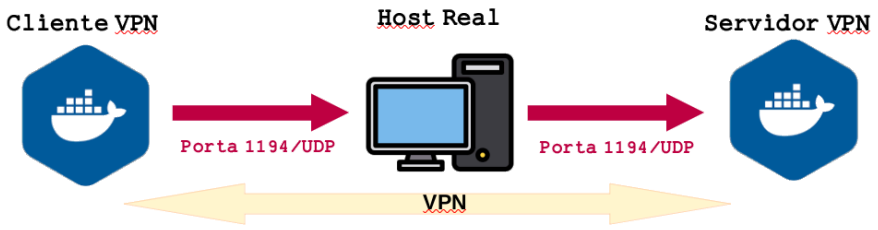
Instituto Federal de Educação, Ciência e Tecnologia do RN
Câmpus Currais Novos
Curso: Tecnologia em Sistemas para Internet
Disciplina: Segurança e Preservação de Dados

Atividade Prática em Laboratório (Roteiro) - VPN com OpenVPN

Utilizando contêineres docker vamos montar o seguinte cenário:



1) Crie um contêiner docker usando uma imagem leve (Debian 12 slim). Usaremos o parâmetro **`--privileged`** para garantir que a interface **`tun`** (necessária para o OpenVPN) possa ser criada e mapearmos a porta 1194/UDP do host real para a porta 1194/UDP do contêiner onde estará executando o servidor OpenVPN:

```
$ docker run -it --privileged -p 1194:1194/udp --name openvpn_server debian:12-slim /bin/bash
```

(`--privileged` é para fins de laboratório; em produção, usa-se `--cap-add=NET_ADMIN` e `--device=/dev/net/tun`)

2) Atualize o APT e instale os pacotes necessários para a prática:

```
# apt update
# apt install -y openvpn easy-rsa iproute2 nano
```

O OpenVPN utiliza certificados para autenticação.
Usaremos o **`easy-rsa`** para criar a Autoridade Certificadora (CA) e os certificados do servidor e do cliente.

3) Copie os scripts do `easy-rsa` para um diretório de trabalho seguro e inicialize o PKI.:

```
# cp -r /usr/share/easy-rsa/ /etc/openvpn/
# cd /etc/openvpn/easy-rsa/
# ./easyrsa init-pki
```

4) Crie a Autoridade Certificadora (CA) que assinará todos os outros certificados:

```
# ./easyrsa build-ca nopass
```

Quando a execução do script solicitar o "`Common Name`", use um nome como "`OpenVPN-CA`".

5) Crie o Certificado e Chave do Servidor, gere a solicitação de certificado (req) e assine-a com a CA:

```
# ./easyrsa gen-req server nopass
```

Quando a execução do script solicitar o "`Common Name`", use um nome como "`server`".

```
# ./easyrsa sign-req server server
```

Quando a execução do script solicitar digite '`yes`' para confirmar a assinatura.

6) Gere os parâmetros Diffie-Hellman (DH), criando o o arquivo DH (crucial para a troca segura de chaves):

```
# ./easyrsa gen-dh
```

7) Gere chaves TLS-Auth (HMAC) adicionais para proteção contra ataques de Negação de Serviço (DoS) e varredura de portas:

```
# openvpn --genkey secret ta.key
```

8) Copie todos os arquivos gerados para o diretório de configuração do OpenVPN:

```
# cp pki/ca.crt pki/dh.pem pki/private/server.key pki/issued/server.crt ta.key /etc/openvpn/
```

Criação e Configuração do Servidor OpenVPN

9) Crie e edite o arquivo de configuração principal do Servidor OpenVPN (`server.conf`):

```
# nano /etc/openvpn/server.conf
```

Conteúdo do arquivo **`server.conf`**:

```
# Configurações de Conexão
port 1194
proto udp
dev tun

# Configurações da PKI e Criptografia
ca ca.crt
cert server.crt
key server.key
dh dh.pem
tls-auth ta.key 0

# Configurações de Rede
server 10.8.0.0 255.255.255.0 # Rede interna da VPN
push "redirect-gateway def1 bypass-dhcp" # Redireciona todo o tráfego do cliente
push "dhcp-option DNS 8.8.8.8" # DNS para os clientes

# Otimização e Segurança
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
compress lz4-v2

# Log
verb 3
```

10) Habilite o encaminhamento de pacotes (IP Forwarding) para que o tráfego dos clientes VPN possa sair do contêiner para a internet (ou rede externa do host):

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

11) Inicie o Serviço OpenVPN em background:

```
# cd /etc/openvpn
# openvpn --config /etc/openvpn/server.conf &
```

Verifique a nova interface com o comando '`ip a`' (deve aparecer tun0).

Criação e Configuração dos arquivos de um Cliente OpenVPN

12) Volte ao diretório `easy-rsa` e crie um certificado para o cliente:

```
# cd /etc/openvpn/easy-rsa/
# ./easyrsa gen-req client1 nopass
(Quando solicitado o "Common Name", use "client1")

# ./easyrsa sign-req client client1
(Digite 'yes' para confirmar a assinatura)
```

O cliente precisa de um arquivo que combine suas chaves, o certificado da CA e as configurações de conexão.

13) No Host principal (fora do docker) crie um diretório e o arquivo de configuração do Cliente (.ovpn):

```
$ mkdir client1_vpn
$ cd client1_vpn
```

14) Copie os arquivos necessários do Contêiner para o Host:

```
$ docker cp openvpn_server:/etc/openvpn/ca.crt .
```

```
$ docker cp openvpn_server:/etc/openvpn/ta.key .
$ docker cp openvpn_server:/etc/openvpn/easy-rsa/pki/issued/client1.crt .
$ docker cp openvpn_server:/etc/openvpn/easy-rsa/pki/private/client1.key .
```

15) Crie e edite o arquivo de configuração do cliente no Host:

```
# nano client1.ovpn
```

Conteúdo do arquivo **`client1.ovpn`**:

```
client
dev tun
proto udp
remote Endereço_IP_do_Servidor 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client1.crt
key client1.key
remote-cert-tls server
tls-auth ta.key 1
compress lz4-v2
verb 3
```

Criando um Contêiner Docker para usar como Cliente OpenVPN

16) Crie um contêiner docker usando novamente uma imagem leve (Debian 12 slim) agora para funcionar como cliente VPN:

```
$ docker run -it --privileged --name openvpn_client1 debian:12-slim /bin/bash
```

17) Atualize o APT e instale os pacotes necessários para a prática:

```
# apt update
# apt install -y openvpn iproute2 nano
```

18) Crie uma pasta para colocar os arquivos de configuração para acessar a VPN:

```
# cd
# mkdir acessar_vpn
```

19) No Host principal (fora do docker) dentro do diretório 'client1_vpn' copie os arquivos para o novo docker:

```
$ cd ~/client1_vpn
$ docker cp * openvpn_client1:/root/acessar_vpn
```

No contêiner `client1_vpn` altere o arquivo `client1.ovpn` trocando o campo `Endereço_IP_do_Servidor` pelo IP do Host principal (que está mapeando a porta 1194/UDP para o Servidor VPN).

20) Finalmente no contêiner `openvpn_client1` conecte ao contêiner `openvpn_server`:

```
# cd ~/acessar_vpn
# openvpn --config client1.ovpn
```

O contêiner cliente1 está agora conectado ao contêiner server via VPN