICMP трафика, посылаемого с устройства Host-b-StrongSwan на Host-b-Hub1. Для этого выполните команду `ping` из linux bash консоли защищаемого устройства Host-b-StrongSwan.

```
root@Host-b-StrongSwan:~# ping 192.168.100.100 -c 5
```

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
```

```
64 bytes from 192.168.100.100: icmp_seq=1 ttl=62 time=945 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=62 time=1.10 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=62 time=2.18 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=62 time=1.13 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=62 time=1.26 ms

--- 192.168.100.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.098/190.190/945.287/377.548 ms
```

2. Проверьте, что на Hub1 и StrongSwan установлено защищенное IPsec соединение. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюза и команду `ipsec status` на StrongSwan.

```
root@Hub1:~/ca# sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (10.0.102.25,500)-(10.0.108.237,500) active 3620 3529

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.100.0-192.168.100.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 480 480
```

```
root@StrongSwan:~#ipsec status
```

```
Routed Connections:
StronSwan-Cisco-Cert{2}:  ROUTED, TUNNEL, reqid 2
StronSwan-Cisco-Cert{2}:   192.168.2.0/24 === 192.168.1.0/24
StronSwan-Gate-Cert{1}:  ROUTED, TUNNEL, reqid 1
StronSwan-Gate-Cert{1}:   192.168.2.0/24 === 192.168.100.0/24
Security Associations (1 up, 0 connecting):
StronSwan-Gate-Cert[1]: ESTABLISHED 69 seconds ago, 10.0.108.237[C=RU, L=Zelenograd,
O=S-Terra CSP, CN=StrongSwan]... 10.0.102.25[C=RU, L=Zelenograd, O=S-Terra CSP, OU=RnD,
CN=Hub1]
StronSwan-Gate-Cert{3}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c35bb5e0_i 7483fd6d_o
StronSwan-Gate-Cert{3}:   192.168.2.0/24 === 192.168.100.0/24
```

Видно, что на Hub1 и StrongSwan установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`) на Hub1. При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетями Hub1 и Android

1. В случае, если VPN-профиль был создан в соответствии с разделом «Создание VPN профиля на Android», то выполните следующи шаги

   1.1. Найдите в настройках «VPN» при помощи поиска и перейдите по результату (см. Рисунок 87):

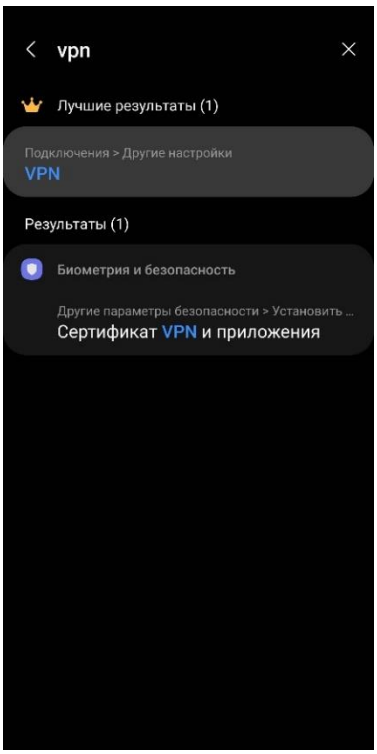


Рисунок 87

1.2. В открывшемся окне перейдите во вкладку «VPN» (см. Рисунок 88):

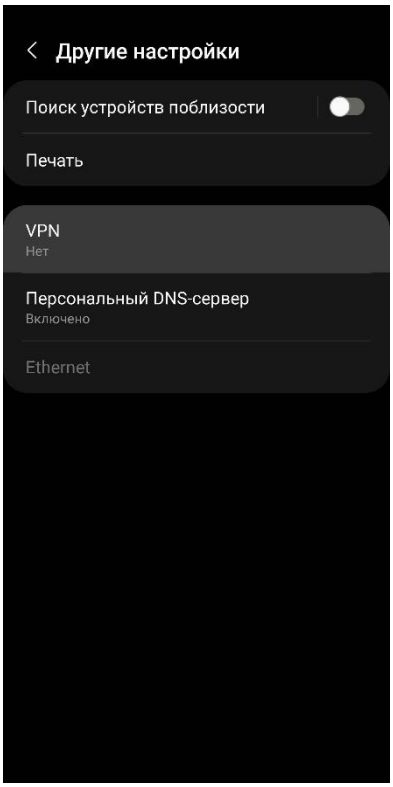


Рисунок 88

1.3. В открывшемся окне выберите созданный в разделе «Создание VPN профиля на Android» VPN профиль «Android Certificate» (см. Рисунок 89):

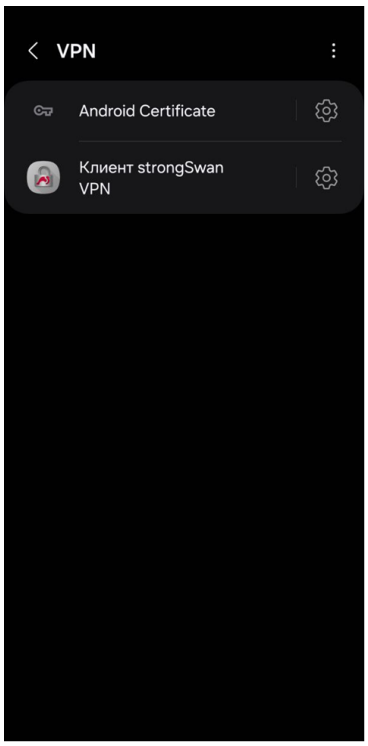


Рисунок 89

1.4. Во всплывающем окне нажмите кнопку «Подключиться», чтобы подключиться к криптошлюзу (см. Рисунок 90):

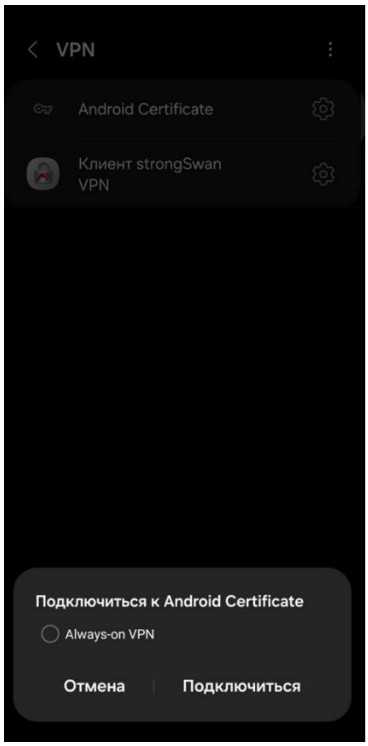


Рисунок 90

1.5. После успешного подключения под созданным профилем будет статус «Подключено» (см. Рисунок 91):

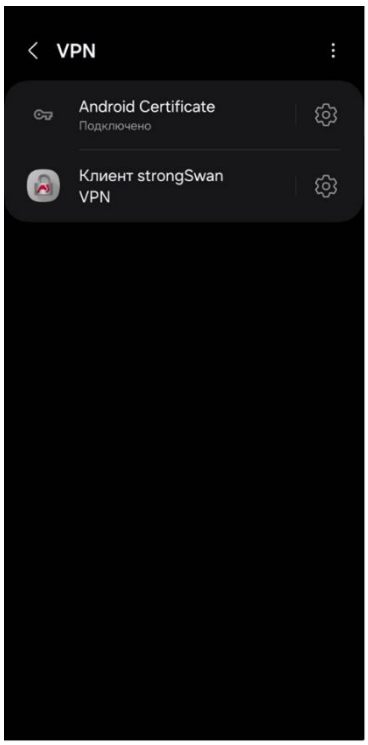


Рисунок 91

Если этого не произошло, то проверьте настройки устройства в соответствии с разделом «Создание VPN профиля на Android». Если после проверки настройки статус профиля VPN при подключении не изменился на «Подключено», то создайте VPN-профиль согласно разделу «Создание VPN профиля на Android при помощи «StrongSwan VPN Client»».

2. В случае, если VPN-профиль был создан в соответствии с разделом «Создание VPN профиля на Android при помощи «StrongSwan VPN Client»», то выполните следующи шаги:

   2.1. Войдите в приложение «StrongSwan VPN Client» (см. Рисунок 92)

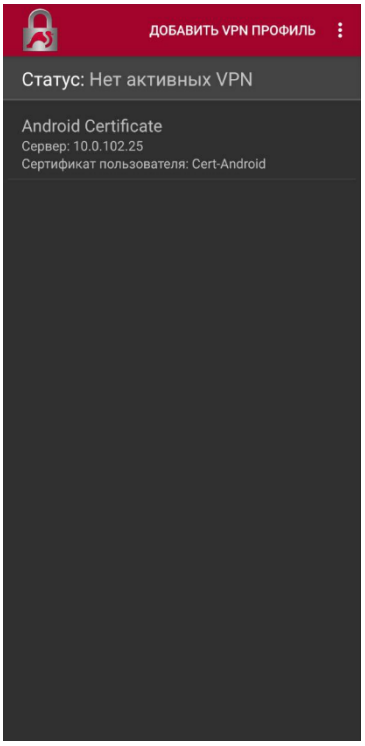


Рисунок 92

2.2. В открывшемся окне выберите ранее созданый VPN профиль «Android Certificate» (см. Рисунок 93):

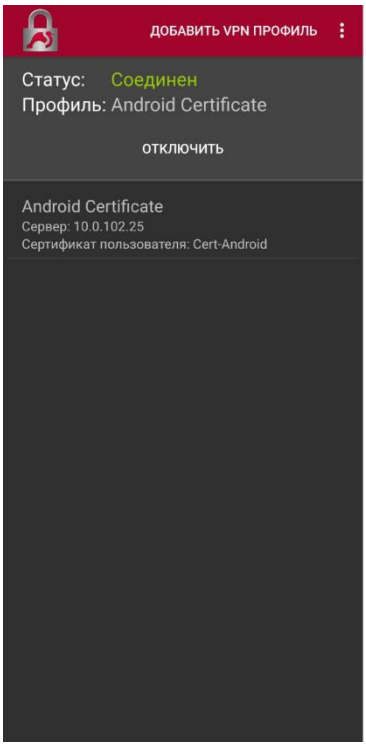


Рисунок 93

Если этого не произошло, то проверьте настройки устройства в соответствии с разделом «Создание VPN профиля на Android при помощи «StrongSwan VPN Client»».

3. После построения защищенного соединения между Android и криптошлюзом можно проверить работоспособность при помощи SSH-соединения. Для этого подключитесь по SSH к Host-behind-Hub1 с Android (подключение со стороннего приложения на Android для работы с терминалом см. Рисунок 94):

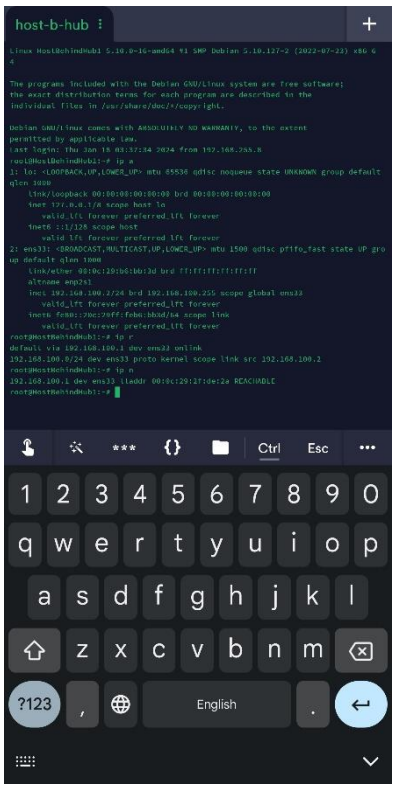


Рисунок 94

4. Также можно подключиться к криптошлюзу (см. Рисунок 95):



Рисунок 95

5. Проверьте, что на криптошлюзе установлено защищенное IKEv2/IPsec соединение.

```
root@Hub1:~/ca# sa_mgr show
ISAKMP sessions: 0 initiated, 0 responded
```

```
IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 2 (10.0.102.25,4500)-(10.140.1.163,34818) active 2657 3340

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 5776 6448
```

Разница в принятой и переданной информации (в байтах) по IPsec обусловлена тем, что в Android: при настроенном split tunneling кроме трафика может передаватся часть служебных данных (например, DNS-запросы по IPsec, иначе весь трафик маршрутизируется по IPsec.

Видно, что между криптошлюзом Hub1 и Android установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетью Hub1 и пользовательского компьютера с ОС Windows

1. В панели задач выберите значок сети, потом выберите VPN-подключение «IKEv2-IPsec-RSA» и нажмите кнопку «Подключиться» (см. Рисунок 96):

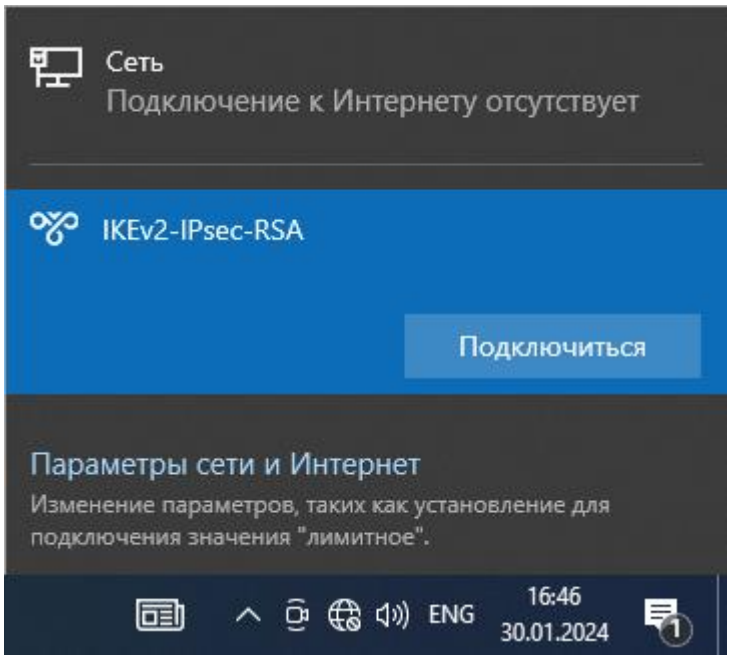


Рисунок 96

2. После успешного соединения статус соединения должен измениться на «Подключено» (см. Рисунок 97):

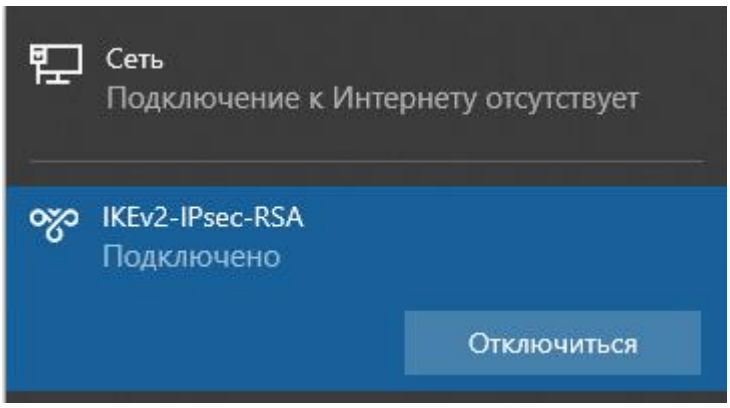


Рисунок 97

3. Проверьте с помощью ICMP трафика построение защищенного соединения на западных алгоритмах, для этого в командной строке выполните команду `ping` до хоста за криптошлюзом:

```
C:\Users\User>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=1мс TTL=63
Ответ от 192.168.100.100: число байт=32 время=3мс TTL=63
Ответ от 192.168.100.100: число байт=32 время=1мс TTL=63
Ответ от 192.168.100.100: число байт=32 время=4мс TTL=63

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 = 4 мсек, Среднее = 2 мсек
```

4. Проверьте, что на криптошлюзе установлено защищенное IKEv2/IPsec соединение.

