Настройка strongSwan IPsec VPN сервера и клиента

Описание стенда

Операционная система сервера = Ubuntu 20.04

Доменное имя сервера = server.astradomain.ad

Операционная система клиента = Ubuntu 20.04

Имя VPN клиента = user

Настройка сервера

Установка пакетов для работы

За основу инструкции по настройке сервера была взята следующая инструкция.

В первую очередь поставим все необходимые пакеты для работы:

Установка пакетов

```
sudo apt update
#    strongSwan VPN
sudo apt-get install strongswan strongswan-pki libcharon-extra-plugins libcharon-extauth-plugins

#     -   (       )
sudo apt install opensc libengine-pkcs11-openssl1.1
```

Генерация ключевых пар и сертификатов УЦ и сервера:

Создание ключевых пар и сертификатов УЦ и сервера

```
#
mkdir -p ~/pki/{cacerts,certs,private}
chmod 700 ~/pki

#
pki --gen --type rsa --size 4096 --outform pem > ~/pki/private/ca-key.pem
#
pki --self --ca --lifetime 3650 --in ~/pki/private/ca-key.pem \
    --type rsa --dn "CN=VPN root CA" --outform pem > ~/pki/cacerts/ca-cert.pem

#
pki --gen --type rsa --size 4096 --outform pem > ~/pki/private/server-key.pem
#
#  ,   --dn  --san
pki --pub --in ~/pki/private/server-key.pem --type rsa \
    | pki --issue --lifetime 1825 \
        --cacert ~/pki/cacerts/ca-cert.pem \
        --cakey ~/pki/private/ca-key.pem \
        --dn "CN=server.astradomomain.ad" --san server.astradomain.ad  \
        --flag serverAuth --flag ikeIntermediate --outform pem \
    >  ~/pki/certs/server-cert.pem

#      strongSwan
sudo cp -r ~/pki/* /etc/ipsec.d/
```

Настройка strongSwan

Сохраним предыдущую конфигурацию:

Резервная копия предыдущей конфигурации

```
sudo mv /etc/ipsec.conf{,.original}
```

Откроем файл **/etc/ipsec.conf** и вставим туда следующее содержимое:

/etc/ipsec.conf

```
config setup
    charondebug="ike 1, knl 1, cfg 0"
    uniqueids=no

conn ikev2-vpn
    auto=add
    compress=no
    type=tunnel
    keyexchange=ikev2
    fragmentation=yes
    forceencaps=yes
    dpdaction=clear
    dpddelay=300s
    rekey=no
    left=%any

#
    leftid=@server.astradomain.ad

    leftcert=server-cert.pem
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    right=%any
    rightid=%any

#
    rightauth=eap-tls

    rightsourceip=10.10.10.0/24
    rightdns=8.8.8.8,8.8.4.4
    rightsendcert=never
    eap_identity=%identity
    ike=chacha20poly1305-sha512-curve25519-prfsha512,aes256gcm16-sha384-prfsha384-ecp384,aes256-sha1-modp1024,
aes128-sha1-modp1024,3des-sha1-modp1024!
    esp=chacha20poly1305-sha512,aes256gcm16-ecp384,aes256-sha256,aes256-sha1,3des-sha1!
```

Укажем какой ключ использовать серверу для аутентификации себя клиенту **/etc/ipsec.secrets**:

/etc/ipsec.secrets

```
: RSA "server-key.pem"
```

Настройка firewall

Настройка firewall

```
sudo ufw allow OpenSSH
sudo ufw enable
sudo ufw allow 500,4500/udp
```

Узнаем имя интерфейса, к которому подключен сервер. Данное имя нам потребуется при настройке firewall. Это можно сделать с помощью команды:

Узнаем имя интерфейса

```
ip route show default
#      :
# default via your_server_ip dev eth0 proto static
#   -- eth0
```

Добавим в файл настроек firewall **/etc/ufw/before.rules** следующие строки:

/etc/ufw/before.rules

```
#
#
*nat
-A POSTROUTING -s 10.10.10.0/24 -o eth0 -m policy --pol ipsec --dir out -j ACCEPT
-A POSTROUTING -s 10.10.10.0/24 -o eth0 -j MASQUERADE
COMMIT

#
#
*mangle
-A FORWARD --match policy --pol ipsec --dir in -s 10.10.10.0/24 -o eth0 -p tcp -m tcp --tcp-flags SYN,RST SYN -
m tcpmss --mss 1361:1536 -j TCPMSS --set-mss 1360
COMMIT

#
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]

#
-A ufw-before-forward --match policy --pol ipsec --dir in --proto esp -s 10.10.10.0/24 -j ACCEPT
-A ufw-before-forward --match policy --pol ipsec --dir out --proto esp -d 10.10.10.0/24 -j ACCEPT
```

Добавим в файл **/etc/ufw/sysctl.conf** следующие строки:

/etc/ufw/sysctl.conf

```
net/ipv4/ip_forward=1
net/ipv4/conf/all/accept_redirects=0
net/ipv4/conf/all/send_redirects=0
net/ipv4/ip_no_pmtu_disc=1
```

