OpenVPN

Présentation

OpenVPN est une solution de connexion VPN basée sur les librairies OpenSSL, transportée sur les protocoles TCP/UDP (donc non filtrée par les Firewall contrairement à IpSec).

Mise en place sous Debian 9

La démarche suivante est appuyée sur un article de Linux Magazine assez ancien et sur la source http://damiengustave.fr/mise-en-place-dun-vpn-avec-openvpn-2-3-et-easy-rsa-3/ (de 2014, en trouver une plus récente).

Prérequis : paramétrage réseau du serveur

Pour que le serveur VPN puisse autoriser le passage du flux du réseau virtuel vers le réseau local, on doit lui activer quelques options et paramètres :

- Permettre le routage entre les réseaux : dans le fichier ***/etc/sysctl.conf***

```
# Décommenter la directive pour conserver le paramétrage à chaque redémarrage
net.ipv4.ip_forward=1
```

- Définir les règles de filtrage autorisant les échanges (exemple avec *iptables*)

```
#autorise l'écoute sur le port
iptables -A INPUT -i <interface_Reseau_Serveur> -p udp -m udp --dport <port_ecoute_vpn> -j ACCEPT
#autorise le routage via l'interface tunnel virtuelle
iptables -A FORWARD -i tun0 -o <interface_Reseau_Serveur> -j ACCEPT
#réalise le NAt entre sur la carte réseau pour le réseau virtuel
iptables -t nat -A POSTROUTING -o <interface_Reseau_Serveur> -j MASQUERADE
iptables -t nat -A POSTROUTING -s <ip_reseau_virtuel/masque> -o <interface_Reseau_Serveur> -j MASQUERADE
```

Pour récupérer les fichiers *easy-rsa*, on devra disposer du logiciel ***git*** :

```
apt install git
```

Procédure

1. Installer le paquetage ***openvpn***

2. récupérer les sources de ***easy-rsa***
3. générer l'infrastructure PKI et l'autorité
4. générer les éléments de sécurité côté serveur
5. générer les éléments de sécurité côté client
6. configurer le fichier ***server.conf***
7. configurer le fichier ***client.conf*** ou ***client.ovpn***

Mode opératoire

Installation

Sous Debian, OpenVPN est référencé dans les dépôts. On peut donc se contenter de l'installer via le gestionnaire de paquetage :

```
apt install openvpn
```

On doit aussi récupérer les fichiers ***easy-rsa*** pour générer les éléments de sécurité.

```
cd /etc/openvpn
git clone https://github.com/OpenVPN/easy-rsa.git
```

Les éléments de sécurité sont générés en s'appuyant sur un fichier ***vars***. Un fichier exemple est fourni dans ***easy-rsa/easyrsa3/vars.example***. On va donc en faire une copie:

```
cd easy-rsa/easyrsa3
cp vars.example vars
```

On peut l'éditer et porter les modifications nécessaires (ou l'utiliser tel quel).

Remarque : Il n'est plus nécessaire de “sourcer” le fichier à chaque action (ancienne commande *source vars* ou *. vars*.

Configuration des éléments de sécurité

Les éléments de sécurité concernent :

- La garantie de l'identité (PKI et CA)
- Les clé/certificat serveur (build-server-full, diffie-hellman, éventuellement authentification TLS)
- Les clé/certificat client (buid-client-full)

Infrastructure de sécurité

L'infrastructure de sécurité permet de créer un *tiers* garantissant la validité des clés et certificat.

Remarque : L'autorité de sécurité n'est pas nécessairement sur le serveur VPN (on aura recours à un tiers externe pour des éléments garantis publiquement).