```
root@Hub1:~/ca# sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (10.0.102.25,500)-(10.0.107.16,500) active 2381 4396

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(192.168.255.1,*) * ESP tunn 256 9264
```

Разница в принятой и переданной информации (в байтах) по IPsec обусловлена тем, что в Windows: при настроенном split tunneling кроме трафика может передаватся часть служебных данных (например, DNS-запросы по IPsec, иначе весь трафик маршрутизируется по IPsec.

Видно, что между криптошлюзом и компьютером пользователя с ОС Windows установлено защищенное IKEv2/IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетями Hub1 и iOS

1. В системных настройках в меню «VPN» выберите добавленную ранее VPN конфигурацию и нажмите на ползунок включения (см. Рисунок 98):

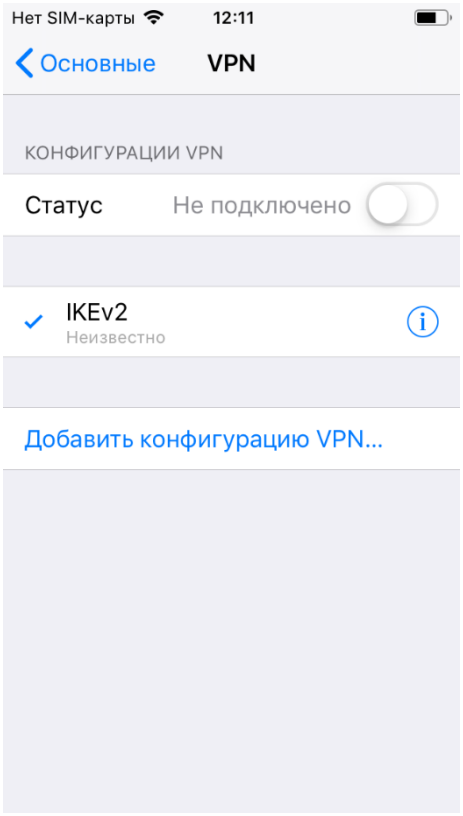


Рисунок 98

2. После успешного соединения статус соединения должен измениться на «Подключено» (см. Рисунок 99):

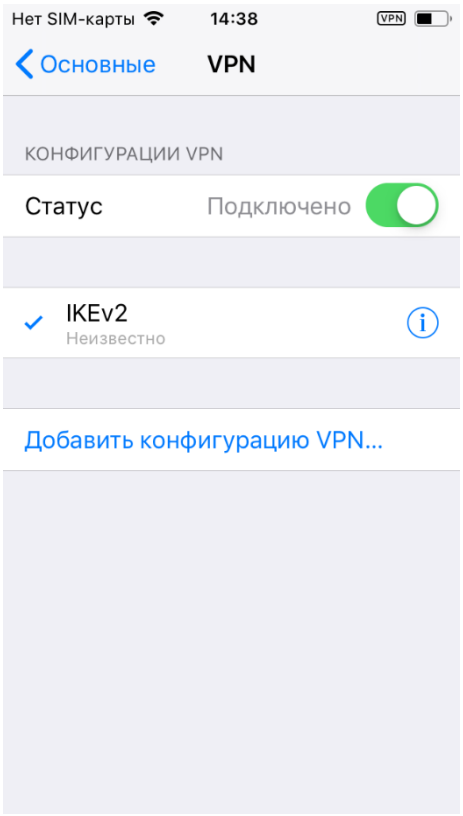


Рисунок 99

Если этого не произошло, то проверьте настройки устройства в соответствии с разделом «Создание VPN-подключения на iOS».

3. Проверьте с помощью ICMP трафика построение защищенного соединения на западных алгоритмах, для этого в командной строке выполните команду `ping` с хоста за криптошлюзом:

```
root@Host-b-Hub1:~# ping -c 5 192.168.255.1
```

```
PING 192.168.255.1 (192.168.255.1) 56(84) bytes of data.
64 bytes from 192.168.255.1: icmp_seq=1 ttl=64 time=118 ms
64 bytes from 192.168.255.1: icmp_seq=2 ttl=64 time=141 ms
64 bytes from 192.168.255.1: icmp_seq=3 ttl=64 time=61.1 ms
64 bytes from 192.168.255.1: icmp_seq=4 ttl=64 time=82.6 ms
64 bytes from 192.168.255.1: icmp_seq=5 ttl=64 time=106 ms

--- 192.168.255.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 61.107/101.715/141.144/27.762 ms
```

4. Проверьте, что на криптошлюзе установлено защищенное IKEv2/IPsec соединение.

```
root@Hub1:~# sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 2 (10.0.102.25,4500)-(10.140.1.175,4500) active 2405 2578

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 1440 16288
```

Разница в принятой и переданной информации (в байтах) по IPsec обусловлена тем, что iOS перенаправляет весь трафик в IPsec туннель.

Видно, что между криптошлюзом Hub1 и iOS установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетями Hub1 и macOS

1. В системных настройках в меню «VPN» выберите добавленную ранее VPN конфигурацию и нажмите на ползунок включения (см. Рисунок 100):

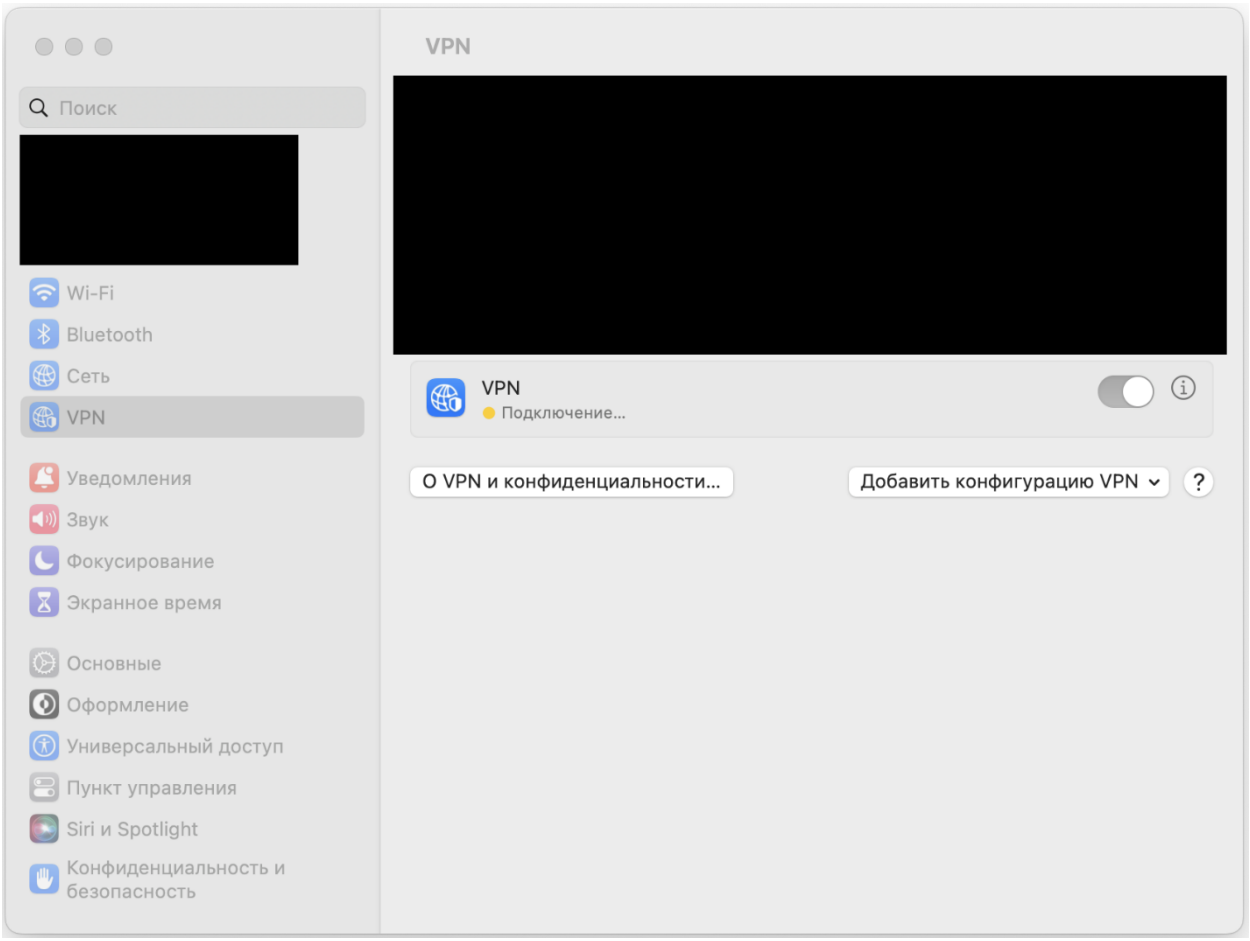


Рисунок 100

2. Введите пароль от устройства (см. Рисунок 101):

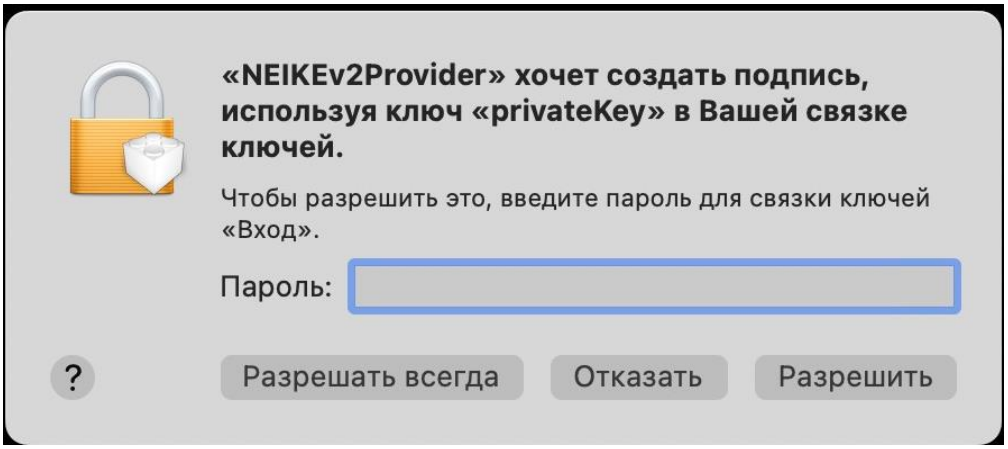


Рисунок 101

3. После успешного соединения статус соединения должен измениться на «Подключено в течение …» (см. Рисунок 102):

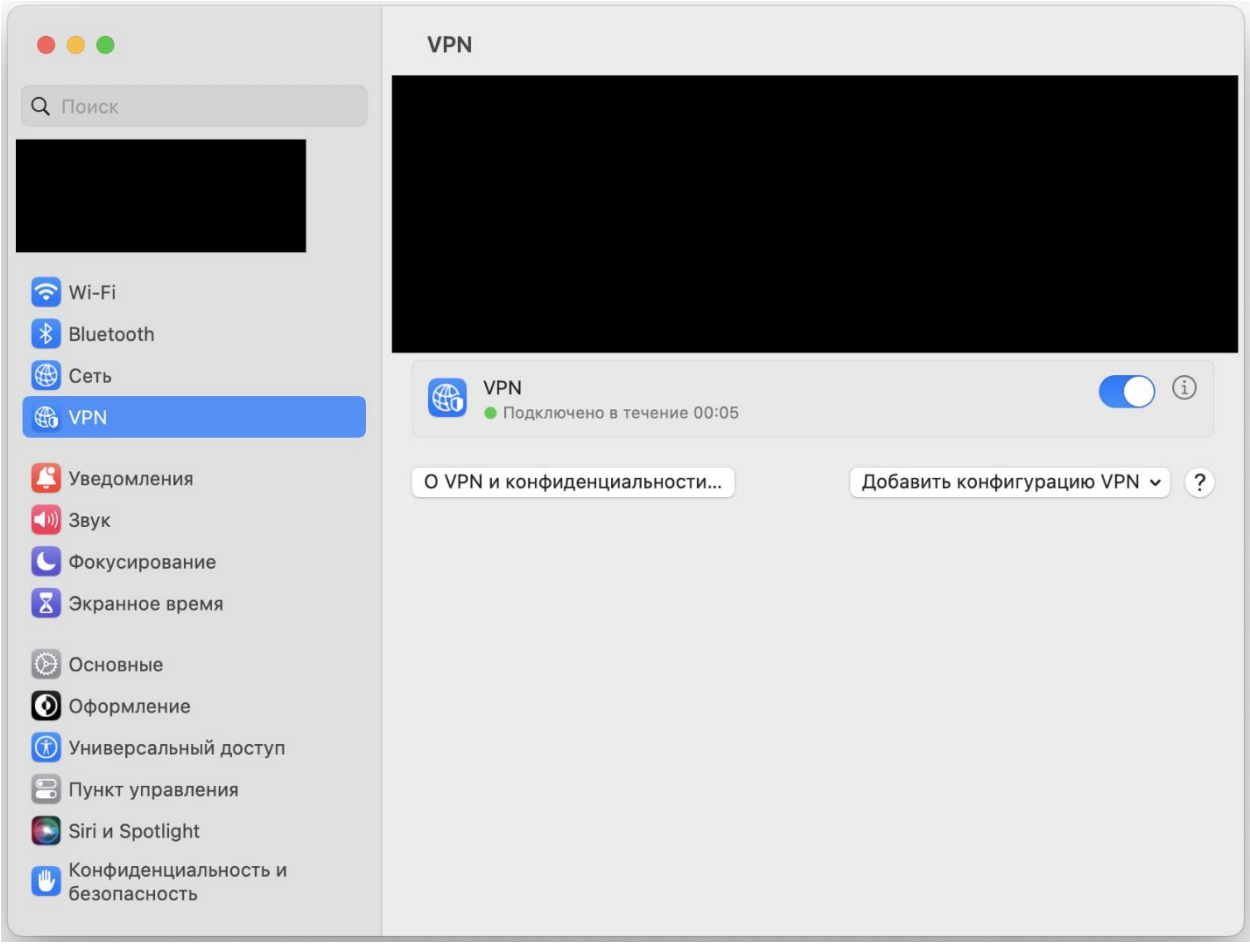


Рисунок 102

Если этого не произошло, то проверьте настройки устройства в соответствии с разделом «Создание VPN-подключения на macOS».

4. Проверьте с помощью ICMP трафика построение защищенного соединения на западных алгоритмах, для этого в командной строке выполните команду `ping` до Host-b-Hub1:

```
~  ping 192.168.100.100 -c 5                                    ✔ 12:03:15
PING 192.168.100.100 (192.168.100.100): 56 data bytes
64 bytes from 192.168.100.100: icmp_seq=0 ttl=64 time=5.542 ms
64 bytes from 192.168.100.100: icmp_seq=1 ttl=64 time=11.626 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=64 time=6.018 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=64 time=11.097 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=64 time=13.340 ms

--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 5.542/9.525/13.340/3.150 ms
```

5. Проверьте, что на криптошлюзе установлено защищенное IKEv2/IPsec соединение.

```
root@Hub1:~# sa_mgr show
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 2 (10.0.102.25,4500)-(10.140.1.175,4500) active 2405 2578

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 1440 16288
```

Разница в принятой и переданной информации (в байтах) по IPsec обусловлена тем, что macOS перенаправляет весь трафик в IPsec туннель.

Видно, что между криптошлюзом Hub1 и macOS установлено защищенное IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетями Hub1 и AnyConnect

1. На панели задач откройте поиск и введите «Cisco». После появления «Cisco AnyConnect Secure Mobility Client», нажмите кнопку «Открыть» (см. Рисунок 103):

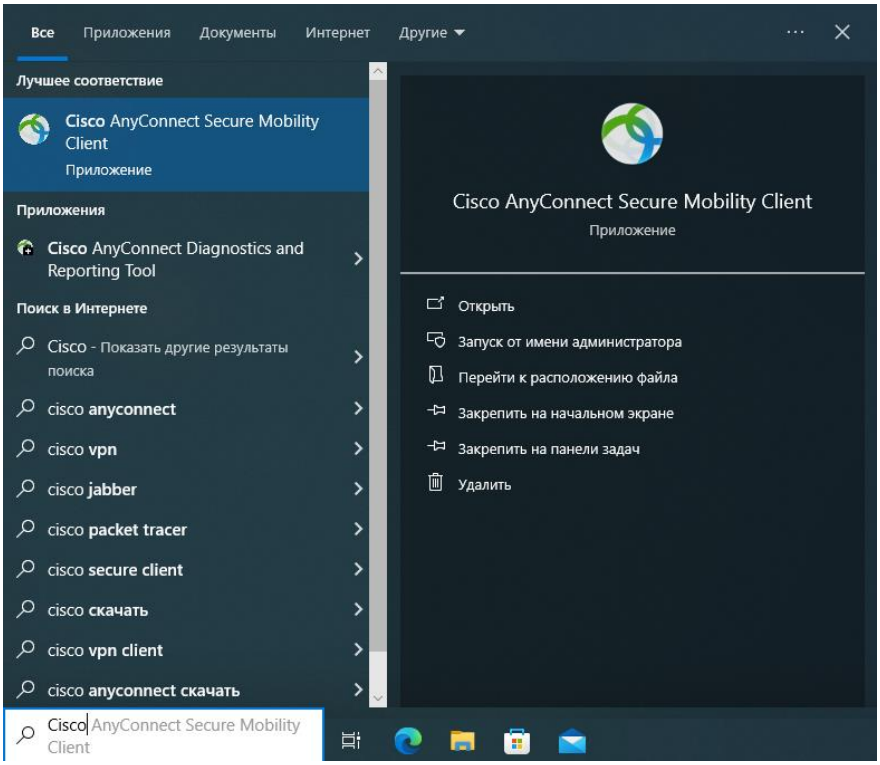


Рисунок 103

2. В открывшемся окне «Cisco AnyConnect Secure Mobility Client» нажмите кнопку «Connect» (см. Рисунок 104):

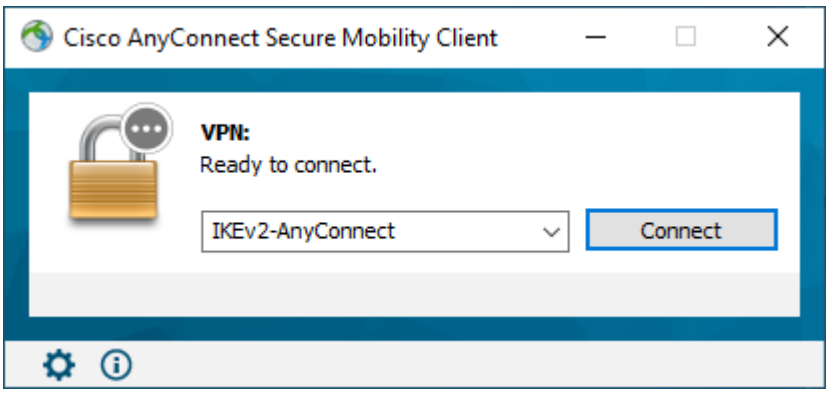


Рисунок 104

3. После успешного соединения окно «Cisco AnyConnect Secure Mobility Client» и в окне отобразится уведомление с успешным подключением (см. Рисунок 105):



Рисунок 105

Если этого не произошло, повторите подключение. Если вновь не отобразится уведомление об успешном подключении, проверьте настройки устройства в соответствии с разделом «Создание VPN подключения на AnyConnect».

4. Проверьте с помощью ICMP трафика построение защищенного соединения на западных алгоритмах, для этого в командной строке выполните команду `ping` до хоста за криптошлюзом:

```
C:\Users\User>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=2мс TTL=63
Ответ от 192.168.100.100: число байт=32 время=2мс TTL=63
Ответ от 192.168.100.100: число байт=32 время=1мс TTL=63
Ответ от 192.168.100.100: число байт=32 время=1мс TTL=63

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 = 2 мсек, Среднее = 1 мсек
```

5. Проверьте, что на криптошлюзе установлено защищенное IKEv2/IPsec соединение:

```
root@Hub1:~# sa_mgr show

ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 2 (10.0.102.25,4500)-(10.0.105.146,57578) active 2440 4093

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (0.0.0.0-255.255.255.255,*)-(192.168.255.1,*) * ESP nat-t-tunn 256 20656
```

Разница в принятой и переданной информации (в байтах) по IPsec обусловлена тем, что AnyConnect перенаправляет весь трафик в IPsec туннель. Реализации управления маршрутами на AnyConnect с криптошлюза в версии 5.0 нет.

Видно, что между криптошлюзом и AnyConnect установлено защищенное IKEv2/IPsec соединение. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Проверка построения защищенного соединения между подсетями Cisco и StrongSwan

Проверку построения защищенного соединения между подсетями Cisco и StrongSwan следует проводить при настройке перешифрования в центре.

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

```
root@Host-b-Cisco:~# 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=3 ttl=61 time=736 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=61 time=2.16 ms
64 bytes from 192.168.2.100: icmp_seq=5 ttl=61 time=2.70 ms

--- 192.168.2.100 ping statistics ---
5 packets transmitted, 3 received, 40% packet loss, time 4029ms
rtt min/avg/max/mdev = 2.156/247.023/736.211/345.907 ms
```

2. Проверьте, что на криптошлюзе Hub1 установлено защищенное IPsec соединение и изменилось количество байт, переданных по защищенному IPsec соединению. Для этого выполните команду `run sa_mgr show` из cisco-like консоли криптошлюза, команду `show crypto session на` Cisco и команду `ipsec status` на StrongSwan.