Перезагрузка firewall

Перезагрузка firewall

```
sudo ufw disable
sudo ufw enable
```

Запуск VPN сервера

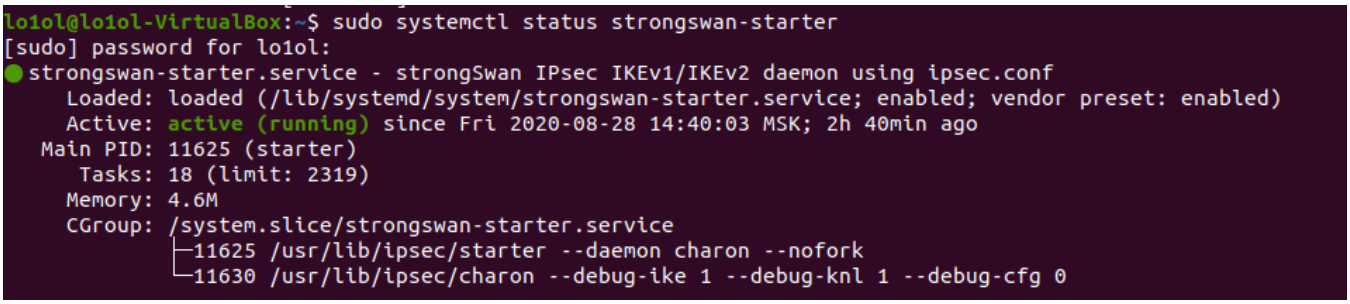
Запуск сервера

```
sudo systemctl restart strongswan-starter
```

Проверить, что сервер поднялся успешно, можно, например, с помощью команды:

Проверка запуска сервера

```
sudo systemctl status strongswan-starter
```



Добавление нового клиента со смарт-картой

Модуль pkcs11 для работы со смарт-картами

Необходимо использовать модуль opensc-pkcs11.so из состава OpenSC.

Отформатируем смарт-карту, сгенерируем на ней ключи и получим сертификат.

Подготовка смарт-карты

```
#
pkcs15-init --erase-card -p rutoken_ecp
pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk ""
pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678" --puk "" --so-pin "87654321" --finalize

#
ID=45
pkcs15-init -G rsa/2048 --auth-id 02 --label "My Private Key" --public-key-label "My Public Key" --id $ID

#
openssl
> engine dynamic -pre SO_PATH:/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1 -
pre LOAD -pre MODULE_PATH:opensc-pkcs11.so
#
> req -engine pkcs11 -new -key 45 -keyform engine -out req.csr -subj "/C=RU/CN=user"
> exit

#
pki --issue --lifetime 1825 \
        --cacert ~/pki/cacerts/ca-cert.pem \
        --cakey ~/pki/private/ca-key.pem \
        --in req.csr --type pkcs10 --outform pem \
    >  ~/pki/certs/client-cert.pem

#
pkcs15-init --store-certificate ~/pki/certs/client-cert.pem --auth-id 02 --id $ID --format pem
```

Убедиться, что сертификат и ключи загружены на токен можно с помощью команды:

проверка

```
pkcs11-tool -O -l
```

Теперь Рутокен готов к работе и его можно отдать клиенту.

Настройка клиента

Проверка доступности сервера

Проверим, что сервер доступен.

проверка доступности сервера

```
ping server.astradomain.ad
```

Установка пакетов

Установим необходимые пакеты для работы:

Установка пакетов

```
sudo apt update
#    strongSwan VPN
sudo apt-get install strongswan libstrongswan-extra-plugins  libcharon-extra-plugins

#     - (       )
sudo apt install opensc libengine-pkcs11-openssl1.1
```

Настройка strongSwan

Корневой сертификат сервера положим в директорию **/etc/ipsec.d/cacerts**:

Установка корневого сертификата

```
sudo cp /path/to/ca-cert.pem /etc/ipsec.d/cacerts
```

Настроим файл конфигурации strongSwan **/etc/ipsec.conf**:

/etc/ipsec.conf

```
config setup

conn ikev2-rw
#
    right=server.astradomain.ad
    rightid=@server.astradomain.ad

    rightsubnet=0.0.0.0/0
    rightauth=pubkey
    leftsourceip=%config
#
    leftid=user
    leftcert=%smartcard:45
    leftauth=eap
    eap_identity=%identity
    auto=start
```

Настроим файл паролей аутентфикации strongSwan **/etc/ipsec.secrets** и укажем, какой ключ нужно использовать для аутентификации по смарт-карте:

/etc/ipsec.secrets

```
#
#: PIN %smartcard:<keyid> <pin code>
: PIN %smartcard:45 "12345678"
```

Более подробно о способах задания паролей смарт-карт можно почитать здесь.

Настройка модуля pkcs11

Настроим использование pkcs11 модулей в strongSwan. Для этого откроем файл конфигурации **/etc/strongswan.d/charon/pkcs11.conf** и отредактируем настройки модулей pkcs11:

/etc/strongswan.d/charon/pkcs11.conf

```
modules {

        opensc-pkcs11 {

            # Whether to automatically load certificates from tokens.
            # load_certs = yes

            # Whether OS locking should be enabled for this module.
            # os_locking = no

            # Full path to the shared object file of this PKCS#11 module.
            path = /usr/lib/x86_64-linux-gnu/opensc-pkcs11.so

        }
```

Подключение к сети

Подключите смарт-карту и инициализируйте подключение с помощью команды:

Подключение к сети VPN

```
sudo systemctl stop strongswan-starter
sudo systemctl start strongswan-starter
```

Проверить, что соединение прошло успешно, можно, например, выведя список своих ip адресов С помощью команды:

Проверка успешности подключения

```
ip addr show
```

Среди них появится ваш виртуальный ip:

