O objetivo deste tutorial é fazer o deploy de uma VPN usando OpenVPN.

Índice

Passo 1:

Instalando OpenVPN e EasyRSA

De início, instale o OpenVPN.

```
$ sudo apt update
$ sudo apt install openvpn
```

OpenVPN é uma VPN TLS/SSL. Isso significa que utiliza certificados para criptografar o tráfego na rede. Para configuração de certificados confiáveis, deve ser criado seu próprio certificate authority (CA). Para isso, é necessário a versão mais recente do EasyRSA, que será usado para construir a CA public key infrastructure (PKI).

Então, faça o download da release do EasyRSA. Para isso, localize o link de através da página https://github.com/OpenVPN/easy-rsa/releases/latest, copie-o e faça o download usando o `wget` como no exemplo a seguir:

```
$ wget -P ~/ https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz
```

Então extraia o tarball no diretório `home`:

```
$ cd ~
$ tar xvf EasyRSA-unix-v3.0.6.tgz
```

Renomeie o diretório do arquivo extraído para somente EasyRSA:

```
$ mv EasyRSA-v3.0.6/ EasyRSA/
```

Exclua o arquivo baixado, ele não é mais necessário.

```
$ rm EasyRSA-unix-v3.0.6.tgz
```

Assim está finalizada a instalação da infraestrutura básica da VPN.

Passo 2:

Configurando as Variáveis do EasyRSA e Construindo o CA

Vá para o diretório do EasyRSA:

```
$ cd ~/EasyRSA/
```

Dentro do directory existe um arquivo nomeado `vars.example`. Faça a cópia dele and renomeie-a para `vars`, sem extensão de arquivo:

```
$ cp vars.example vars
```

Abra o arquivo `vars`:

```
$ nano vars
```

Encontre as configurações que definem campos padrão para novos certificados. Tem a seguinte aparência no arquivo:

[~/EasyRSA/vars]

```
. . .
#set_var EASYRSA_REQ_COUNTRY    "US"
#set_var EASYRSA_REQ_PROVINCE   "California"
#set_var EASYRSA_REQ_CITY       "San Francisco"
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL      "me@example.net"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"
. . .
```

Descomente essas linhas e altere os valores padrão para os seguintes:

[~/EasyRSA/vars]

```
. . .
set_var EASYRSA_REQ_COUNTRY     "BR"
set_var EASYRSA_REQ_PROVINCE    "Minas Gerais"
set_var EASYRSA_REQ_CITY        "Perdoes"
set_var EASYRSA_REQ_ORG         "Minasnet Servicos de Provedor de Internet Ltda"
set_var EASYRSA_REQ_EMAIL       "atendimento@minasnet.net"
set_var EASYRSA_REQ_OU          "Centro de Gerenciamento de Redes"
. . .
```

Ao final, salve e feche o arquivo.

Dentro do diretório EasyRSA tem um script chamado `easyrsa` que é usado para executar uma variedade de tarefas envolvendo criação e manutenção do CA. Execute o script com a opção `init-pki` para iniciar a public key infrastructure:

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

Entre com os seguintes parâmetros dentro do prompt:

- **Passphrase:** ********
- **Common Name:** SERVER-OPENVPN

> Mantenha senhas secretas!

Após isso, execute:

```
$ ./easyrsa build-ca
```

Isso contruirá o CA e criará dois arquivos importantes — `ca.crt` e `ca.key` — que compõem os lados público e privado do certificado SSL.

Com isso, o CA está pronto para assinar requisições certificadas.

Passo 3:

Criando o Server Certificate, Key e Encryption Files

Execute `easyrsa`, desta vez usando a opção `gen-req` seguida do common name da máquina.

```
$ ./easyrsa gen-req SERVER-OPENVPN
```

Isso cria a private key do servidor no arquivo `server.req`. Copie o server key para `/etc/openvpn/`:

```
$ sudo cp ~/EasyRSA/pki/private/SERVER-OPENVPN.key /etc/openvpn/
```

Então assine a requição executando `easyrsa` com a opção `sign-req`, seguida pelo *request type* e o *common name*:

```
$ ./easyrsa sign-req server SERVER-OPENVPN
```

Depois, copie os arquivos `server.crt` e `ca.crt` para `/etc/openvpn/`:

```
$ sudo cp pki/issued/SERVER-OPENVPN.crt /etc/openvpn/
$ sudo cp pki/ca.crt /etc/openvpn/
```

Crie uma chave forte de Diffie-Hellman para ser usada durante a troca de chaves:

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

Isso pode demorar alguns minutos. Após o término, gere uma assinatura HMAC para fortalecer a verificação de integridade TLS do servidor:

```
$ sudo openvpn --genkey --secret ta.key
```

Quando terminar, copie os dois novos arquivos para `/etc/openvpn/`:

```
$ sudo cp ta.key /etc/openvpn/
$ sudo cp pki/dh.pem /etc/openvpn/
```

Com isso, todos os arquivos de chave de certificação necessários ao servidor forma gerados. A infraestrutura está pronta para gerar as chaves correspondentes para os clientes que irão acessar ao servidor OpenVPN.

Passo 4:

Configurando o Serviço OpenVPN

Inicie copiando o arquivo de exemplo de configuração OpenVPN e extraia-o para servir de base para o servidor:

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
$ sudo gzip -d /etc/openvpn/server.conf.gz
```

Abra o arquivo de configuração extraído:

```
$ sudo nano /etc/openvpn/server.conf
```

Encontre a seção HMAC pesquisando pela diretiva `tls-auth`. Esta linha deve estar descomentada, então remova o ";" para descomentá-la. Abaixo desta linha, adicione o parâmetro `key-direction`, com valor "0":

[/etc/openvpn/server.conf]

```
tls-auth ta.key 0 # This file is secret
key-direction 0
```

Depois, encontre a seção de cryptographic ciphers pesquisando pelas linhas comentadas `cipher`. A cifra `AES-256-CBC` oferece bom nível de criptografia e é bem suportada. Novamente, esta linha deve ser descomentada, remova o ";" se houver. Abaixo dela, adicione a diretiva `auth` para selecionar o algoritmo HMAC message digest, sendo `SHA256` uma boa escolha:

[/etc/openvpn/server.conf]

```
cipher AES-256-CBC
auth SHA256
```

Depois, encontre a linha contendo a diretiva `dh` que define os parâmetros de Diffie-Hellman. Se necessário, mude o nome de arquivo listado aqui removendo o `2048` de forma que fique com o nome do arquivo gerado anteriormente:

[/etc/openvpn/server.conf]

```
dh dh.pem
```

Encontre as configurações `user` e `group` e remova o ";" no início da linha:

[/etc/openvpn/server.conf]

```
user nobody
group nogroup
```

Modifique as linhas de `cert` e `key` para os apropriados arquivos `.crt` e `.key`, devendo ficar desta forma:

[/etc/openvpn/server.conf]

```
cert SERVER-OPENVPN.crt
key SERVER-OPENVPN.key
```

Também altere o protocolo para `tcp`, descomentando `tcp` e comentando `udp`:

[/etc/openvpn/server.conf]

```
proto tcp;
;proto udp;
```

Após alterar para TCP, será necessário mudar o valor da diretiva `explicit-exit-notify` de 1 para 0, pois ela é utilizada apenas pelo UDP. Não mudar esse valor pode causar erro no serviço OpenVPN:

[/etc/openvpn/server.conf]

```
explicit-exit-notify 0
```

As configurações feitas até aqui não força o tráfego dos dados pela VPN após a conexão do cliente. Para forçar os clientes a usarem o túnel da VPN, é necessário proceder com as configurações a seguir.

Encontre a seção `redirect-gateway` e remova o ";" no início da linha para deixá-la descomentada:

[/etc/openvpn/server.conf]

```
push "redirect-gateway def1 bypass-dhcp"
```

Abaixo dela, encontre a seção `dhcp-option`. Novamente, remova o ";" e deixe da seguinte forma:

[/etc/openvpn/server.conf]

```
push "dhcp-option DNS 177.66.48.12"
push "dhcp-option DNS 177.66.48.13"
```

As mudanças feitas no arquivo de exemplo `server.conf` até aqui são necessárias para o pleno funcionamento do servidor e da conexão.

Passo 5:

Ajustando as Configurações de Rede do Servidor

Ajuste a configuração padrão de redirecionamento de IP do servidor modificando o arquivo `/etc/sysctl.conf`:

```
$ sudo nano /etc/sysctl.conf
```

Dentro, procure pela linha comentada que coloca `net.ipv4.ip_forward`. Remova o "#" do início da linha para descomentar essa configuração:

[/etc/sysctl.conf]

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

Salve e feche o arquivo.

Para carregar as novas configurações para a sessão atual, digite:

```
$ sudo sysctl -p
```

Algumas das configurações do firewall devem ser modificadas para habilitar o mascaramento. Antes de abrir as configurações do firewall e habilitar o mascaramento, primeiramente deveser encontrada a interface de rede pública do servidor. Para isso, digite:

```
$ ip route | grep default
```

Com a interface associada à sua rota padrão, abra o arquivo `/etc/ufw/before.rules` para adicionar a configuração necessária:

```
$ sudo nano /etc/ufw/before.rules
```

No início do arquivo, adicione as linhas abaixo:

[/etc/ufw/before.rules]

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be errors
*filter
. . .
```

Salve e feche o arquivo quando concluir.

Depois, voce tem que configurar o UFW para permitir redirecionamento de pacotes por padrão. Para isso, abra o arquivo `/etc/default/ufw`:

```
$ sudo nano /etc/default/ufw
```

Dentro, encontre a diretiva `DEFAULT_FORWARD_POLICY` e modifique o valor de `DROP` para `ACCEPT`:

[/etc/default/ufw]

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Salve e feche o arquivo.

Agora, ajuste o firewall para habilitar o tráfego do OpenVPN:

```
$ sudo ufw allow 1194/tcp
```

Após adicionar a regra no firewall, desabilite e reabilite o UFW para reiniciá-lo:

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

O servidor está agora pronto para lidar com tráfego da VPN.

Passo 6:

Iniciando e Habilitando o Serviço OpenVPN

O servidor está pronto para inciar o serviço OpenVPN. Antes de iniciá-lo, o passphrase é pedido pelo `systemd`, apenas digite o seguinte comando e preencha o prompt com o passphrase definido anteriormente:

```
$ sudo systemd-tty-ask-password-agent
```

Inicie o servidor OpenVPN:

```
$ sudo systemctl start openvpn@server
```

Verifique se o serviço foi iniciado com sucesso:

```
$ sudo systemctl status openvpn@server
```

Você também pode verificar se a interface do túnel OpenVPN `tun0` está disponível:

```
$ ip addr show tun0
```

O servidor OpenVPN em execução neste momento, se nenhum erro tiver ocorrido. Devido ao passphrase definido no EasyRSA, não é possível iniciar o serviço automaticamente no boot, devendo, portantos, ser iniciado manualmente com os comandos definidos nesta seção.

Passo 7:

Configurando Camanda de Autenticação por Usuário e Senha

Até este momento, o servidor OpenVPN em execução funciona pela autenticação por troca de chaves. Para aumentar o nível de segurança do servidor OpenVPN, será adicionado mais uma camada de autenticação, exigindo, além das chaves, login e senha através do Linux PAM.

Para isso, modifique o arquivo `server.conf` para habilitar o plugin do PAM:

```
$ sudo nano /etc/openvpn/server.conf
```

Adicone ao final dele a seguinte linha:

[/etc/openvpn/server.conf]

```
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so login
```

Salve e feche o arquivo. Após, reinicie o servidor para que o novo módulo seja carregado corretamente:

```
$ sudo reboot
```

Após reiniciar, execute o script de início do servidor OpenVPN, que está no diretório do administrador, já que não é possível iniciá-lo automaticamente:

```
$ ./start-openvpn
```

Digite o passphrase e pronto. Finalmente, verifique se o servidor está em produção:

```
$ sudo systemctl status openvpn@server
```

Se o status não indicar nenhum erro, a VPN está pronta para uso.

Passo 8:

Criando a Infraestrutura de Configuração de Clientes

Primeiramente, faça o download da base de configurações de gerenciamento de clientes OpenVPN:

```
$ wget -P /tmp/ https://williamabreu.github.io/MNET-tutorials/openvpn/download/mnet-openvpn-configs.tar.gz
```

Depois, extraia o tarball:

```
$ cd /tmp/
$ tar xvf mnet-openvpn-configs.tar.gz
```

Entre no diretório extraído e execute o autoconfigurador:

```
$ cd mnet-openvpn-configs/
$ ./configure
```

Agora com a base de configurações de clientes OpenVPN instaladas em `~/client-configs/`, navegue até o diretório raíz do administrador:

```
$ cd ~
```

Neste diretório estão os scripts para facilitar a criação de usuários e início do servidor OpenVPN, são os executáveis `add-openvpn-user` e `start-openvpn`.

Para adicionar um novo usuário, como exemplo o de username sendo `client1`, execute o comando:

```
./add-openvpn-user client1
```

Preencha os dados do prompt corretamente.

Assim que terminar o processo, o arquivo que deve ser enviado para o cliente para que ele possa acessar à VPN está disponível em `~/client-configs/ovpn-files/`, sendo o arquivo cujo nome é o respectivo username com extensão `.ovpn`. Como no exemplo, o arquivo seria `client1.ovpn`. Esse arquivo deve ser enviado por um meio seguro e não pode ser compartilhado.

Repita este último procedimento sempre que for criar um novo usuário da VPN.

> PRONTO!

Referências

- https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-debian-9
- https://www.youtube.com/watch?v=V6DGD4QRXVU