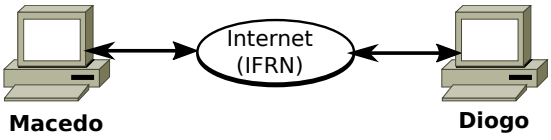
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Professor: Macêdo Firmino
Disciplina: Segurança de Rede
Prática 14: VPN com OpenVPN.

Olá, meus alunos!! Como é que vocês vão? Na aula de hoje iremos aprender sobre a VPN. Iremos instalar o servidor OpenVPN em uma máquina Ubuntu, vamos utilizar uma chave (tls-crypt) específica para cada cliente e aprendermos também a instalação e configuração do OpenVPN no cliente Ubuntu. Usaremos como base o tutorial disponibilizado pelo site `https://simplificandoredes.com/instalar-open-vpn-em-linux/`. Vamos lá!!! Preparados???

Configurando o Ambiente

Para estudarmos estes conceitos e ferramenta iremos utilizar duas máquinas virtuais. Uma para transmitir o arquivo (Diogo) e outra para verificar a integridade (Macedo).



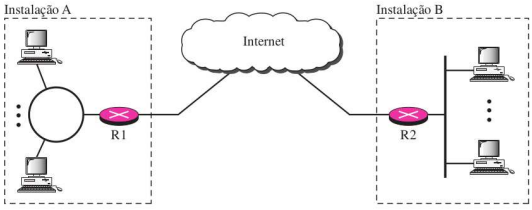
VPNs

Uma rede privada é desenvolvida para uso interno em uma organização. Ela possibilita o acesso a recursos compartilhados e, ao mesmo tempo, fornece privacidade. Uma organização pequena com uma única sede pode usar uma LAN isolada garantindo privacidade ao transferir informações.

Uma organização maior, com várias sedes, pode criar uma internet privada. As LANs em locais diferentes podem ser interligadas por meio de roteadores e linhas alugadas. Nessa situação, a organização cria uma internet privada que esteja completamente isolada da Internet global.

Entretanto, as redes privadas apresentam como desvantagem os custos de instalação e manutenção. Para interligar várias instalações em locais diferentes, uma organização precisaria de várias linhas alugadas, implicando alto aluguel mensal. Uma solução seria usar a Internet global tanto para comunicação privada como pública.

Uma tecnologia denominada rede privada virtual (VPN) possibilita que organizações usem a Internet global para enviar mensagens seguras entre diferentes prédios da organização. A VPN cria uma rede que é privada, mas virtual. É privada, pois garante sigilo dentro da organização. E é virtual, porque não usa WANs privadas reais; a rede é fisicamente pública, embora virtualmente privada.



OpenVPN

O OpenVPN é um software livre e open-source para criar redes virtuais privadas do tipo ponto-a-ponto ou server-to-multiclient através de túneis criptografados entre computadores. O OpenVPN permite autenticação ponto-a-ponto através de chaves secretas compartilhadas, certificados digitais ou autenticação com usuário e senha.

Ele utiliza a arquitetura cliente/servidor e os protocolos SSLv3/TLSv1. O OpenVPN pode ser executado em transportes do User Datagram Protocol (UDP) ou do Transmission Control Protocol (TCP), multiplexando túneis SSL.

O OpenVPN está disponível em duas versões: OpenVPN Community Edition (versão gratuita e de código aberto) e a OpenVPN Access Server (paga com mais recursos). Iremos utilizar a versão gratuita.

Configurando o Servidor

Instalação

Para instalação, execute os seguintes comandos a seguir:

```
sudo apt-get install openvpn
sudo apt-get install easy-rsa
```

Normalmente, o OpenVPN já vem instalado por padrão nas distribuições Linux.

Criação de Diretórios

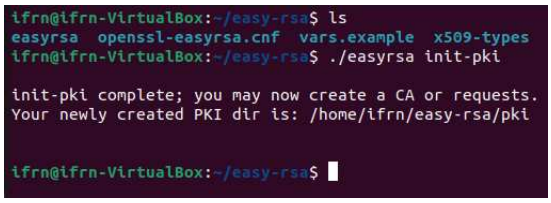
Inicialmente iremos criar um diretório para o easy-rsa e liga-lo através de links simbólicos. Usaremos links simbólicos para que futuras atualizações possam ser replicadas para nossas configurações.

```
mkdir ~/easy-rsa
ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

Irá surgir quatro links: easyrsa (executável), openssl-easyrsa.cnf (arquivo de configuração), vars.example (exemplo de arquivo de configuração), x509-types (diretório).

Antes de criarmos a chave privada e certificado do seu servidor OpenVPN, você precisa criar um diretório local da infraestrutura de chaves públicas. A PKI em seu servidor VPN é usado apenas como um lugar conveniente e centralizado para armazenar solicitações de certificado e certificados públicos. Na pasta $\sim /easy-rsa/$ execute o comando:

```
./easyrsa init-pki
```



Irá surgir a pasta pki com arquivos de configuração (openssl-easyrsa.cnf e safessl-easyrsa.cnf) e as pastas private e reqs (ambas vazias).

Após inicializar sua PKI no servidor OpenVPN, você está pronto para continuar para o próximo passo, que é a criação de uma Autoridade Certificadora.

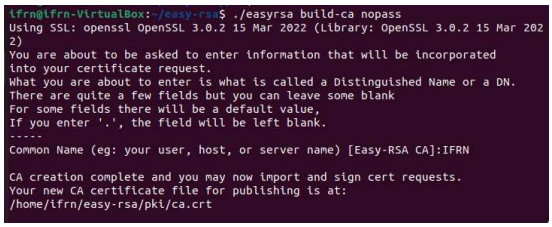
Criando a Autoridade Certificadora

Na aula, estamos criando uma CA no mesmo servidor VPN por uma questão de simplicidade e praticidade. No entanto, o ideal é que a CA esteja em um outro servidor separado.

Ao criar a CA teremos a opção de preencher alguns campos. O campo principal é o common name. O nosso campo common name terá o valor "IFRN". Além disso vamos usar a opção "nopass"para evitar que tenhamos que usar password quando formos assinar um certificado.

Para criar a CA, pasta $\sim /easy-rsa/$ execute o comando:

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



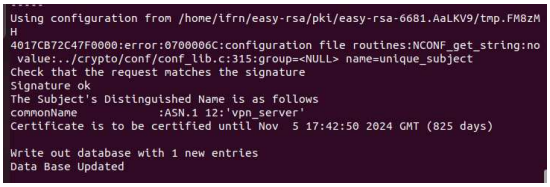
Na pasta pki será inserido os arquivos de configuração, os certificados, chave privada e as demais informações da autoridade certificadora.

Criando as Chaves do Servidor OpenVPN

Agora iremos criar as chaves do servidor. Embora já tenha executado este comando no servidor CA como parte dos pré-requisitos, é necessário executá-lo aqui. Isso se dá, pois seu servidor OpenVPN e servidor CA possuem diretórios da PKI separados e chaves distintas.

Para criarmos a chave do servidor OpenVPN, vamos usar os comandos na pasta $\sim /easy-rsa/$:

```
./easyrsa build-server-full vpn_server
nopass
```

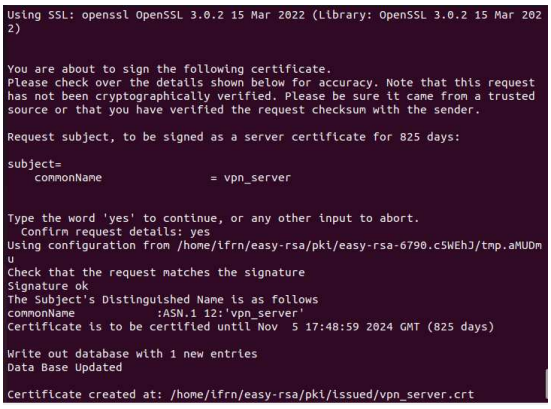


Na pasta pki/private será apresentado a chave do vpn_server. Na pasta pki/issued irá surgir um certificado do servidor, porém ainda não assinado.

Assinando o certificado do servidor OpenVPN

Agora precisamos assinar o certificado do servidor OpenVPN. Então, para isso na pasta $\sim /easy-rsa/$, execute o comando abaixo.

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



Caso solicite uma confirmação dos dados, verifiq ue e digite yes. Ao final irá ser gerado o certificado assinado na pasta pki/issued/vpn_server.crt

Gerando o Parâmetro Diffie Hellman

Para gerar os parâmetros Diffie hellman vamos para o diretório $\sim /easy-rsa/$ e em seguida vamos executar o comando:

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

Como resultado será criado os parâmetros para geração de chaves de 2048 bits e colocado no arquivo ∼/easy-rsa/pki/dh.pem”

Criando um TLS Crypt

O TLS Crypt possibilita o fornecimento de uma chave pré-compartilhada específica para cada cliente. Dessa forma, podemos reduzir os danos caso a chave de um cliente venha a ser comprometida.

Agora vamos criar uma chave TLS Crypt v2 que será usada pelo servidor Open VPN. Assim, inicialmente vamos entrar no diretório ”∼/easy-rsa/pki/”.

```
cd ~/easy-rsa/pki/

openvpn --genkey tls-crypt-v2-
 server private/vpn_server.pem
```

Observe que foi criada uma chave vpn_server.pem no diretório ”∼/easy-rsa/pki/private/”.

Configurando o Servidor OpenVPN

Inicialmente vamos para o diretório ”/etc/openvpn/server”, usaremos o comando:

```
cd /etc/openvpn/server
```

Em seguida, vamos criar um arquivo de configuração com o nome de server.conf.

```
sudo gedit server.conf
```

Nesse arquivo, vamos adicionar as linhas de configuração abaixo:

```
port 1194
proto udp
dev tun

ca ca.crt
cert vpn_server.crt
key vpn_server.key
tls-crypt-v2 vpn_server.pem
dh dh.pem

#network to be used in VPN
server 192.168.200.0 255.255.255.0

push "redirect-gateway autolocal"

ifconfig-pool-persist /var/log/openvpn/ipp.txt
keepalive 10 120
cipher AES-256-GCM
persist-key
persist-tun

status /var/log/openvpn/openvpn-
status.log
verb 4
explicit-exit-notify 1
```

No arquivo de configuração, inicialmente temos a linha ”port 1194”que indica a porta que o servidor OpenVPN usará. Depois temos a opção ”proto udp”que indica que a VPN irá usar o protocolo da camada de transporte UDP. Em seguida, temos a linha ”dev tun”que indica que usaremos um túnel de rotas IP para a VPN.

Depois disso, é informado o certificado da autoridade certificadora, nesse exemplo o arquivo é o ”ca.crt”e a autoridade certificadora também está na mesma máquina do servidor VPN.

Agora vamos informar, respectivamente, o arquivo que tem o certificado do servidor VPN (”cert vpn_server.crt”), a chave que será usada pelo servidor (”key vpn_server.key”), a chave TLS Crypt V2 do servidor OpenVPN (”tls-crypt-v2 server vpn_server.pem”) e algoritmo de criação de chave simétrica e suas configurações, ou seja, arquivo Diffie hellman (”dh dh.pem”).

Na sequência é informado o range de IPs que serão usados na VPN, (”server 192.168.200.0 255.255.255.0”). Esse range de IPs pode ser alterado para evitar que coincida com alguma rede interna.

As opções ”push ”redirect-gateway autolocal força a utilização da VPN como *gateway*, ”ifconfig-pool-persist /var/log/openvpn/ipp.txt”informa o arquivo que será usado para armazenar os endereços virtuais dos clientes da VPN, e ”keepalive 10 120”que indica que o servidor deverá enviar um ping a cada 10 segundos para os clientes e que se o cliente ficar 120 segundos sem responder será considerado desconectado da VPN.

Na sequência será selecionado o algoritmo de criptografia simétrica utilizado "cipher AES-256-GCM"e o arquivo onde serão armazenados os logs das conexões em andamento ("status /var/log/openvpn/openvpn-status.log").

Copiando os Arquivos

Agora iremos copiar os arquivos de certificados e chaves do servidor VPN para dentro do diretório "/etc/openvpn/server/". Inicialmente será o certificado da autoridade certificadora seguindo pelo arquivo de Diffie hellman (geração de chaves simétricas).

```
cd ~/easy-rsa/pki/

sudo cp ca.crt /etc/openvpn/server/

sudo cp dh.pem /etc/openvpn/server/
```

Depois, vamos para o diretório "∼/easy-rsa/pki/private/"para copiarmos os arquivos, ?vpn_server.key? e ?vpn_server.pem?, que contem as chaves necessárias para o servidor VPN.

```
cd ~/easy-rsa/pki/private/

sudo cp vpn_server.key /etc/openvpn/server/

sudo cp vpn_server.pem /etc/openvpn/server/
```

Agora vamos copiar o certificado do servidor que encontra-se no diretório "∼/easy-rsa/pki/issued/", através dos comandos.

```
cd ~/easy-rsa/pki/issued/

sudo cp vpn_server.crt /etc/openvpn/server/
```

```
ifrn@ifrn-VirtualBox:/etc/openvpn/server$ ls
ca.crt  server.conf     vpn_server.key
dh.pem  vpn_server.crt  vpn_server.pem
ifrn@ifrn-VirtualBox:/etc/openvpn/server$
```

Configurando o Firewall

Precisamos assegurar que nosso servidor VPN faça NAT corretamente. Isso porque, as conexões dos clientes serão roteadas por ele. Dessa forma, precisamos habilitar e criar algumas regras de NAT no IPTABLES.

Inicialmente será habilitar o encaminhamento na máquina que será servidora do OpenVPN. Para isso, vamos inserir uma linha no arquivo "/etc/sysctl.conf". Dessa forma, vamos usar o comando abaixo:

```
sudo gedit /etc/sysctl.conf
```

E adicionar a linha abaixo no final do arquivo:

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

Em algumas distribuições Linux, a linha acima já irá aparecer comentada ou já configurada. Agora vamos recarregar o arquivo usando o comando:

```
sudo sysctl -p
```

```
ifrn@ifrn-VirtualBox:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
ifrn@ifrn-VirtualBox:~$
```

Agora precisamos verificar qual é a interface do servidor VPN que encaminha dados para o default gateway. Para isso, vamos usar o comando abaixo:

```
ip route list default
```

Como resultado será mostrado qual a interface do servidor VPN que está fazendo o roteamento padrão, no nosso caso é a "enp0s3".

```
ifrn@ifrn-VirtualBox:~$ ip route list default
default via 10.0.2.1 dev enp0s3 proto dhcp metric 100
ifrn@ifrn-VirtualBox:~$
```

Na sequência, vamos editar o arquivo ("/etc/ufw/before.rules") que faz a leitura preliminar das regras do firewall, através do comando abaixo:

```
sudo gedit /etc/ufw/before.rules
```

Dentro desse arquivo "before.rules", vamos adicionar as linhas abaixo no início do arquivo. Dessa forma, essas linhas que inserimos vão permitir o NAT em nosso servidor VPN.

```
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o enp0s3 -j MASQUERADE
COMMIT
```

Agora vamos editar o arquivo "/etc/default/ufw"para permitir o redirecionamento no firewall. Para isso vamos encontrar a linha "DEFAULT_FORWARD_POLICY="DROP e vamos alterar para "DEFAULT_FORWARD_POLICY="ACCEPT .

```
sudo gedit /etc/default/ufw

DEFAULT_FORWARD_POLICY="ACCEPT"
```

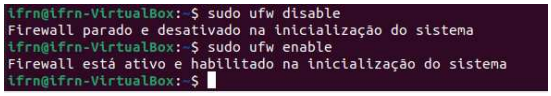
Por último, vamos criar uma regra para permitir acesso na porta do servidor OpenVPN. Em nosso caso, na porta UDP 1194. Para isso use o comando:

```
sudo ufw allow 1194/udp
```

Agora devemos reiniciar o Firewall para iniciar com as novas configurações. Para isso vamos usar os comandos abaixo:

```
sudo ufw disable

sudo ufw enable
```



Inicializando o Servidor OpenVPN

Vamos inicializar o servidor OpenVPN. Para isso, vamos executar o comando abaixo:

```
sudo systemctl start openvpn-
server@server.service
```

Em seguida vamos verificar se o servidor inicializou corretamente com o seguinte comando:

```
sudo systemctl status openvpn-
server@server.service
```

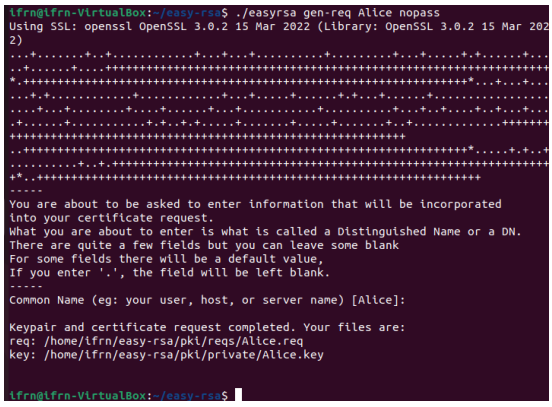
Configurando o Cliente

Instalado e configurado o servidor OpenVPN, agora iremos instalar e configurar o cliente OpenVPN. Para isso, vamos inicialmente criar um certificado e chaves de criptografia para o cliente. Neste exemplo, iremos chamar um cliente hipotético de "Alice".

Criando e assinando o Certificado

Para criar o certificado do cliente OpenVPN , vamos entrar no diretório "∼/easy-rsa", dentro do servidor OpenVPN. Em seguida, vamos digitar o comando abaixo:

```
./easyrsa gen-req Alice nopass
```



No comando anterior criamos o certificado para o usuário Alice. Depois iremos assinar o certificado do cliente OpenVPN, utilizando o o comando abaixo:

```
./easyrsa sign-req client Alice
```

Criando a Chave TLS Crypt

Agora iremos criar a TLS Crypt v2 que será usada pela cliente Alice. Para isso, dentro do servidor OpenVPN, vamos entrar no diretório "∼/easy-rsa/pki/", e executar o comando.

```
openvpn --tls-crypt-v2
private/vpn_server.pem --genkey
tls-crypt-v2-client private/Alice.pem
```

Podemos verificar que a chave de Alice foi criada dentro do diretório "∼/easy-rsa/pki/private/".

Preparando os Arquivos do Clientes

Para configurar o cliente Openvpn, vamos precisar criar um diretório para armazenarmos os arquivos de configuração VPN. Para isso vamos usar o comando abaixo:

```
mkdir ~/vpn_clients
```

Dentro desse diretório vamos criar subdiretórios para cada cliente. Em nosso caso, temos a cliente Alice. Então vamos criar um subdiretório alice.

```
cd ~/vpn_clients

mkdir alice
```

Agora vamos copiar os arquivos ".crt", ".key"e ".pem"e os outros arquivos da Alice para o diretório "∼/vpn_clients/alice?.

```
cd ~/easy-rsa/pki/
cp ca.crt ~/vpn_clients/alice

cd ~/easy-rsa/pki/issued/
cp Alice.crt ~/vpn_clients/alice

cd ~/easy-rsa/pki/private/
cp Alice.key ~/vpn_clients/alice
cp Alice.pem ~/vpn_clients/alice
```

Criando o Arquivo OVPN do Clientes

Na sequência iremos criar o arquivo "make_client_ovpn.sh"dentro do diretório de Alice. Esse arquivo é um *script* que configura a cliente Alice. Para isso, no diretório "∼/vpn_clients/alice", escreva o *script* abaixo:

```
#!/bin/bash

# 1 argument = Client_identifier
cat <(echo -e 'client') \
<(echo -e 'proto udp') \
<(echo -e 'dev tun') \
<(echo -e 'remote 192.168.0.1 1194') \
<(echo -e 'resolv-retry infinite') \
<(echo -e 'nobind') \
<(echo -e 'persist-key') \
<(echo -e 'persist-tun') \
<(echo -e 'remote-cert-tls server') \
<(echo -e 'cipher AES-256-GCM') \
<(echo -e '#user nobody') \
<(echo -e '#group nobody') \
<(echo -e 'verb 3') \
<(echo -e '<ca>') \
ca.crt \
<(echo -e '</ca>\n<cert>') \
${1}.crt \
<(echo -e '</cert>\n<key>') \
${1}.key \
<(echo -e '</key>\n<tls-crypt-v2>') \
${1}.pem \
<(echo -e '</tls-crypt-v2>') \
> ${1}.ovpn
```

No script, as linhas: "client"indica que é um cliente OpenVPN; "proto udp"indica que vai usar o protocolo UDP; "dev tun"indica que vai usar túnel IP; "remote"**192.168.0.1** 1194? indica o IP do servidor OpenVPN e a porta que será usada; "resolv-retry infinite"indica que vai ficar tentando resolver o nome do servidor VPN; "nobind"indica não vai usar uma porta específica, "persist-key"e "persist-tun"permite preservar estado das conexões em caso de reinicialização; "remote-cert-tls server"indica o tls do servidor;

O script indica ainda que irá ser utilizado os arquivos "1.*crt*?+?1.key"+ "1.*pem*?, *sendoo*?1"o nome de identificação que você designou para o cliente. Esse identificador do cliente é primeiro argumento que vamos passar para o script.

Depois de criado o arquivo make_client_ovpn.sh, vamos torná-lo executável. Para isso vamos usar o comando abaixo:

```
chmod +x make_client_ovpn.sh
```

Depois disso vamos executar o *script*, no nosso caso, a cliente Alice.

```
./make_client_ovpn.sh Alice
```

Observe que foi criado um arquivo ".ovpn"dentro da pasta da Alice.

Enviando o Arquivo OVPN para o Cliente

Agora vamos copiar o arquivo Alice.ovpn para o computador do cliente. Para isso, podemos usar diversas formas para copiar como por exemplo SFTP, email, pendrive. Nesta aula ire utilizar o netcat (já utilizado em outras aulas).

Na máquina cliente (Alice), vamos iniciar um servidor netcat usando o comando abaixo para abrir um socket para ouvir a porta 8888 e jogar o conteúdo recebido dentro do arquivo Alice.ovpn.

```
nc -vnl -w 2 8888 > Alice.ovpn
```

Agora vamos para a máquina servidora OpenVPN e vamos para a pasta da Alice. Em seguida vamos digitar o seguinte comando, para enviar o conteúdo do arquivo Alice.ovpn para a máquina cliente (192.168.0.2) utilizando a porta TCP 8888.

```
nc -vn 192.168.0.2 8888 < Alice.ovpn
```

Instalando o OpenVPN no Cliente

Na máquina cliente (Alice) iremos instalar o OpenVPN através do comando:

```
sudo apt-get install openvpn
```

Conectando o Cliente na VPN

Agora vamos conectar o cliente na VPN. Em nosso caso, o cliente é a Alice. Portanto, vamos usar o comando abaixo:

```
sudo openvpn --config Alice.ovpn
```

Testando a VPN

Se toda a configuração deu certo, o cliente já estará acessando a Internet e enviando as mensagens para o servidor usando VPN (criptografia e autenticação). Existem diversas formas para testar se o cliente está encaminhando seu tráfego para a VPN, por exemplo utilizando o Wireshark ou pelo comando traceroute.

Por exemplo, utilizando o comando traceroute para um site da internet, devemos ter um roteamento do tráfego do cliente para o servidor OpenVPN e depois para o site da internet. Por exemplo, vamos rastrear o caminho para o google.com.

```
traceroute www.google.com
```

Atividade

1. Façam duplas e realize a configuração cliente/servidor OpenVPN nas máquinas virtuais do laboratório e depois realize os testes.