



ОНЛАЙН-ОБРАЗОВАНИЕ

Домашнее задание: VPN

1. Между двумя виртуалками поднять vpn в режимах:

- tun

- tap

Описать в чём разница, замерить скорость между виртуальными машинами в туннелях, сделать вывод об отличающихся показателях скорости.

2. Поднять RAS на базе OpenVPN с клиентскими сертификатами, подключиться с локальной машины на виртуалку.

3 (*). Самостоятельно изучить, поднять ocserv и подключиться с хоста к виртуалке.

1. TUN/TAP режимы VPN

1. Для выполнения первого пункта необходимо написать Vagrantfile, который будет поднимать 2 виртуальные машины server и client.

Типовой Vagrantfile для данной задачи:

```
# -*- mode: ruby -*-
```

```
# vi: set ft=ruby :
```

```
Vagrant.configure(2) do |config|
  config.vm.box = "centos/7"
  config.vm.define "server" do |server|
    server.vm.hostname = "server.loc"
    server.vm.network "private_network", ip: "192.168.10.10"
  end
  config.vm.define "client" do |client|
    client.vm.hostname = "client.loc"
    client.vm.network "private_network", ip: "192.168.10.20"
  end
end
```

1. TUN/TAP режимы VPN

2. После запуска машин из Vagrantfile заходим на VM server и выполняем следующие действия на server и client машинах:

- устанавливаем epel репозиторий:

```
yum install -y epel-release
```

- устанавливаем пакет openvpn, easy-rsa и iperf3

```
yum install -y openvpn iperf3
```

- Отключаем SELinux (при желании можно написать правило для openvpn)

```
setenforce 0 (работает до ребута)
```

3. Настройка openvpn сервера:

- создаём файл-ключ

```
openvpn --genkey --secret /etc/openvpn/static.key
```

- создаём конфигурационный файл vpn-сервера

```
vi /etc/openvpn/server.conf
```

1. TUN/TAP режимы VPN

Файл `server.conf` должен содержать следующий конфиг.

```
dev tap
ifconfig 10.10.10.1 255.255.255.0
topology subnet
secret /etc/openvpn/static.key
comp-lzo
status /var/log/openvpn-status.log
log /var/log/openvpn.log
verb 3
```

- Запускаем `openvpn` сервер и добавляем в автозагрузку.

```
systemctl start openvpn@server
systemctl enable openvpn@server
```

1. TUN/TAP режимы VPN

4. Настройка openvpn клиента.

- создаём конфигурационный файл клиента

```
vi /etc/openvpn/server.conf
```

- Файл должен содержать следующий конфиг

```
dev tap
```

```
remote 192.168.10.10
```

```
ifconfig 10.10.10.2 255.255.255.0
```

```
topology subnet
```

```
route 192.168.10.0 255.255.255.0
```

```
secret /etc/openvpn/static.key
```

```
comp-lzo
```

```
status /var/log/openvpn-status.log
```

```
log /var/log/openvpn.log
```

```
verb 3
```

1. TUN/TAP режимы VPN

- На сервер клиента в директорию `/etc/openvpn/` необходимо скопировать файл-ключ `static.key`, который был создан на сервере.
 - Запускаем `openvpn` клиент и добавляем в автозагрузку

```
systemctl start openvpn@server
systemctl enable openvpn@server
```
5. Далее необходимо замерить скорость в туннеле.
- на `openvpn` сервере запускаем `iperf3` в режиме сервера

```
iperf3 -s &
```
 - на `openvpn` клиенте запускаем `iperf3` в режиме клиента и замеряем скорость в туннеле

```
iperf3 -c 10.10.10.1 -t 40 -i 5
```
6. Повторяем пункты 1-5 для режима работы `tun`. Конфигурационные файлы сервера и клиента изменятся только в директиве `dev`. Делаем выводы о режимах, их достоинствах и недостатках.

2. RAS на базе OpenVPN

Для выполнения данного задания можно воспользоваться Vagrantfile из 1 задания, только убрать 1 VM. После запуска VM отключаем SELinux (`setenforce 0`) или создаём правило для него.

1. Устанавливаем репозиторий EPEL.

```
yum install -y epel-release
```

2. Устанавливаем необходимые пакеты.

```
yum install -y openvpn easy-rsa
```

3. Переходим в директорию `/etc/openvpn/` и инициализируем pki

- `cd /etc/openvpn/`
- `/usr/share/easy-rsa/3.0.3/easyrsa init-pki`

4. Сгенерируем необходимые ключи и сертификаты для сервера

- `echo 'rasvpn' | /usr/share/easy-rsa/3.0.3/easyrsa build-ca nopass`
- `echo 'rasvpn' | /usr/share/easy-rsa/3.0.3/easyrsa gen-req server nopass`
- `echo 'yes' | /usr/share/easy-rsa/3.0.3/easyrsa sign-req server server`
- `/usr/share/easy-rsa/3.0.3/easyrsa gen-dh`
- `openvpn --genkey --secret ta.key`

2. RAS на базе OpenVPN

5. Сгенерируем сертификаты для клиента.

- `echo 'client' | /usr/share/easy-rsa/3/easyrsa gen-req client nopass`
- `echo 'yes' | /usr/share/easy-rsa/3/easyrsa sign-req client client`

6. Создадим конфигурационный файл `/etc/openvpn/server.conf` (файл конфигурации показан на слайде #10)

7. Зададим параметр `iroute` для клиента

```
echo 'iroute 192.168.33.0 255.255.255.0' > /etc/openvpn/client/client
```

8. Запускаем `openvpn` сервер и добавляем в автозагрузку.

```
systemctl start openvpn@server
```

9. `systemctl enable openvpn@server`

10. Скопируем следующие файлы сертификатов и ключ для клиента на хост-машину.

```
/etc/openvpn/pki/ca.crt
```

```
/etc/openvpn/pki/issued/client.crt
```

```
/etc/openvpn/pki/private/client.key
```

(файлы рекомендуется расположить в той же директории, что и `client.conf`)

11. Создадим конфигурационный файл клиента `client.conf` на хост-машине (файл конфигурации показан на слайде #11).

2. RAS на базе OpenVPN

Файл конфигурации `server.conf` >

```
port 1207
proto udp
dev tun
ca /etc/openvpn/pki/ca.crt
cert /etc/openvpn/pki/issued/server.crt
key /etc/openvpn/pki/private/server.key
dh /etc/openvpn/pki/dh.pem
server 10.10.10.0 255.255.255.0
route 192.168.10.0 255.255.255.0
push "route 192.168.10.0 255.255.255.0"
ifconfig-pool-persist ipp.txt
client-to-client
client-config-dir /etc/openvpn/client
keepalive 10 120
comp-lzo
persist-key
persist-tun
status /var/log/openvpn-status.log
log /var/log/openvpn.log
verb 3
```

2. RAS на базе OpenVPN

Файл конфигурации `client.conf` >

```
dev tun
proto udp
remote 192.168.10.10 1207
client
resolv-retry infinite
ca ./ca.crt
cert ./client.crt
key ./client.key
route 192.168.10.0 255.255.255.0
persist-key
persist-tun
comp-lzo
verb 3
```

В этом конфигурационном файле указано, что файлы сертификатов располагаются в директории, где располагается `client.conf`. Но при желании можно разместить сертификаты в других директориях и в конфиге скорректировать пути.

2. RAS на базе OpenVPN

12. После того, как все готово, подключаемся к openvpn сервер с хост-машины.

```
openvpn --config client.conf
```

13. При успешном подключении проверяем пинг в внутреннему IP адресу сервера в туннеле.

```
ping -c 4 10.10.10.1
```

14. Также проверяем командой `ip r (netstat -rn)` на хостовой машине, что сеть туннеля импортирована в таблицу маршрутизации.

3. (*) OpenConnect сервер

3 (*). Самостоятельно изучить, поднять ocserv и подключиться с хоста к виртуалке.

Материал по данному заданию необходимо самостоятельно изучить и поднять OpenConnect сервер и подключиться к нему с хост-машины.