```
root@Hub1:~# sa_mgr show
```

```
ISAKMP sessions: 0 initiated, 0 responded

IKEv2 connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) State Sent Rcvd
1 1 (10.0.102.25,500)-(10.0.108.195,500) active 2509 2810
2 2 (10.0.102.25,500)-(10.0.108.237,500) active 2564 2393

IPsec connections:
Num Conn-id (Local Addr,Port)-(Remote Addr,Port) Protocol Action Type Sent Rcvd
1 1 (192.168.2.0-192.168.2.255,*)-(192.168.1.0-192.168.1.255,*) * ESP tunn 384 384
2 2 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) * ESP tunn 384 384
```

```
Cisco#show crypto session
```

```
Crypto session current status

Interface: GigabitEthernet1
Profile: CISCO_ALGS
Session status: UP-ACTIVE
Peer: 10.0.102.25 port 500
  Session ID: 1
  IKEv2 SA: local 10.0.108.195/500 remote 10.0.102.25/500 Active
  IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 192.168.2.0/255.255.255.0
        Active SAs: 2, origin: crypto map
  IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 192.168.100.0/255.255.255.0
        Active SAs: 0, origin: crypto map
```

```
root@StrongSwan:~# ipsec status
```

```
Routed Connections:
StronSwan-Cisco-Cert{2}:  ROUTED, TUNNEL, reqid 2
StronSwan-Cisco-Cert{2}:   192.168.2.0/24 === 192.168.1.0/24
StronSwan-Gate-Cert{1}:  ROUTED, TUNNEL, reqid 1
StronSwan-Gate-Cert{1}:   192.168.2.0/24 === 192.168.100.0/24
Security Associations (1 up, 0 connecting):
```

```
StronSwan-Gate-Cert[2]: ESTABLISHED 107 seconds ago, 10.0.108.237[C=RU, O=S-Terra CSP,
CN=StrongSwan]... 10.0.102.25 [C=RU, L=Zelenograd, O=S-Terra CSP, OU=RnD, CN=Hub1]
StronSwan-Cisco-Cert{4}:  INSTALLED, TUNNEL, reqid 2, ESP SPIs: ca798a46_i f2c0a0f3_o
StronSwan-Cisco-Cert{4}:   192.168.2.0/24 === 192.168.1.0/24
```

Видно, что между криптошлюзами Hub1, Cisco и StrongSwan установлено защищенное IPsec соединение и количество полученных и отправленных пакетов изменилось на счётчиках обоих IPsec соединений. Если этого не произошло, то проверьте файл журнала (команда `run less /var/log/cspvpngate.log`). При необходимости, увеличьте уровень логирования (команда `logging trap debugging` в консоли cisco-like) и заново инициируйте защищенное соединение.

Приложение

Конфигурации криптошлюза Hub1 версии 5.0.26253 (5.0 BF) и выше

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity address
username          cscons          privilege          15          secret          5
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Hub1
enable secret 5 PC9d7N5HlAyLrzuA3qRJvQ==
!
!
!
!
!
ipv6 default filtering deny
!
!
interface GigabitEthernet0/0
 ip address 10.0.102.25 255.255.0.0
!
interface GigabitEthernet0/1
 ip address 192.168.100.1 255.255.255.0
!
!
!
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

2. Конфигурация LSP:

```
GlobalParameters(
      Title                     = "Hub1 5.0 LSP IKEv2"
      Version                   = LSP_5_0
      CRLHandlingMode           = DISABLE
      PreserveIPsecSA           = FALSE
)

IKEAuth2 IKEV2-AUTH-RSA-SIG-LOCAL (
      Identity                  = IdentityEntry(
            DistinguishedName*  = USER_SPECIFIC_DATA)
      Credential                = CertDescription()
      SendCertMode              = ALWAYS
      SendRequestMode           = ALWAYS
)

IKEAuth2 IKEV2-AUTH-RSA-SIG-REMOTE (
      Credential                = CertDescription()
)

IdentityEntry ANYCONNECT (
      DistinguishedName         = CertDescription(
      Subject                   = TEMPLATE,"L=Zelenograd,O=S-Terra CSP,CN=AnyConnect"
      )
)

IKEAuth2 IKEV2-AUTH-RSA-SIG-ANYCONNECT (
      Identity                  = ANYCONNECT
      Credential                = CertDescription()
)

AddressPool IKECFG_POOL_CLIENTS(
      IPAddresses               = 192.168.255.1..192.168.255.254
      DNSServers                = 8.8.8.8

)

IKEProposal2 IKE2-PROPOSAL (
      EncryptionAlg             = ENCR_AES_CBC_256
      IntegrityAlg              = AUTH_HMAC_SHA2_256_128
      PrfAlg                    = PRF_HMAC_SHA2_256
      KeyAlg                    = MODP_2048
)

IKERule2 IKERule2:VPN:1
(
      IKEPeerIPFilter           = 10.0.108.195
      Proposals                 = IKE2-PROPOSAL
      LocalAuth                 = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                = IKEV2-AUTH-RSA-SIG-REMOTE
      DPDMode                   = PERIODIC
      DPDIdleDuration           = 5
      Priority                  = 10
)

IKERule2 IKERule2:VPN:2
(
      IKEPeerIPFilter           = 10.0.108.237
      Proposals                 = IKE2-PROPOSAL
      LocalAuth                 = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                = IKEV2-AUTH-RSA-SIG-REMOTE
```

```
      DPDMode                    = PERIODIC
      DPDIdleDuration            = 5
      Priority                   = 20
)

IKERule2 IKERule2:VPN:3:DMAP:1
(
      Proposals                  = IKE2-PROPOSAL
      LocalAuth                  = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                 = IKEV2-AUTH-RSA-SIG-ANYCONNECT
      IKECFGPool                 = IKECFG_POOL_CLIENTS
      DPDMode                    = PERIODIC
      DPDIdleDuration            = 14
      Priority                   = 100
)

IKERule2 IKERule2:VPN:3:DMAP:2
(
      Proposals                  = IKE2-PROPOSAL
      LocalAuth                  = IKEV2-AUTH-RSA-SIG-LOCAL
      RemoteAuth                 = IKEV2-AUTH-RSA-SIG-REMOTE
      IKECFGPool                 = IKECFG_POOL_CLIENTS
      DPDMode                    = PERIODIC
      DPDIdleDuration            = 14
      Priority                   = 110
)

IPsecProposal2 IKE2-IPSEC-PROPOSAL (
      ESPEncryptionAlg           = ENCR_AES_CBC_256
      ESPIntegrityAlg            = AUTH_HMAC_SHA2_256_128
      PFSAlg                     = MODP_2048
)

IPsecProposal2 IKE2-IPSEC-PROPOSAL-CISCO (
      ESPEncryptionAlg           = ENCR_AES_CBC_256
      ESPIntegrityAlg            = AUTH_HMAC_SHA2_256_128
)

IPsecAction2 IPsecAction2:VPN:1
(
      TunnelingParameters        = TunnelEntry(
            PeerAddress          = 10.0.108.195
            Assemble             = TRUE
      )
      ContainedProposals         = IKE2-IPSEC-PROPOSAL-CISCO
      IKERule                    = IKERule2:VPN:1
)

IPsecAction2 IPsecAction2:VPN:2(
      TunnelingParameters        = TunnelEntry(
            PeerAddress          = 10.0.108.237
            Assemble             = TRUE
      )
      ContainedProposals         = IKE2-IPSEC-PROPOSAL
      IKERule                    = IKERule2:VPN:2
)
IPsecAction2 IPsecAction2:VPN:3:DMAP:1(
      TunnelingParameters        = TunnelEntry(
             Assemble            = TRUE
      )
      ContainedProposals         = IKE2-IPSEC-PROPOSAL-CISCO
      IKERule                    = IKERule2:VPN:3:DMAP:1
```

```
)

IPsecAction2 IPsecAction2:VPN:3:DMAP:2(
      TunnelingParameters       = TunnelEntry(
            Assemble            = TRUE
      )
      ContainedProposals        = IKE2-IPSEC-PROPOSAL
      IKERule                   = IKERule2:VPN:3:DMAP:2
)

FilterChain IPsecPolicy:VPN (
      Filters                   = Filter (
            ProtocolID          = 17
            SourcePort          = 500, 4500
            Action              = PASS
            PacketType          = LOCAL_UNICAST, LOCAL_MISDIRECTED
      ),
      Filter (
            SourceIP            = 192.168.100.0/24
            DestinationIP       = 192.168.1.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:1 >
            LogEventID          = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_CISCO"
      ),
      Filter (
            SourceIP            = 192.168.100.0/24
            DestinationIP       = 192.168.2.0/24
            Action              = PASS
            ExtendedAction      = ipsec< sa = IPsecAction2:VPN:2 >
            LogEventID          =
"IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_AND_STRONGSWAN"
      ),
      Filter (
            DestinationIP       = 192.168.255.0/24
            Action              = PASS
            ExtendedAction      =    ipsec<    sa    =    IPsecAction2:VPN:3:DMAP:1,
IPsecAction2:VPN:3:DMAP:2 >
            LogEventID          =
"IPsec:Protect:VPN:3:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS,                VPN:3:DMAP:2:
IPSEC_ACL_HUB1_AND_CLIENTS"
      )
)

NetworkInterface (
            LogicalName         = "GigabitEthernet0/0"
            IPsecPolicy         = IPsecPolicy:VPN
)

NetworkInterface (
            LogicalName               = "GigabitEthernet0/1"
)
```

Конфигурация несертифицированного криптошлюза Hub1

1. Консоль cisco-like:

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity address
```

```
username          cscons          privilege          15          secret          5
$6$RiWz4IVm$qMp6sbpYnTfSUxuaJPZ9UCq4yy8/xheIEbSmbTfMzmbQqepMD6R1b18vHTdFQt949xNXzpRXW
A9dZ/gCK70z./
aaa new-model
!
!
hostname Hub1
enable secret 5 HGJcyG+CRmCjINGFkW48VQ==
!
!
!
!
!
!
!
crypto ikev2 proposal IKEV2-PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
 prf sha256
!
crypto ikev2 policy IKEV2-POLICY
 proposal IKEV2-PROPOSAL
!
crypto ikev2 authentication sig IKEV2-AUTH-RSA-SIG-LOCAL
 identity dn
 set identity-mapping auto
!
crypto ikev2 authentication sig IKEV2-AUTH-RSA-SIG-REMOTE
 set identity-mapping auto
!
crypto ikev2 authentication sig IKEV2-AUTH-RSA-SIG-ANYCONNECT
 identity dn L=Zelenograd,O=S-Terra CSP,CN=AnyConnect
 set identity-mapping auto
!
crypto ikev2 profile IKEV2-PROFILE-RSA-SIG
 authentication remote IKEV2-AUTH-RSA-SIG-REMOTE
 authentication local IKEV2-AUTH-RSA-SIG-LOCAL
 set policy IKEV2-POLICY
 dpd 5 periodic
!
crypto ikev2 profile IKEV2-PROFILE-RSA-SIG-CLIENTS
 authentication remote IKEV2-AUTH-RSA-SIG-REMOTE
 authentication local IKEV2-AUTH-RSA-SIG-LOCAL
 set policy IKEV2-POLICY
 dpd 14 periodic
!
crypto ikev2 profile IKEV2-PROFILE-RSA-SIG-ANYCONNECT
 authentication remote IKEV2-AUTH-RSA-SIG-ANYCONNECT
 authentication local IKEV2-AUTH-RSA-SIG-LOCAL
 set policy IKEV2-POLICY
 dpd 14 periodic
!
crypto ikev2 ipsec-proposal IKEV2-IPSEC-PROPOSAL
 esp encryption aes-cbc-256
 esp integrity sha256
 pfs 14
!
crypto ikev2 ipsec-proposal IKEV2-IPSEC-PROPOSAL-CISCO
 esp encryption aes-cbc-256
 esp integrity sha256
!
```

```
ip local pool IKECFG_POOL_CLIENTS 192.168.255.1 192.168.255.254
!
ip access-list extended IPSEC_ACL_HUB1_AND_CISCO
 permit ip 192.168.100.0 0.0.0.255 192.168.1.0 0.0.0.255
!
ip access-list extended IPSEC_ACL_HUB1_AND_STRONGSWAN
 permit ip 192.168.100.0 0.0.0.255 192.168.2.0 0.0.0.255
!
ip access-list extended IPSEC_ACL_HUB1_AND_CLIENTS
 permit ip any 192.168.255.0 0.0.0.255
!
ipv6 default filtering deny
!
!
crypto dynamic-map DMAP 1
 match address IPSEC_ACL_HUB1_AND_CLIENTS
 set ikev2-ipsec-proposal IKEV2-IPSEC-PROPOSAL-CISCO
 set ikev2-profile IKEV2-PROFILE-RSA-SIG-ANYCONNECT
 set pool IKECFG_POOL_CLIENTS
 set dns 8.8.8.8
 set mode tunnel
 set packets-assemble before-encapsulation
 set dead-connection history off
crypto dynamic-map DMAP 2
 match address IPSEC_ACL_HUB1_AND_CLIENTS
 set ikev2-ipsec-proposal IKEV2-IPSEC-PROPOSAL
 set ikev2-profile IKEV2-PROFILE-RSA-SIG-CLIENTS
 set pool IKECFG_POOL_CLIENTS
 set dns 8.8.8.8
 set mode tunnel
 set packets-assemble before-encapsulation
 set dead-connection history off
!
crypto map VPN 1 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_CISCO
 set ikev2-ipsec-proposal IKEV2-IPSEC-PROPOSAL-CISCO
 set ikev2-profile IKEV2-PROFILE-RSA-SIG
 set mode tunnel
 set peer 10.0.108.195
 set packets-assemble before-encapsulation
 set dead-connection history off
crypto map VPN 2 ipsec-isakmp
 match address IPSEC_ACL_HUB1_AND_STRONGSWAN
 set ikev2-ipsec-proposal IKEV2-IPSEC-PROPOSAL
 set ikev2-profile IKEV2-PROFILE-RSA-SIG
 set mode tunnel
 set peer 10.0.108.237
 set packets-assemble before-encapsulation
 set dead-connection history off
crypto map VPN 3 ipsec-isakmp dynamic DMAP
!
interface GigabitEthernet0/0
 ip address 10.0.102.25 255.255.0.0
 crypto map VPN
!
interface GigabitEthernet0/1
 ip address 192.168.100.1 255.255.255.0
!
!
!
!
crypto pki trustpoint s-terra_technological_trustpoint
```

```
 revocation-check none
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:   Thu Oct 30 08:19:59 2025

GlobalParameters(
    Title                      = "This LSP was automatically generated by CSP Converter
at Thu Oct 30 08:19:59 2025 (user: cscons)"
    Version                       = LSP_5_0
    CRLHandlingMode               = OPTIONAL
    PreserveIPsecSA               = FALSE
)

FirewallParameters(
    TCPSynSentTimeout = 30
    TCPFinTimeout = 5
    TCPClosedTimeout = 30
    TCPSynRcvdTimeout = 30
    TCPEstablishedTimeout = 3600
    TCPHalfOpenLow = 400
    TCPHalfOpenMax = 500
    TCPSessionRateLow = 400
    TCPSessionRateMax = 500
    TCPSessionsMax = 65536
    TCPStrictnessLevel = 3
    UDPNotEstablishedTimeout = 30
    UDPEstablishedTimeout = 120
    UDPSessionsMax = 0
)

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

IKEProposal2 IKEV2-PROPOSAL (
    EncryptionAlg = ENCR_AES_CBC_256
    IntegrityAlg  = AUTH_HMAC_SHA2_256_128
    PrfAlg        = PRF_HMAC_SHA2_256
    KeyAlg        = MODP_2048
)

IPsecProposal2 IKEV2-IPSEC-PROPOSAL-CISCO (
    ESPEncryptionAlg = ENCR_AES_CBC_256
```

```
    ESPIntegrityAlg  = AUTH_HMAC_SHA2_256_128
)

IdentityEntry id:local:IKEV2-AUTH-RSA-SIG-LOCAL (
    DistinguishedName = USER_SPECIFIC_DATA
)

IKEAuthSignature2 local:IKEV2-AUTH-RSA-SIG-LOCAL (
    Identity  = id:local:IKEV2-AUTH-RSA-SIG-LOCAL
    Credential  = CertDescription()
    SendCertMode  = ALWAYS
)

IKEAuthSignature2 remote:IKEV2-AUTH-RSA-SIG-REMOTE (
    Identity  = IdentityEntry()
    Credential  = CertDescription()
    SendRequestMode  = ALWAYS
)

IPsecProposal2 IKEV2-IPSEC-PROPOSAL (
    ESPEncryptionAlg = ENCR_AES_CBC_256
    ESPIntegrityAlg  = AUTH_HMAC_SHA2_256_128
    PFSAlg = MODP_2048
)

IdentityEntry id:remote:IKEV2-AUTH-RSA-SIG-ANYCONNECT (
    DistinguishedName =
        CertDescription( Subject = TEMPLATE,"L=Zelenograd,O=S-Terra CSP,CN=AnyConnect"
)
)

IKEAuthSignature2 remote:IKEV2-AUTH-RSA-SIG-ANYCONNECT (
    Identity  = id:remote:IKEV2-AUTH-RSA-SIG-ANYCONNECT
    Credential  = CertDescription()
    SendRequestMode  = ALWAYS
)

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

AddressPool IKECFG_POOL_CLIENTS
(
    IPAddresses = 192.168.255.1..192.168.255.254
    DNSServers = 8.8.8.8
)

IKEParameters(
    FragmentSize = 0
)

IKERule2 IKERule2:VPN:1
(
```

```
    IKEPeerIPFilter = 10.0.108.195
    Proposals = IKEV2-PROPOSAL
    LocalAuth  = local:IKEV2-AUTH-RSA-SIG-LOCAL
    RemoteAuth  = remote:IKEV2-AUTH-RSA-SIG-REMOTE
    DPDMode             = PERIODIC
    DPDIdleDuration     = 5
    Priority            = 10
)

IKERule2 IKERule2:VPN:2
(
    IKEPeerIPFilter = 10.0.108.237
    Proposals = IKEV2-PROPOSAL
    LocalAuth  = local:IKEV2-AUTH-RSA-SIG-LOCAL
    RemoteAuth  = remote:IKEV2-AUTH-RSA-SIG-REMOTE
    DPDMode             = PERIODIC
    DPDIdleDuration     = 5
    Priority            = 20
)

IKERule2 IKERule2:VPN:3:DMAP:1
(
    Proposals = IKEV2-PROPOSAL
    LocalAuth  = local:IKEV2-AUTH-RSA-SIG-LOCAL
    RemoteAuth  = remote:IKEV2-AUTH-RSA-SIG-ANYCONNECT
    IKECFGPool          = IKECFG_POOL_CLIENTS
    DPDMode             = PERIODIC
    DPDIdleDuration     = 14
    Priority            = 100
)

IKERule2 IKERule2:VPN:3:DMAP:2
(
    Proposals = IKEV2-PROPOSAL
    LocalAuth  = local:IKEV2-AUTH-RSA-SIG-LOCAL
    RemoteAuth  = remote:IKEV2-AUTH-RSA-SIG-REMOTE
    IKECFGPool          = IKECFG_POOL_CLIENTS
    DPDMode             = PERIODIC
    DPDIdleDuration     = 14
    Priority            = 110
)

IPsecAction2 IPsecAction2:VPN:1
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 10.0.108.195
        Assemble = TRUE
    )
    ContainedProposals = IKEV2-IPSEC-PROPOSAL-CISCO
    DFHandling = COPY
    IKERule = IKERule2:VPN:1
)

IPsecAction2 IPsecAction2:VPN:2
(
    TunnelingParameters = TunnelEntry(
        PeerAddress = 10.0.108.237
        Assemble = TRUE
    )
    ContainedProposals = IKEV2-IPSEC-PROPOSAL
    DFHandling = COPY
    IKERule = IKERule2:VPN:2
```

```
)

IPsecAction2 IPsecAction2:VPN:3:DMAP:1
(
    TunnelingParameters = TunnelEntry(
        Assemble = TRUE
    )
    ContainedProposals = IKEV2-IPSEC-PROPOSAL-CISCO
    DFHandling = COPY
    IKERule = IKERule2:VPN:3:DMAP:1
)

IPsecAction2 IPsecAction2:VPN:3:DMAP:2
(
    TunnelingParameters = TunnelEntry(
        Assemble = TRUE
    )
    ContainedProposals = IKEV2-IPSEC-PROPOSAL
    DFHandling = COPY
    IKERule = IKERule2:VPN:3:DMAP:2
)

FilterChain IPsecPolicy:VPN (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.1.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction2:VPN:1 >
        LogEventID = "IPsec:Protect:VPN:1:IPSEC_ACL_HUB1_AND_CISCO"
    ),
    Filter (
        Label = "VPN:2:IPSEC_ACL_HUB1_AND_STRONGSWAN"
        SourceIP = 192.168.100.0/24
        DestinationIP = 192.168.2.0/24
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction2:VPN:2 >
        LogEventID = "IPsec:Protect:VPN:2:IPSEC_ACL_HUB1_AND_STRONGSWAN"
    ),
    Filter (
        Label = "VPN:3:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS"
        SourceIP = 0.0.0.0/0
        DestinationIP = 192.168.255.0/24
        Action = PASS
        ExtendedAction     =     ipsec<     sa     =     IPsecAction2:VPN:3:DMAP:1,
IPsecAction2:VPN:3:DMAP:2 >
        LogEventID                                                                =
"IPsec:Protect:VPN:3:DMAP:1:IPSEC_ACL_HUB1_AND_CLIENTS$VPN:3:DMAP:2:IPSEC_ACL_HUB1_AN
D_CLIENTS"
    )
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
    IPsecPolicy = IPsecPolicy:VPN
```

```
)

NetworkInterface (
    LogicalName = "GigabitEthernet0/1"
    InputFilter = IPv6Default
    OutputFilter = IPv6Default
)
```

Конфигурация УЦ на базе OpenSSL

1. Конфигурационный файл `openssl.cnf`:

```
# OpenSSL root CA configuration file.
# Copy to `/root/ca/openssl.cnf`.

[ ca ]
# `man ca`
default_ca = CA_default

[ CA_default ]
# Directory and file locations.
dir               = /root/ca
certs             = $dir/certs
crl_dir           = $dir/crl
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand

# The root key and root certificate.
private_key       = $dir/private/ca.key.pem
certificate       = $dir/certs/ca.cert.cer

# For certificate revocation lists.
crlnumber         = $dir/crlnumber
crl               = $dir/crl/ca.crl.pem
crl_extensions    = crl_ext
default_crl_days  = 30

# SHA-1 is deprecated, so use SHA-2 instead.
default_md        = sha256

name_opt          = ca_default
cert_opt          = ca_default
default_days      = 375
preserve          = no
policy            = policy_loose

# This extension is copied from the subjectAltName query
copy_extensions   = copy

[ policy_loose ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ req ]
# Options for the `req` tool (`man req`).
```

```
default_bits        = 2048
distinguished_name  = req_distinguished_name
string_mask         = utf8only

# SHA-1 is deprecated, so use SHA-2 instead.
default_md          = sha256

# Extension to add when the -x509 option is used.
x509_extensions     = v3_ca

[ req_distinguished_name ]
# See <https://en.wikipedia.org/wiki/Certificate_signing_request>.
commonName                      = Common Name
countryName                     = Country Name (2 letter code)
stateOrProvinceName             = State or Province Name
localityName                    = Locality Name
0.organizationName              = Organization Name
organizationalUnitName          = Organizational Unit Name
emailAddress                    = Email Address

# Optionally, specify some defaults.
countryName_default             =
stateOrProvinceName_default     =
localityName_default            =
0.organizationName_default      =
organizationalUnitName_default  =
emailAddress_default            =

[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints       = critical, CA:true
keyUsage               = critical, digitalSignature, cRLSign, keyCertSign

[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid,issuer:always
keyUsage                = digitalSignature
extendedKeyUsage        = clientAuth

[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid,issuer:always
keyUsage                = digitalSignature, keyEncipherment
extendedKeyUsage        = serverAuth

[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier = keyid:always
```

Конфигурация Cisco

1. Конфигурация cisco:

```
Building configuration...

Current configuration : 8543 bytes
```

```
!
! Last configuration change at 14:42:36 CST Mon Jul 15 2024
! NVRAM config last updated at 11:56:38 CST Mon Jul 15 2024
!
version 17.3
service timestamps debug datetime msec localtime
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Cisco
!
boot-start-marker
boot-end-marker
!
!
logging count
enable secret 9 $9$aqFbc0DFGNRSTU$W11vLCe.qvPPKJiwDPbV9ZcWxJqZ4zMeW8m0V4lWvcM
!
no aaa new-model
clock timezone CST 3 0
clock calendar-valid
!
!
!
!
!
!
!
ip domain name ikev2.cisco.com
!
!
!
login on-success log
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
```

```
!
!
!
!
crypto pki trustpoint CA
 enrollment terminal
 subject-name cn=Cisco-CSR100v, o=S-Terra CSP, l=Zelenograd, c=RU
 revocation-check none
 rsakeypair VPN
!
!
crypto pki certificate chain CA
 certificate 1001
  3082043F 30820327 A0030201 02020210 01300D06 092A8648 86F70D01 010B0500
  30713129 30270603 5504030C 20532D54 65727261 20435350 20546573 7420526F
  6F742043 412D4F70 656E5353 4C310B30 09060355 04061302 52553113 30110603
  5504070C 0A5A656C 656E6F67 72616431 14301206 0355040A 0C0B532D 54657272
  61204353 50310C30 0A060355 040B0C03 526E4430 1E170D32 34303731 35313132
  3333395A 170D3235 30373235 31313233 33395A30 50310B30 09060355 04061302
  52553113 30110603 55040713 0A5A656C 656E6F67 72616431 14301206 0355040A
  130B532D 54657272 61204353 50311630 14060355 0403130D 43697363 6F2D4353
  52313030 76308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100A481 4A93D319 5BFF6A7A BD87F6E0 98BAF55A D324D8FB 7BAB50C1
  62994688 1C1A59AF B1A39507 C3C2C7DD 22236549 8BCF3F36 98D6E081 28B4FE3B
  77E7F7D6 CAE170C9 CC220919 0945ECC7 FEDC5CEF 3F333C91 FA763FBE B7D6CFB2
  4D20AAF8 8DA08CAA 67B67AA5 F0457CCA 3BFB31DD 951A9E6F B42017C9 8DA82A58
  8A7B6416 E7679366 90DEDD57 E4A95A38 A15E0BE2 3202CF6D 145A7C38 288F6B8F
  11873542 0DF19F2A 89F80B83 0011D05D 228B7A24 9CC7AE59 A23F3F3D C7F98085
  48F2A4C9 F61293D0 DC7CAE10 BA90A980 5FC919BB 544631E8 91F2A729 A9EA2F85
  F6E5782D C29E4FA3 A8CEE8F3 5E696D8D 9457BE40 60356AB3 877227B0 6D382AAE
  CA07E776 98230203 010001A3 82010030 81FD3009 0603551D 13040230 00301D06
  03551D0E 04160414 AB27C52D 474E6AB6 8B84494E C61C1A37 3EAADE19 3081AE06
  03551D23 0481A630 81A38014 3C6E2257 B850A595 A50ECC95 78020449 7C5C28CD
  A175A473 30713129 30270603 5504030C 20532D54 65727261 20435350 20546573
  7420526F 6F742043 412D4F70 656E5353 4C310B30 09060355 04061302 52553113
  30110603 5504070C 0A5A656C 656E6F67 72616431 14301206 0355040A 0C0B532D
  54657272 61204353 50310C30 0A060355 040B0C03 526E4482 14770CFD 38EF6FE6
  9BB169CE 48FED6F4 50FF0136 A6300B06 03551D0F 04040302 05A03013 0603551D
  25040C30 0A06082B 06010505 07030130 0D06092A 864886F7 0D01010B 05000382
  01010033 1CFB438D 0EC657CB D844D32F 19688CC7 9F40F71F BB1DA1D8 7224CC49
  CAD9D051 C6EB4CED C95241FE 40BD3DA3 0DCB8436 F5585F49 78371DC6 86D9DCDC
  2855D2B7 03BE00F6 C577DE65 0DE7E391 B277E890 48AD6EA7 B392725F 9681EB7F
  8F224040 2402FE88 CC0B3601 243F591F FB375194 2978FAAF 2FF58902 FC0B00C8
  003BB900 21391164 2C4D6C8A A079D6A8 ACF56908 A0AC4FF0 06AD0482 AF1DA510
  1DA655A1 DF26036D 2AD5B7AB EC779780 730586E2 84BCE59F 5CD90EE0 5B6FB80A
  98577BF4 62AE6CD3 9CC195B0 97A0E3AB 667B6D4A 1213C88B 169B4301 DE06FA71
  C9453B7E 88FF2372 9B143C71 C8691391 0BA9B451 AA469DFA BABF333B 8FFECE17 0A6F04
        quit
 certificate ca 770CFD38EF6FE69BB169CE48FED6F450FF0136A6
  308203D3 308202BB A0030201 02021477 0CFD38EF 6FE69BB1 69CE48FE D6F450FF
  0136A630 0D06092A 864886F7 0D01010B 05003071 31293027 06035504 030C2053
  2D546572 72612043 53502054 65737420 526F6F74 2043412D 4F70656E 53534C31
  0B300906 03550406 13025255 31133011 06035504 070C0A5A 656C656E 6F677261
  64311430 12060355 040A0C0B 532D5465 72726120 43535031 0C300A06 0355040B
  0C03526E 44301E17 0D323430 32303731 34313935 325A170D 34343032 30323134
  31393532 5A307131 29302706 03550403 0C20532D 54657272 61204353 50205465
  73742052 6F6F7420 43412D4F 70656E53 534C310B 30090603 55040613 02525531
  13301106 03550407 0C0A5A65 6C656E6F 67726164 31143012 06035504 0A0C0B53
  2D546572 72612043 5350310C 300A0603 55040B0C 03526E44 30820122 300D0609
  2A864886 F70D0101 01050003 82010F00 3082010A 02820101 00D4C6EE EB50B70C
  6599D01F BEDA8462 34031670 68028974 F3CBD975 DC1AA806 0AABD1DD 340D6A04
  EFEE311F 16603C83 EF5309D6 BA99EAC1 509DA7F3 6C2A9BC1 AB1C8CD5 71473989
```

```
  73270ABD D5AAACC0 35EE8B7C 28A63222 E72095AD EE75AAE4 DFBD5158 B03AEFCE
  65A83462 08C605D9 971178D5 A430111B AD494D4D 28260445 3C7BF138 CA83FCE9
  9E718583 C76A973A 989821A9 54A1EF22 752F9C15 2AACC17E A50B8D3F 0403622F
  CDFDC634 0CD44CE5 F8B69A54 5E113A10 E47A568D 8D58D37C 78703B22 DE2EE55C
  EF9F05CC 0920F18F 84BB7A39 B8FD34D6 D86F737D 4EAA7A21 D05397B5 8176F504
  0F31AFC3 3F81BDB6 0206B247 5BC76D25 D4B74477 E0145CDF F5020301 0001A363
  3061301D 0603551D 0E041604 143C6E22 57B850A5 95A50ECC 95780204 497C5C28
  CD301F06 03551D23 04183016 80143C6E 2257B850 A595A50E CC957802 04497C5C
  28CD300F 0603551D 130101FF 04053003 0101FF30 0E060355 1D0F0101 FF040403
  02018630 0D06092A 864886F7 0D01010B 05000382 01010023 033B4790 66794AED
  0B674D6A 43F92EEF 1EC3F054 4CAEC5E3 90C8F323 22047DB7 9F50B447 991A5661
  216D5484 8993B419 D85F5CF3 F5758C19 27E94CB0 33A7C4A6 1FA2A1FA B379283D
  5C92445B 81909E48 61A1041F 69937C8C A925690E 5BB361E1 E5F14BE0 B6F3666A
  7FFFC7BA C3638879 73561483 2F46B9BB 3FF6EF47 F0D54754 22FFD9CF 2C22591D
  FB9BA50A 395E4D84 0CD88B44 B4C239B2 4675861E 3943EEB5 027EC7A6 A6E84232
  B42BFE42 2D9729ED A0461261 5B21122E 95970F20 DD94F164 51CAA31C 207D08A4
  87BF9D5A C205594C CB5355C9 58F45F15 5F168079 63C65318 D34C5A1E 3FA30DDB
  FA7EDF73 010D02EB B4C87263 6FFF81A7 78E3D1E6 AA5932
        quit
!
license udi pid CSR1000V sn 9VIGR7H198R
license boot level ax
diagnostic bootup level minimal
memory free low-watermark processor 71477
!
!
spanning-tree extend system-id
!
username           root           privilege           15           secret           9
$9$ipVV5e2Cmzew0.$CoIbG2KV02.FF1rVszZdaMRtjPnzFYVgVJ9liSMvOsk
!
redundancy
!
!
crypto ikev2 proposal 1
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy 1
 proposal 1
!
!
crypto ikev2 profile CISCO_ALGS
 description IKEv2 Win profile
 match identity remote any
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint CA
 dpd 10 5 periodic
!
crypto ikev2 fragmentation mtu 1024
!
!
!
!
!
!
!
!
!
!
```

```
!
!
crypto ipsec transform-set ENCRYPT_AND_INTEGRITY esp-aes 256 esp-sha256-hmac
 mode tunnel
!
!
!
crypto map VPN 1 ipsec-isakmp
 set peer 10.0.102.25
 set transform-set ENCRYPT_AND_INTEGRITY
 set ikev2-profile CISCO_ALGS
 match address IPSEC_ACL_CISCO_AND_HUB1
!
!
!
!
!
!
!
!
interface GigabitEthernet1
 ip address dhcp
 negotiation auto
 no mop enabled
 no mop sysid
 crypto map VPN
!
interface GigabitEthernet2
 ip address 192.168.1.1 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
ip ssh rsa keypair-name ssh-key
ip ssh version 2
ip scp server enable
!
ip access-list extended IPSEC_ACL_CISCO_AND_HUB1
 10 permit ip 192.168.1.0 0.0.0.255 192.168.100.0 0.0.0.255
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
line con 0
logging synchronous level 0
 stopbits 1
line vty 0 4
 login local
```

```
 transport input ssh
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact
email address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
!
!
!
!
!
end
```

Конфигурации StrongSwan

1. Содержание конфигурационного файла `/etc/ipsec.secrets` с указанием сертификата:

```
# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.
: RSA "strongSwanKey.pem"
```

2. Содержание конфигурационного файла `/etc/ipsec.conf`:

```
config setup
        charondebug="all"
        uniqueids=no
conn %default
        type=tunnel
        auto=route
        keyexchange=ikev2
        keyingtries=%forever
        ikelifetime=86400s
        lifetime=3600s
        dpddelay=30s
        dpdtimeout=120s
        dpdaction=restart
        fragmentation=yes
        authby=rsasig
        left=10.0.108.237
        leftsubnet=192.168.2.0/24
        leftcert=strongswan.cer
        ike=aes256-sha256-prfsha256-modp2048!
        esp=aes256-sha256-modp2048!
        mobike=no

conn StronSwan-Gate-Cert
        rightid=%any
        right=10.0.102.25
        rightsubnet=192.168.100.0/24
```

3. Содержание файла `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
```

```
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet dhcp

allow-hotplug ens34
iface ens34 inet static
address 192.168.2.1
netmask 255.255.255.0
```

Конфигурация AnyConnect

1. Содержание конфигурационного файла `C:\ProgramData\Cisco\ Cisco AnyConnect Secure Mobility Client\Profile\VPN.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<AnyConnectProfile                          xmlns="http://schemas.xmlsoap.org/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/encoding/ AnyConnectProfile.xsd">
      <ClientInitialization>
            <UseStartBeforeLogon UserControllable="true">true</UseStartBeforeLogon>
            <AutomaticCertSelection
UserControllable="false">true</AutomaticCertSelection>
            <ShowPreConnectMessage>false</ShowPreConnectMessage>
            <CertificateStore>All</CertificateStore>
            <CertificateStoreMac>All</CertificateStoreMac>
            <CertificateStoreLinux>All</CertificateStoreLinux>
            <CertificateStoreOverride>false</CertificateStoreOverride>
            <ProxySettings>Native</ProxySettings>
            <AllowLocalProxyConnections>true</AllowLocalProxyConnections>
            <AuthenticationTimeout>30</AuthenticationTimeout>
            <AutoConnectOnStart UserControllable="true">false</AutoConnectOnStart>
            <MinimizeOnConnect UserControllable="true">true</MinimizeOnConnect>
            <LocalLanAccess UserControllable="false">true</LocalLanAccess>
            <DisableCaptivePortalDetection
UserControllable="true">false</DisableCaptivePortalDetection>
            <ClearSmartcardPin UserControllable="false">true</ClearSmartcardPin>
            <IPProtocolSupport>IPv4</IPProtocolSupport>
            <AutoReconnect UserControllable="false">true
                  <AutoReconnectBehavior
UserControllable="false">ReconnectAfterResume</AutoReconnectBehavior>
            </AutoReconnect>
            <SuspendOnConnectedStandby>false</SuspendOnConnectedStandby>
            <AutoUpdate UserControllable="false">true</AutoUpdate>
            <RSASecurIDIntegration
UserControllable="false">Automatic</RSASecurIDIntegration>
            <WindowsLogonEnforcement>SingleLocalLogon</WindowsLogonEnforcement>
            <LinuxLogonEnforcement>SingleLocalLogon</LinuxLogonEnforcement>
            <WindowsVPNEstablishment>LocalUsersOnly</WindowsVPNEstablishment>
            <LinuxVPNEstablishment>LocalUsersOnly</LinuxVPNEstablishment>
            <AutomaticVPNPolicy>false</AutomaticVPNPolicy>
            <PPPExclusion UserControllable="false">Disable
                  <PPPExclusionServerIP
UserControllable="false"></PPPExclusionServerIP>
            </PPPExclusion>
            <EnableScripting UserControllable="false">false</EnableScripting>
            <EnableAutomaticServerSelection UserControllable="false">false

      <AutoServerSelectionImprovement>20</AutoServerSelectionImprovement>
```

```
      <AutoServerSelectionSuspendTime>4</AutoServerSelectionSuspendTime>
            </EnableAutomaticServerSelection>
            <RetainVpnOnLogoff>false
            </RetainVpnOnLogoff>

      <CaptivePortalRemediationBrowserFailover>false</CaptivePortalRemediationBrowse
rFailover>
            <AllowManualHostInput>true</AllowManualHostInput>
      </ClientInitialization>
      <ServerList>
            <HostEntry>
                  <HostName>IKEv2-AnyConnect</HostName>
                  <HostAddress>10.0.102.25</HostAddress>
                  <PrimaryProtocol>IPsec
                        <StandardAuthenticationOnly>true
                              <AuthMethodDuringIKENegotiation>IKE-
RSA</AuthMethodDuringIKENegotiation>
                        </StandardAuthenticationOnly>
                  </PrimaryProtocol>
            </HostEntry>
      </ServerList>
</AnyConnectProfile>
```

Настройки PKI (запросы и сертификаты) с использованием УЦ на базе MSCA

Генерация закрытого ключа и запроса на сертификат для криптошлюза.

1. Для этого запустите процесс генерации закрытого ключа и запроса на сертификат криптошлюзу с сохранением файла запроса:

Необходимо кроме дополнительного имени субъекта с внешнем IP адресом добавить расширение «Улучшенный ключ» со значением «Проверка подлинности сервера» в запрос на сертификат у УЦ. Данное расширение критично и без него защищенное соединение IKEv2/IPsec не будет построено. Это расширение добавляется путем указания его OID из набора OID (1.3.6..1.5.5.7.3.1). УЦ в последствии скопирует все добавленные поля из запроса в сам сертификат.

```
root@Hub1:~# cert_mgr create -subj "C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1-Windows-MSCA" -ip 10.0.102.25 -eku_oids 1.3.6..1.5.5.7.3.1 -RSA -2048 -fb64 Hub1.request
```

2. Выпуск сертификата на УЦ для криптошлюза и импорт сертификатов в базу Продукта.

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

   2.2. Скопируйте содержимое выпущенных сертификатов под именем `Hub1.cer` и сертификат УЦ под именем `ca.cer`.

   2.3. Импортируйте сертификат УЦ в базу Продукта:

```
root@Hub1:~#cert_mgr import -f ca.cer -t
```

- ключ `-f` задает месторасположение файла сертификата;
- ключ `-t` используется для импортирования доверенного (trusted) сертификата УЦ.

   2.4. Импортируйте сертификат криптошлюза в базу Продукта:

```
root@Hub1:~#cert_mgr import -f Hub1.cer -l
```

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

```
root@Hub1:~# cert_mgr show
```

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

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

   2.6. Выполните проверку статуса сертификатов в базе Продукта:

```
root@Hub1:~# cert_mgr check
```

```
1 State: Inactive C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD,CN=Hub1-Windows-MSCA
                  Certificate can not be verified.
2 State: Inactive CN=S-Terra CSP Test Root CA-OpenSSL,C=RU,L=Zelenograd,O=S-Terra CSP,OU=RnD
                  Certificate can not be verified.
```

Примечание – Видно, что все сертификаты имеют статус **Inactive** (неактивный) с пометкой: «Certificate can not be verified» (сертификат не может быть проверен). Причиной этому является включенный по умолчанию в консоли cisco-like механизм проверки списка отозванных сертификатов (далее COC или CRL). Так как в базе Продукта COC отсутствует, поэтому проверка не может быть осуществлена.

3. Генерация закрытого ключа и запроса на сертификат Windows.

3.1. На панели задач найдите поиск и введите «Управление сертификатами компьютеров». После появления в списке нужной утилиты нажмите кнопку «Открыть» (см. Рисунок 106):

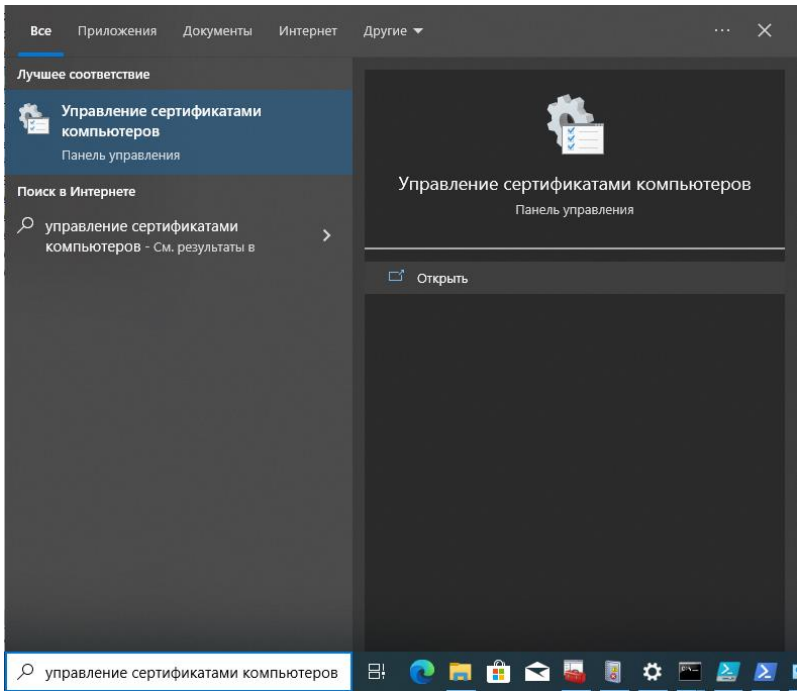


Рисунок 106

3.2. Разрешите этому приложению вносить изменения на вашем устройстве (см. Рисунок 107):

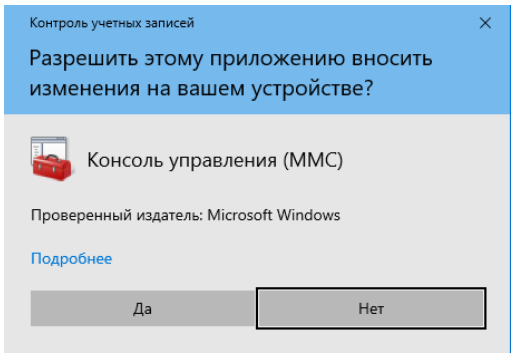


Рисунок 107

3.3. Перейдите в списке слева «Certificates – Local Computer» в пункт «Certificates – Local Computer\Personal\Certificates» (см. Рисунок 108)

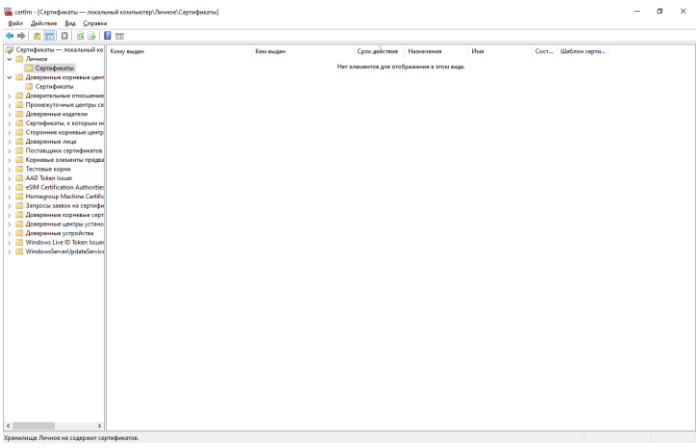


Рисунок 108

3.4. В пустом окне нажмите правой кнопкой мыши и выберите пункт «Все задачи», после выберите пункт «Дополнительные операции» и нажмите кнопку «Создать настраиваемый запрос…» (см. Рисунок 109):

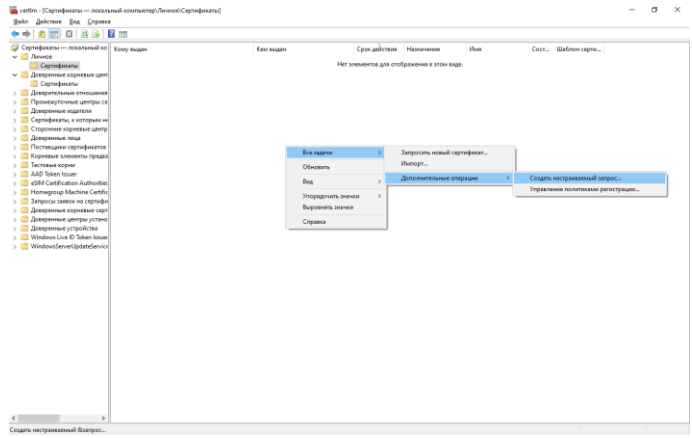


Рисунок 109

3.5. В окне «Регистрация сертификатов» нажмите кнопку «Далее» (см. Рисунок 110):

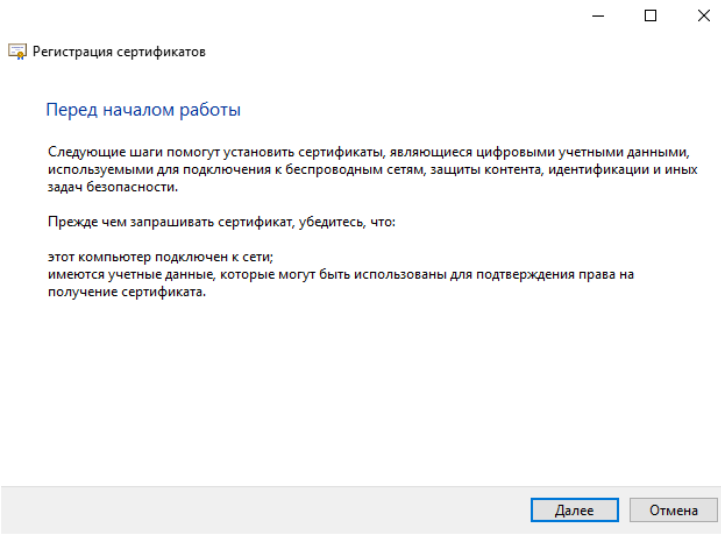


Рисунок 110

3.6. Нажмите еще раз кнопку «Далее», нет необходимости изменять/создавать политику регистрации сертификатов (см. Рисунок 111):

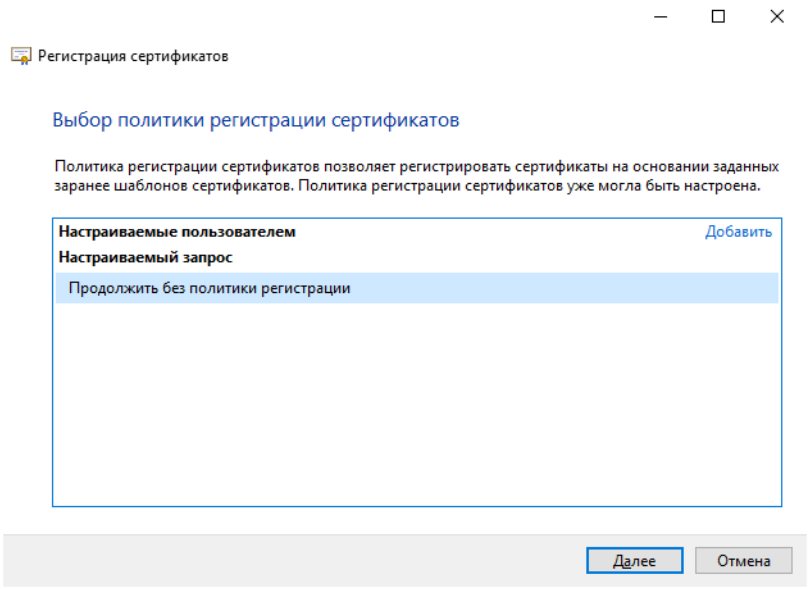


Рисунок 111

3.7. В окне «Пользовательский запрос» нажмите кнопку «Далее», здесь необходимо оставить выбор по умолчанию (см. Рисунок 112):

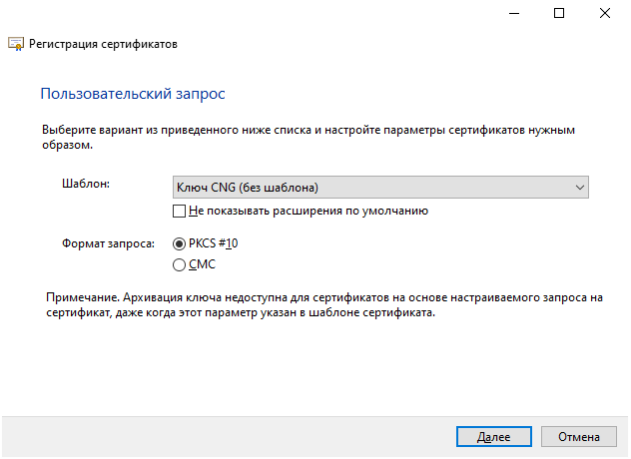


Рисунок 112

3.8. В окне «Сведения о сертификате» раскройте поле, для этого нажмите стрелочку. После нажмите кнопку «Свойства» для добавления необходимых полей для построения IKEv2/IPsec (см. Рисунок 113):

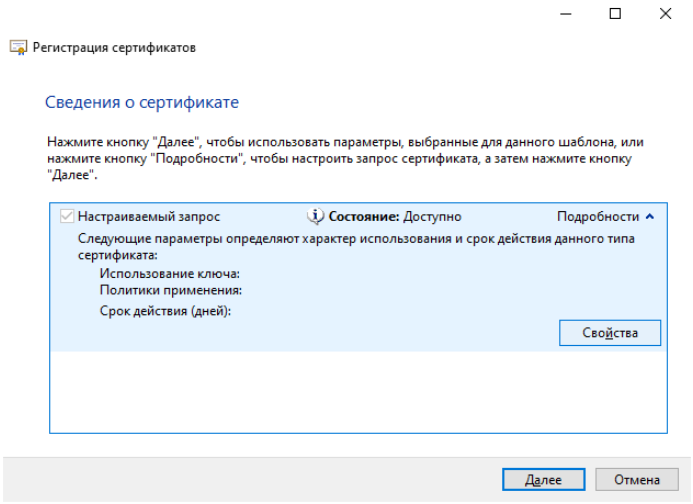


Рисунок 113

3.9. В окне свойства «Свойства сертификата» откройте вкладку «Субъект» (см. Рисунок 114):

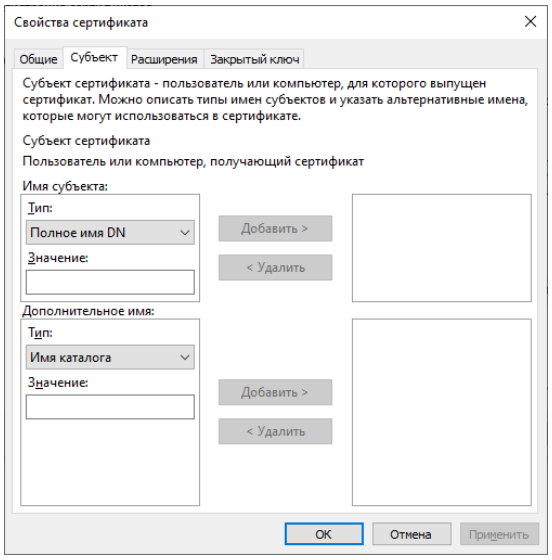


Рисунок 114

3.10. В этой вкладке необходимо добавить следующие типы и их значения в имя субъекта (см. Рисунок 115):

3.10.1. Добавьте тип «Общее имя» со значением «Windows».

3.10.2. Добавьте тип «Страна» со значением «RU».

3.10.3. Добавьте тип «Размещение» со значением «Zelenograd».

3.10.4. Добавьте тип «Организация» со значением «S-Terra CSP».

3.10.5. Добавьте тип «Подразделение» со значением «RnD».

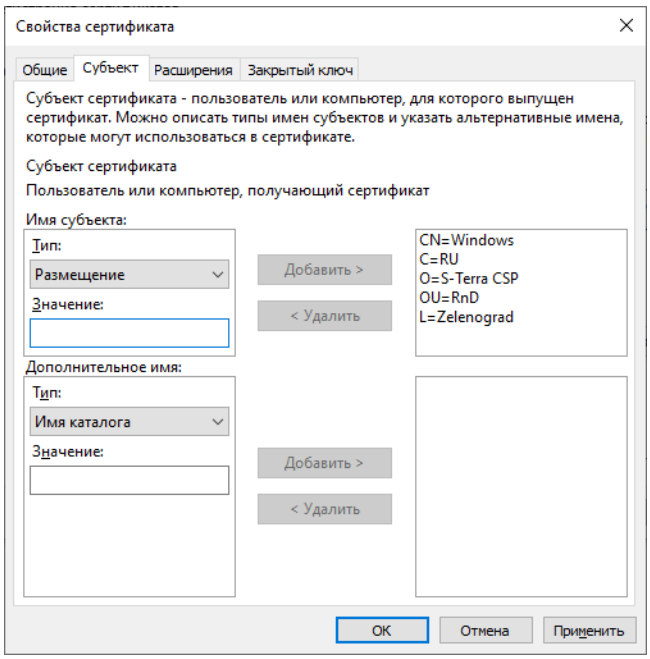


Рисунок 115

3.11. Откройте вкладку «Расширения» и раскройте вкладку «Использование ключа» (см. Рисунок 116):

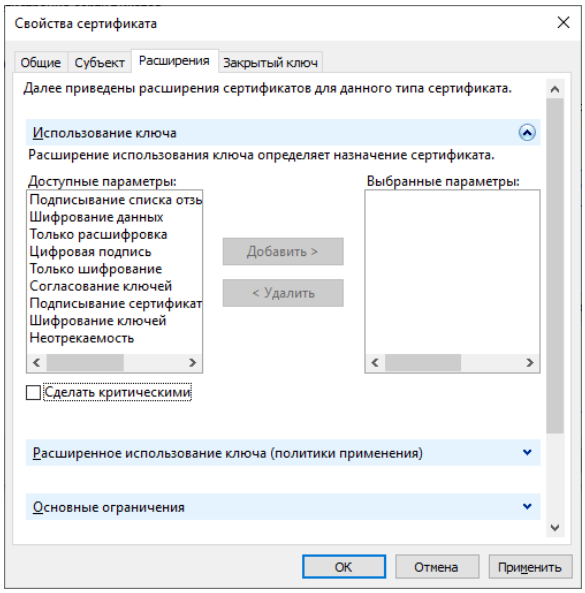


Рисунок 116

3.12. В этой вкладке необходимо убрать галку «Сделать критическими», а также из списка «Доступные параметры» выбрать расширение «Цифровая подпись» и нажмите кнопку «Добавить» (см. Рисунок 117):

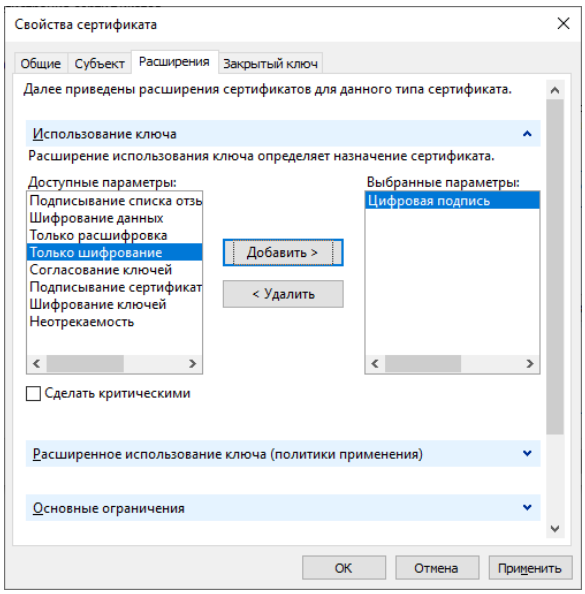


Рисунок 117

3.13. Закройте вкладку «Использование ключа» и откройте «Расширенное использование ключа (политики применения)». Из списка «Доступные параметры» выберите расширение «Проверка подлинности клиента» и нажмите кнопку «Добавить». После чего нажмите кнопку «Применить» и кнопку «ОК» (см. Рисунок 118).

Расширение «Основные ограничения» данным способом настроить корректно нет возможности, так как оно автоматически становится критическим расширением, что в свою очередь не позволяет построить защищенное соединение IKEv2/IPsec.

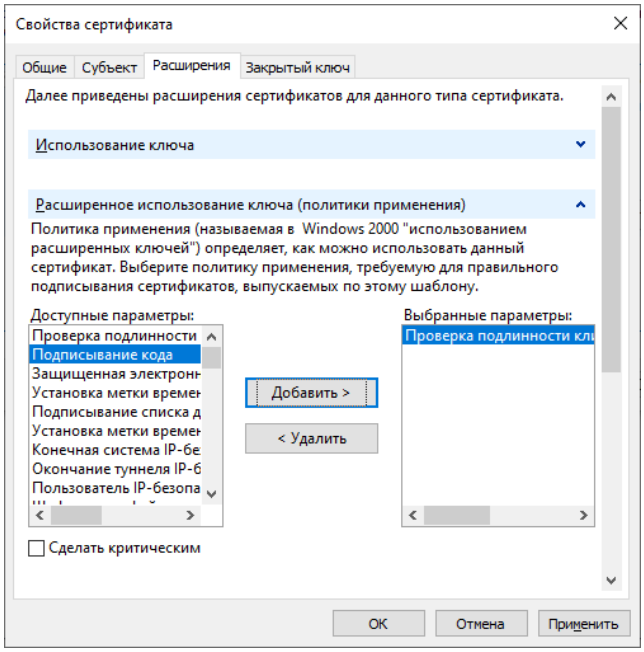


Рисунок 118

3.14. В окне «Сведения о сертификате» нажмите кнопку «Далее» (см. Рисунок 119):

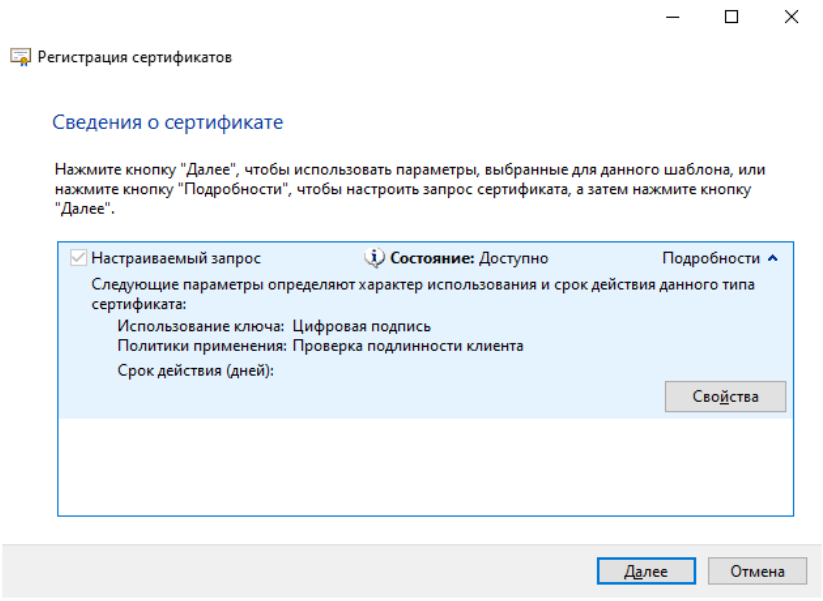


Рисунок 119

3.15. В окне «Где вы хотите сохранить автономный запрос?» укажите место хранения. Выбор формата файла оставьте без изменения и нажмите кнопку «Готово» (см. Рисунок 120):

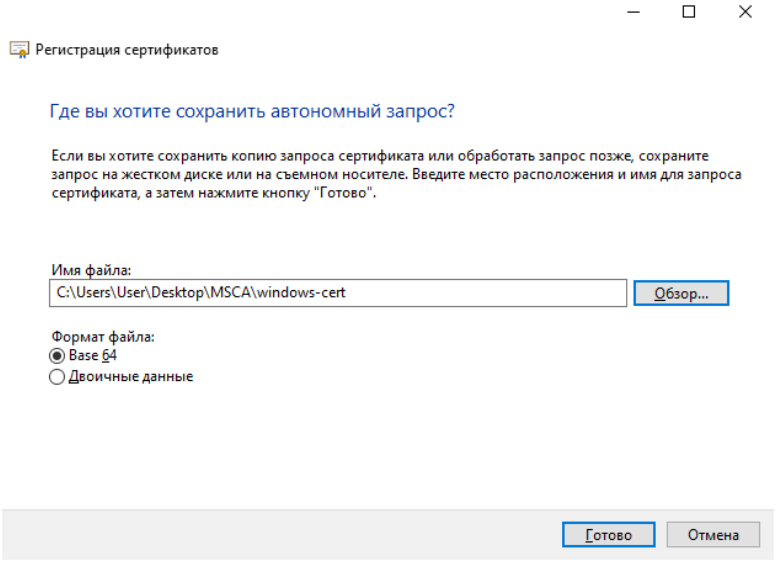


Рисунок 120

4. Выпустите сертификат Windows на УЦ. Импорт сертификатов производится аналогично разделу «Импорт сертификатов на Windows».

Поддержка аутентификации с использованием предварительно распределенных ключей (Preshared Key)

Криптошлюз поддерживает настройку аутентификации сторон с использованием предварительно распределенных ключей между партнерами: криптошлюзом, Cisco, StrongSwan и Android.

Между криптошлюзом и мобильным клиентом на базе ОС Windows нет возможности настроить работу аутентификации с использованием предварительно распределенных ключей. ОС Windows не поддерживает использование предварительно распределенных ключей при аутентификации. Поддержка использования предварительно распределенных ключей при аутентификации есть на ОС Windows Server.

Из-за особенностей работы клиента на базе ОС Windows Server не рекомендуется инициировать построение защищённого соединения между Hub1 и Windows Server с помощью ICMP-трафика, посылаемого с устройства Host-b-Hub1 на Windows Server. Это связано с тем, что после установки соединения могут возникнуть проблемы с прохождением трафика между защищёнными сетями, что приведёт к нарушению защищённого соединения.

Примечание – Настройка аутентификации сторон с использованием предварительно распределенных ключей не описывается в документе.

Настройка трассировки в Windows

1. Для трассировки в Windows необходимо запустить трассировочную сессию для анализа и сбора данных до инициализации Windows защищенного соединения IPsec.

```
C:\Windows\system32>logman start gary -ets -p Microsoft-Windows-RRAS -o ikev2.etl
Команда выполнена успешно.
```

2. Запустите подключение IPsec с IKEv2 на сертификатах.
3. Остановите трассировочную сессию командой:

```
C:\Windows\system32>logman stop gary -ets
Команда выполнена успешно.
```

4. Для вывода в терминал данных трассировочной сессии используйте команду:

```
C:\Windows\system32>wevtutil qe /lf /f:text ikev2.etl
```

Настройка логирования событий Windows

1. На панели задач откройте поиск и введите «Просмотр». После появления «Просмотр событий», нажмите кнопку «Открыть» (см. Рисунок 121):

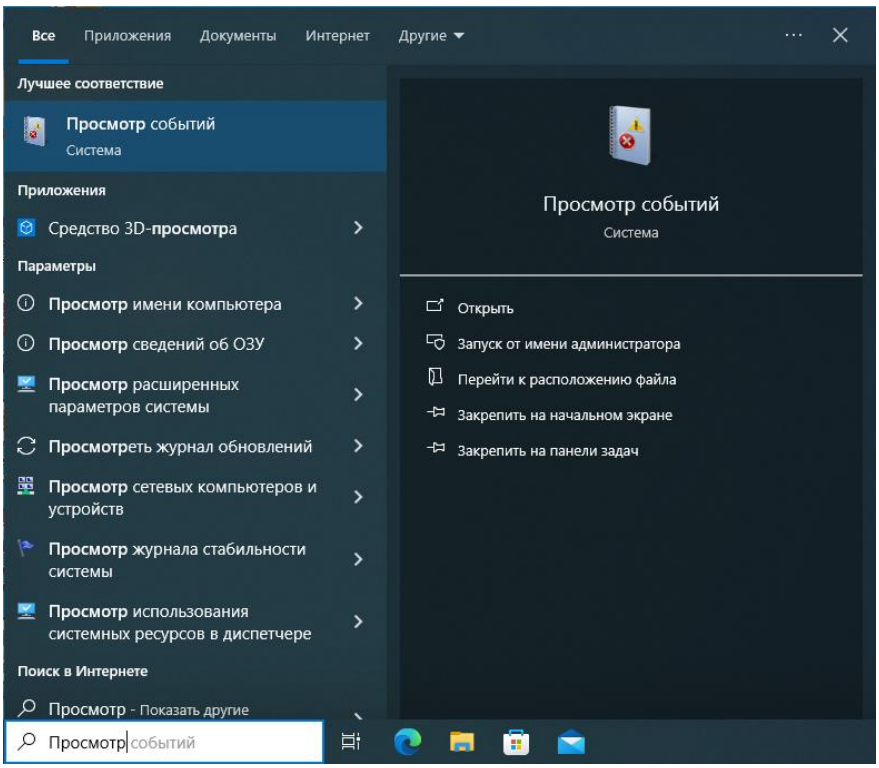


Рисунок 121

2. В открывшемся окне «Просмотр событий» перейдите во вкладку «Журналы Windows» и дождитесь пока прогрузятся журналы (см. Рисунок 122):

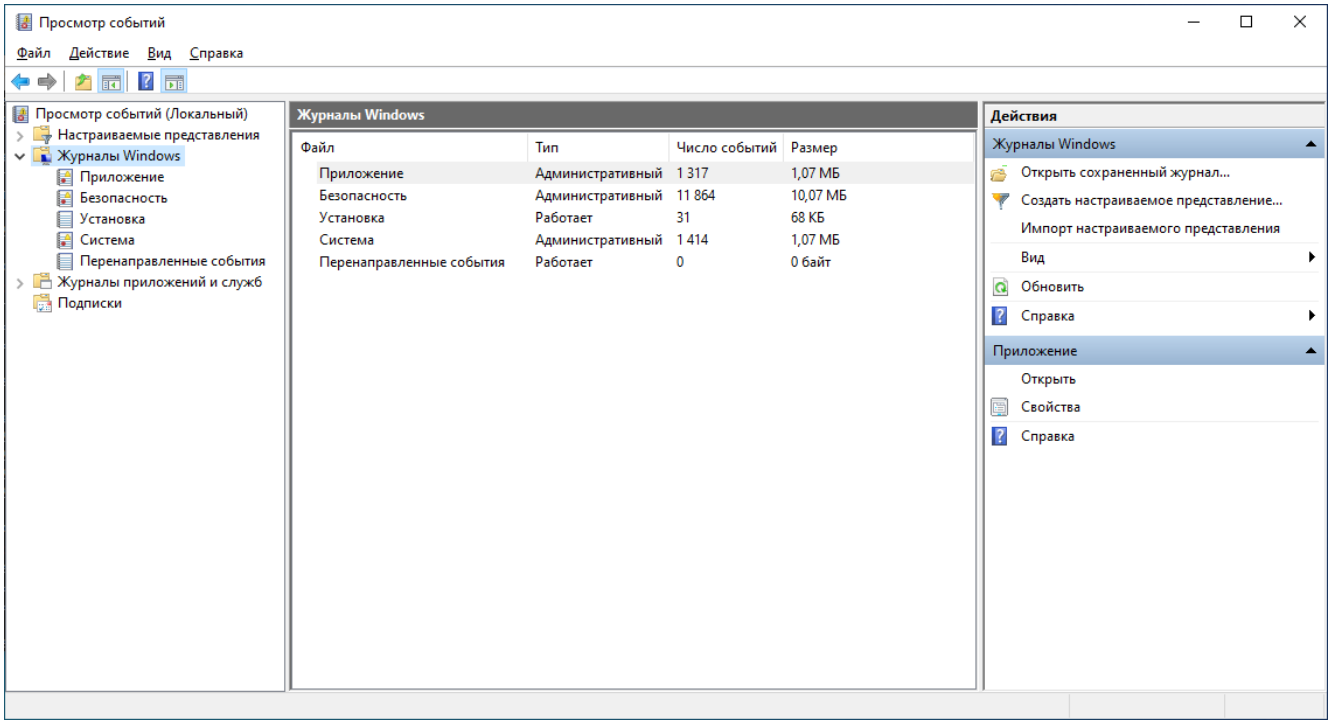


Рисунок 122

3. В списке журналов перейдите в «Приложение» (см. Рисунок 123):

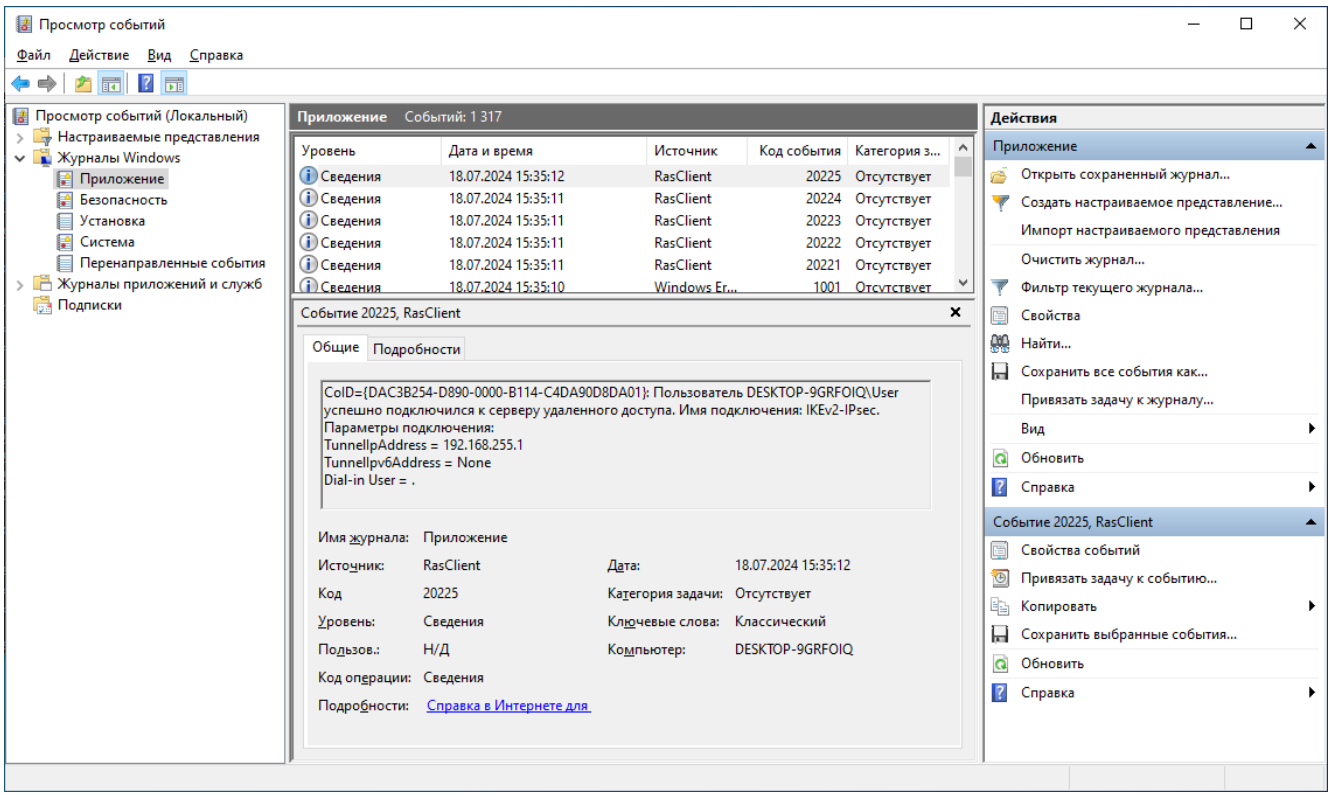


Рисунок 123

4. Отфильтруйте журнал, чтобы просматривать логи VPN-подключения Windows. Для этого нажмите во вкладке «Действия» кнопку «Фильтр текущего журнала…» (см Рисунок 123).

5. В открывшемся окне «Фильтровать текущий журнал» (см. Рисунок 124) выберите в выпадающем списке «Источники событий:» источник «RasClient» (см. Рисунок 125) и нажмите «ОК» (см. Рисунок 124):

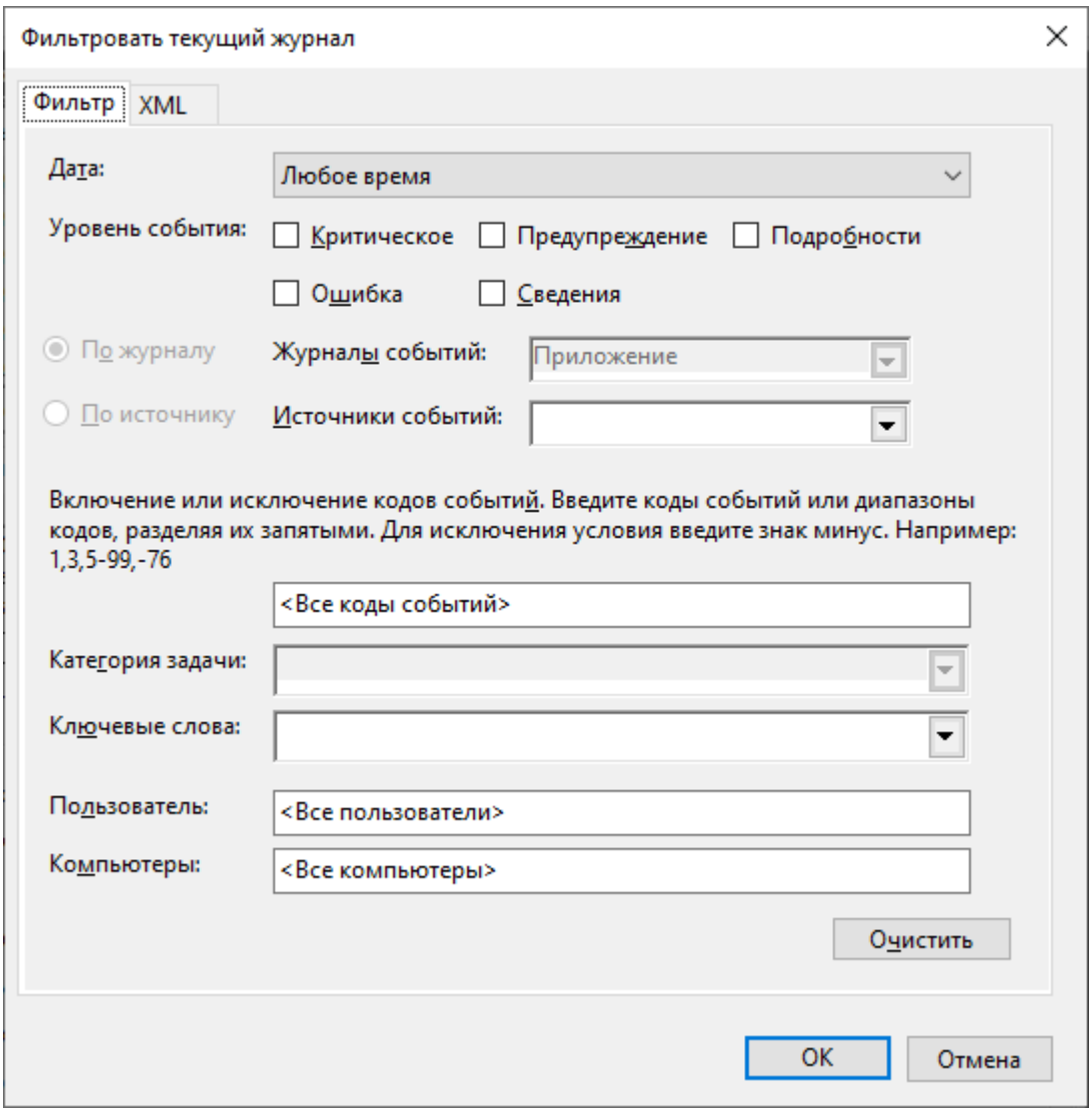

Фильтровать текущий журнал
Фильтр
XML
Дата:
Любое время
Уровень события:
Критическое
Предупреждение
Подробности
Ошибка
Сведения
По журналу
Журналы событий:
Приложение
По источнику
Источники событий:
Включение или исключение кодов событий. Введите коды событий или диапазоны кодов, разделяя их запятыми. Для исключения условия введите знак минус. Например: 1,3,5-99,-76
<Все коды событий>
Категория задачи:
Ключевые слова:
Пользователь:
<Все пользователи>
Компьютеры:
<Все компьютеры>
Очистить
OK
Отмена

Рисунок 124

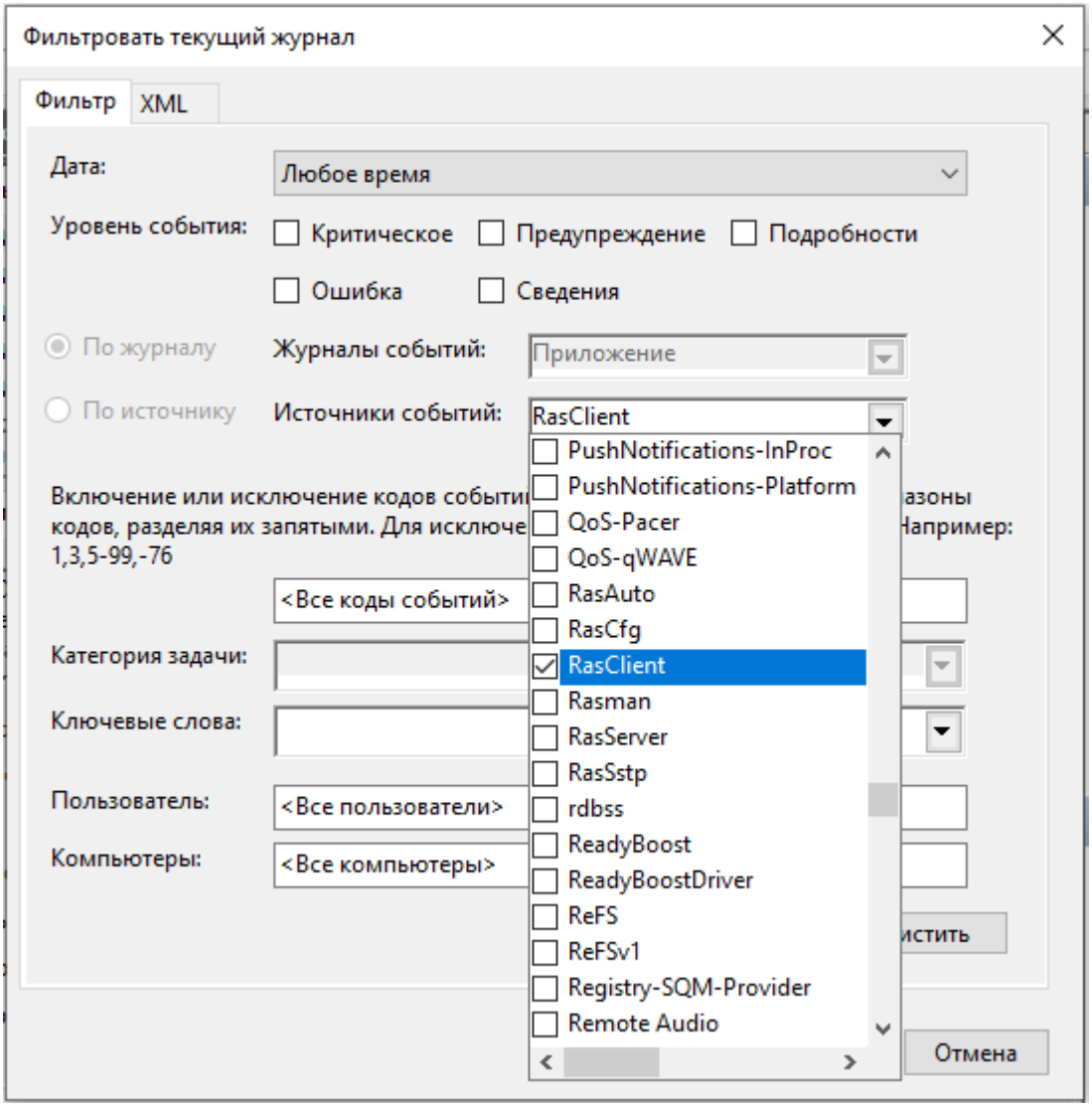


Рисунок 125

6. Просмотрите логи VPN-подключения Windows (см. Рисунок 126):

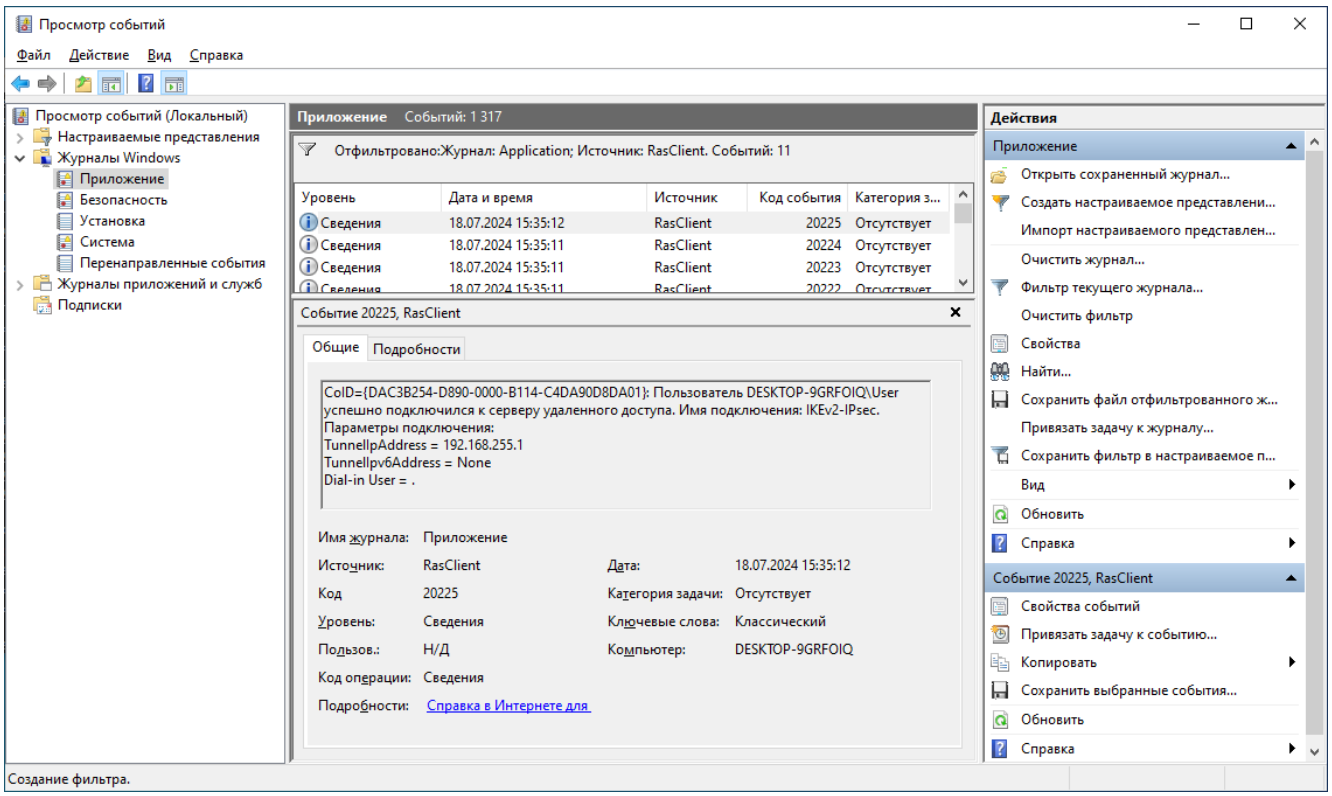

Просмотр событий
Файл Действие Вид Справка
Просмотр событий (Локальный)
Настраиваемые представления
Журналы Windows
Приложение
Безопасность
Установка
Система
Перенаправленные события
Журналы приложений и служб
Подписки
Приложение Событий: 1 317
Отфильтровано:Журнал: Application; Источник: RasClient. Событий: 11
Уровень Дата и время Источник Код события Категория з...
Сведения 18.07.2024 15:35:12 RasClient 20225 Отсутствует
Сведения 18.07.2024 15:35:11 RasClient 20224 Отсутствует
Сведения 18.07.2024 15:35:11 RasClient 20223 Отсутствует
Событие 20225, RasClient
Общие Подробности
CoID={DAC3B254-D890-0000-B114-C4DA90D8DA01}: Пользователь DESKTOP-9GRFOIQ\User успешно подключился к серверу удаленного доступа. Имя подключения: IKEv2-IPsec.
Параметры подключения:
TunnelIpAddress = 192.168.255.1
Tunnelpv6Address = None
Dial-in User = .
Имя журнала: Приложение
Источник: RasClient
Дата: 18.07.2024 15:35:12
Код 20225
Категория задачи: Отсутствует
Уровень: Сведения
Ключевые слова: Классический
Пользов.: Н/Д
Компьютер: DESKTOP-9GRFOIQ
Код операции: Сведения
Подробности: Справка в Интернете для
Действия
Приложение
Открыть сохраненный журнал...
Создать настраиваемое представлени...
Импорт настраиваемого представлен...
Очистить журнал...
Фильтр текущего журнала...
Очистить фильтр
Свойства
Найти...
Сохранить файл отфильтрованного ж...
Привязать задачу к журналу...
Сохранить фильтр в настраиваемое п...
Вид
Обновить
Справка
Событие 20225, RasClient
Свойства событий
Привязать задачу к событию...
Копировать
Сохранить выбранные события...
Обновить
Справка
Создание фильтра.

Рисунок 126

Настройка логирования событий Cisco

1. Задайте размер буфера логирования в 4096000 байт и уровень логирования 7 (debug) на Cisco в режим глобальной конфигурации:

```
Cisco(config)#logging count
Cisco(config)#logging buffered 4096000
Cisco(config)#logging buffered debugging
```

2. Вставьте максимальный уровень отладки по максимальному количеству каналов crypto ikev2 и ip packet на Cisco в привилегированный режим EXEC:

```
Cisco#debug crypto ikev2
IKEv2 default debugging is on
Cisco#debug crypto ikev2 error
IKEv2 error debugging is on
Cisco#debug crypto ikev2 internal
IKEv2 internal debugging is on
Cisco#debug crypto ikev2 packet
IKEv2 packet debugging is on
Cisco#debug crypto ikev2 packet hexdump
IKEv2 packet hexdump debugging is on
```

3. Очистите все IPsec соединения и буфер логирования, чтобы очистить буфер подтвердите, что вы хотите оичтить буфер на Cisco в привилегированный режим EXEC:

```
Cisco#clear crypto sa
Cisco#clear crypto sa counters
Cisco#clear crypto ikev2 sa
Cisco#clear crypto ikev2 stats
Cleared crypto ikev2 statistics
Cisco#clear crypto sa
Cisco#clear crypto sa counters
Cisco#clear logging
Clear logging buffer [confirm]y
```

4. Чтобы просмотреть буфер логирования, введите следующую команду на Cisco в привилегированный режим EXEC:

```
Cisco# show logging
Syslog logging: enabled (0 messages dropped, 3 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)
...
```

Настройка логирования событий StrongSwan

1. Проверьте, установлен ли пакет rsyslog:

```
root@StrongSwan:~#apt show rsyslog
```

```
Package: rsyslog
Version: 8.2302.0-1
Priority: optional
Section: admin
Maintainer: Michael Biebl <biebl@debian.org>
Installed-Size: 1,989 kB
Provides: linux-kernel-log-daemon, system-log-daemon
Depends: libc6 (>= 2.34), libestr0 (>= 0.1.9), libfastjson4 (>= 0.99.8), liblognorm5
(>= 2.0.3), libsystemd0 (>= 246), libuuid1 (>= 2.16), libzstd1 (>= 1.5.2), zlib1g (>=
1:1.1.4)
Recommends: logrotate
Suggests: rsyslog-mysql | rsyslog-pgsql, rsyslog-mongodb, rsyslog-doc, rsyslog-openssl
| rsyslog-gnutls, rsyslog-gssapi, rsyslog-relp
Conflicts: linux-kernel-log-daemon, system-log-daemon
Homepage: https://www.rsyslog.com/
Tag: admin::logging, implemented-in::c, interface::daemon, role::program,
 works-with::db, works-with::logfile
Download-Size: 723 kB
APT-Manual-Installed: yes
APT-Sources: http://deb.debian.org/debian bookworm/main amd64 Packages
Description: reliable system and kernel logging daemon
 Rsyslog is a multi-threaded implementation of syslogd (a system utility
 providing support for message logging), with features that include:
  * reliable syslog over TCP, SSL/TLS and RELP
  * on-demand disk buffering
  * email alerting
  * writing to MySQL or PostgreSQL databases (via separate output plugins)
  * permitted sender lists
  * filtering on any part of the syslog message
  * on-the-wire message compression
  * fine-grained output format control
  * failover to backup destinations
  * enterprise-class encrypted syslog relaying
 .
 It is the default syslogd on Debian systems.
```

2. Установите rsyslog, если до этого он не был установлен:

```
root@StrongSwan:~#apt-get install rsyslog
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libestr0 libfastjson4 liblognorm5
Suggested packages:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl | rsyslog-
gnutls rsyslog-gssapi rsyslog-relp
...
Processing triggers for libc-bin (2.36-9+deb12u7) ...
Processing triggers for man-db (2.11.2-2) ...
```

3. Отредактируйте файл `/etc/ipsec.conf`, который отвечает за настройку StronSwan для этого пропишите настройку конфигурации `charondebug="all"` в разделе `config setup`, которая записывает все отладочные сообщения уровня 7 в файл `/var/log/syslog`:

```
config setup
```

```
        charondebug="all"
...
```

4. Перезапустите StongSwan демон:

```
root@StrongSwan:~#ipsec restart
```

```
Stopping strongSwan IPsec...
Starting strongSwan 5.9.8 IPsec [starter]...
```

5. Чтобы посмотреть отладочные сообщения, откройте файл `/var/log/syslog`:

```
root@StrongSwan:~#less /var/log/syslog
```

Настройка логирования событий AnyConnect

1. На панели задач откройте поиск и введите «Просмотр». После появления «Просмотр событий», нажмите кнопку «Открыть» (см. Рисунок 127):

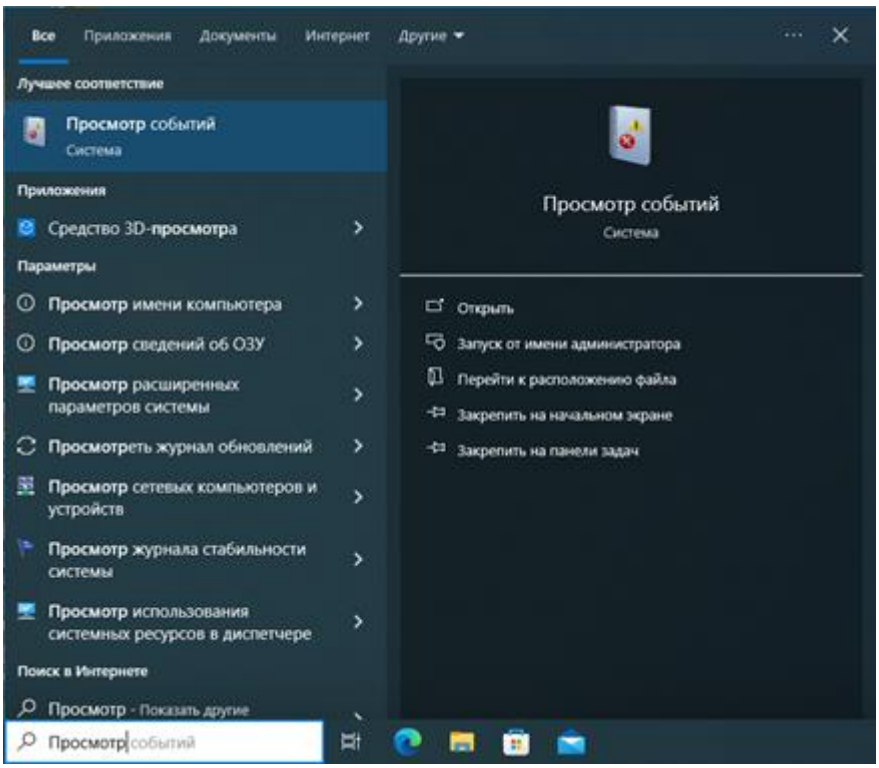


Рисунок 127

2. В открывшемся окне «Просмотр событий» перейдите во вкладку «Журналы приложенний и служб» и дождитесь пока прогрузятся журналы (см. Рисунок 128):

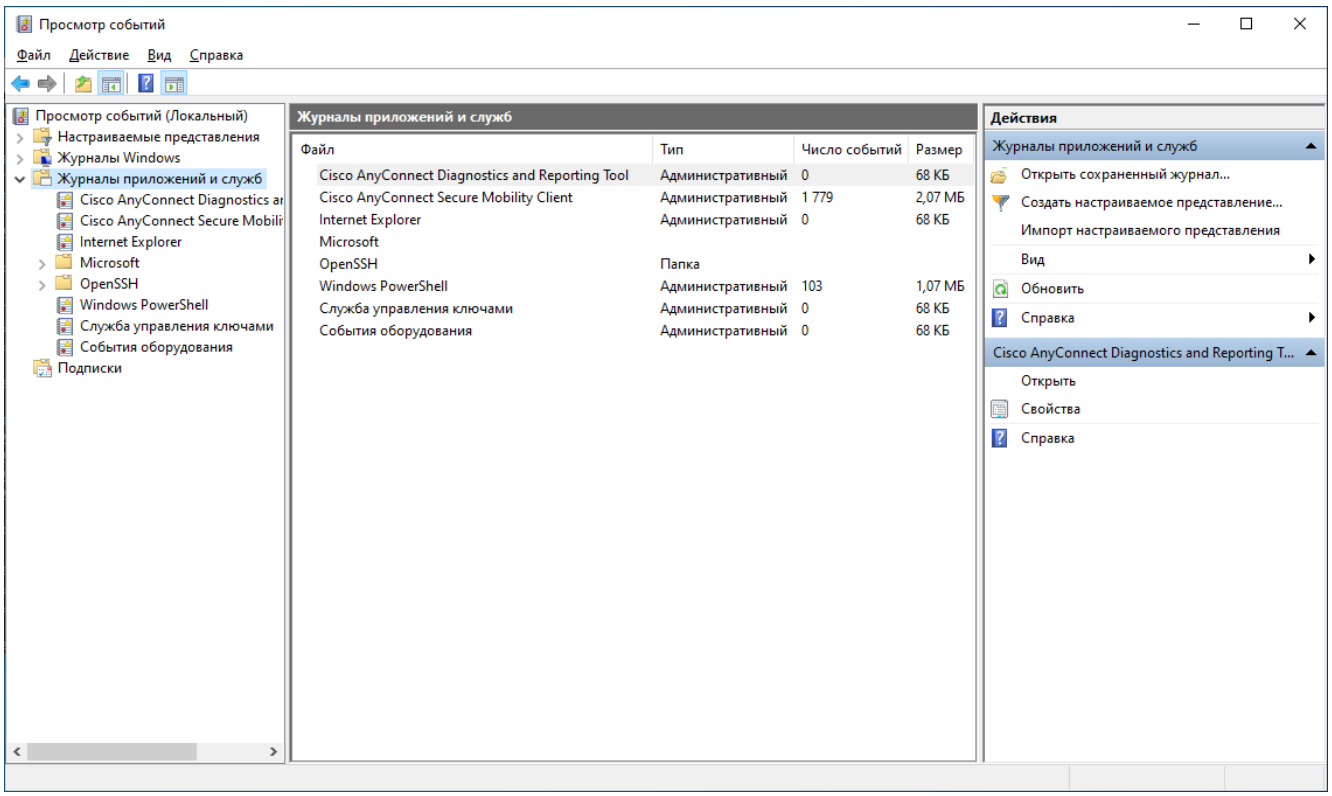


Рисунок 128

3. В списке журналов перейдите в «Cisco AnyConnect Secure Mobility Client» (см. Рисунок 128).
4. Просмотрите логи VPN-подключения AnyConnect (см. Рисунок 129):

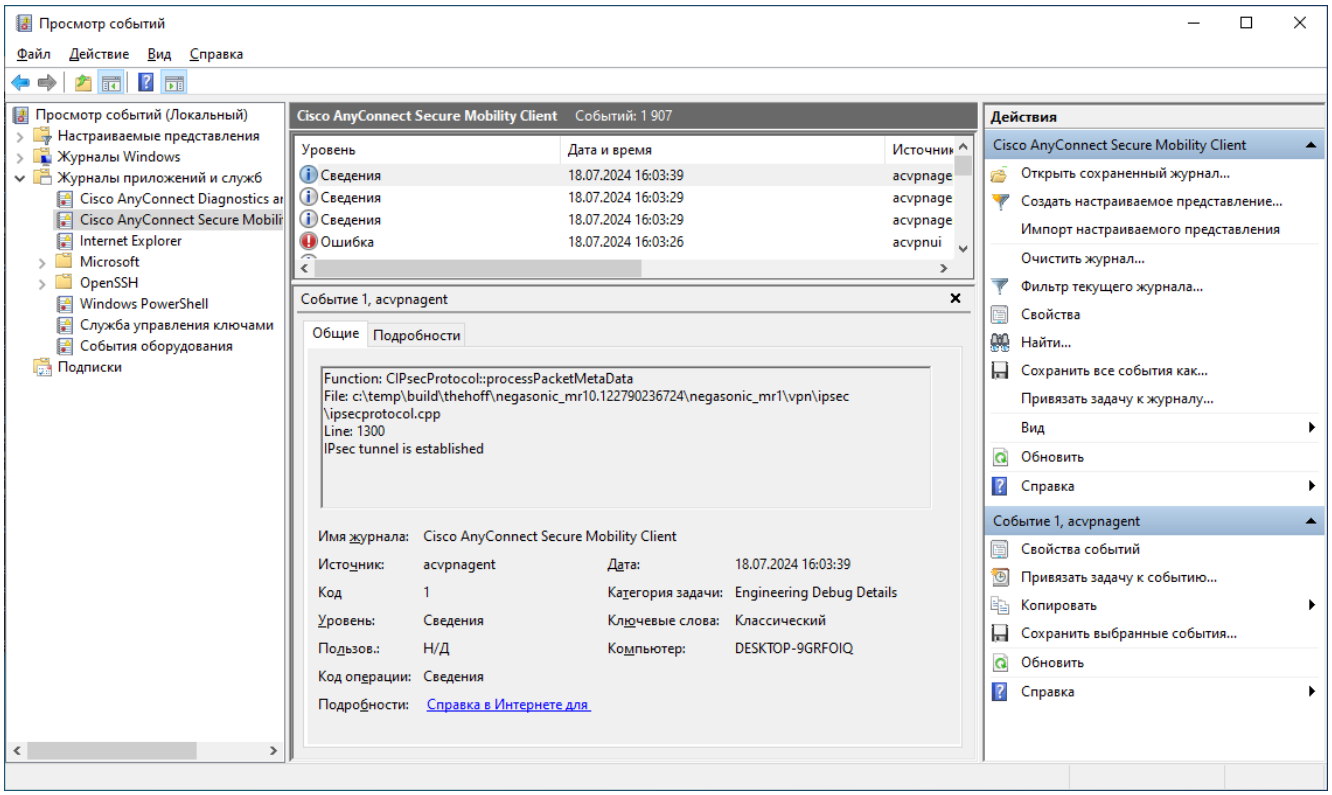


Рисунок 129