Les commandes suivantes permettent de créer le tiers (PKI) et le certificat (CA) qui signera tous les couples clés/certificats ultérieurs :

```
 cd easy-rsa/easyrsa3 //se positionne dans le dossier des scripts si ce n'est déjà fait
./easyrsa init-pki //crée le dossier pki (et sous dossiers) pour le stockage des clés/certificats
./easyrsa build-ca //il vous sera demandé de fournir les informations de votre organisation
                   //le fichier ca.crt sera généré dans le dossier pki
```

Éléments de sécurité du serveur

Le serveur doit disposer d'une clé privée (unique, jamais distribuée) et d'un certificat qui sera diffusé aux clients VPN pour établir la connexion. Une commande unique génère ces deux éléments ainsi qu'un fichier *.req* qui permettra l'importation des certificats auprès d'autres machines (si l'autorité est sur un serveur différent du VPN).

```
./easyrsa build-server-full <nom_du_fichier_server>
```

Les fichiers sont générés dans les dossiers

- ***pki/private*** pour les clés,
- ***pki/issued*** pour les certificats
- ***pki/req*** pour les fichiers d'importation de certificats

On peut ajouter un niveau de sécurité en ajoutant la négociation ***diffie-hellman*** qui recalcule les clés d'échange à intervalle (temps, volume, nombre de requêtes) régulier.

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

Le fichier généré est stocké dans le dossier ***pki***.

On peut ajouter un niveau de sécurité au tunnel en générant une clé secrète qui sera pré-partagée (connue des deux extrémités) et nécessaires à l'établissement des premiers échanges.

```
openvpn --genkey --secret <nom_fichier_secret>.key
```

Éléments de sécurité du client

Pour chaque client, on doit générer les trois fichiers (*key, crt et req*) qui permettront de le distinguer et d'établir une connexion.

Chaque client recevra ses éléments personnels ainsi que les éléments de sécurité publics du serveur (ca, dh, tls).

```
./easyrsa build-client-full <nom_du_fichier_client>
```

Les fichiers sont générés dans les dossiers

- ***pki/private*** pour les clés,
- ***pki/issued*** pour les certificats
- ***pki/req*** pour les fichiers d'importation de certificats

Configuration du service sur le serveur OpenVPN

Le service OpenVPN utilise les fichiers de configuration contenus dans le dossier openvpn (ou ses sous-dossiers) Le fichier de configuration du serveur contient les éléments suivants :

```
# Port protocole et interface
port 1194   #port d'écoute du service. 1194 est la valeur IANA
proto udp   #udp est plus léger que TCP (mais risque de moindre qualité). TCP peut écrouler la liaison.
dev tun     #mode de fonctionnement (tunnel ici, il existe aussi tap)


# fichiers de sécurité à prendre en compte
ca <dossier_du_certificat_autorité>/<nom_fichier_ca>.crt
cert <dossier_du_certificat_serveur>/<nom_certif_serveur>.crt
key <dossier_de_la_cle_serveur>/<nom_fichier_cle>.key
dh <dossier_du_diffieHellmann>/<nomfichier_dh>.pem

# adressage du réseau virtuel
server <réseau_ip_virtuel>  <masque_du_reseau_virtuel>

#Option si on souhaite reconnecter les clients avec la même IP en cas de déconnexion du serveur
ifconfig-pool-persist ipp.txt

#garde en cache des informations pour relancer en cas d'interruption
persist-key
persist-tun

#permet de garder le contact entre les extrémités ou de détecter une déconnexion
keepalive 10 120

#compression qui permet d'alléger le trafic
#comp-lzo est une version obsolète
compress lz4-v2
push "compress lz4-v2" #renvoie la propriété vers le client

#restriction des droits du processus en exécution (sécurité)
user nobody
group nogroup

#gestion des logs
status openvpn-status.log
```

```
verb 3

#la directive push route permet de rediriger le trafic du client dans le
tunnel
push "route <ip_reseau> <masque>"
#exemple push "route 192.168.0.0 255.255.0.0"

# option à utiliser si on veut rediriger tout le trafic du client dans le
tunnel (lui reconfigure sa passerelle)
push "redirect-gateway def1 bypass-dhcp"
# réaffecte les valeurs du serveur DNS connu par le client
push "dhcp-option DNS <ip_dns_principal> "
push "dhcp-option DNS <ip_dns_secondaire>"

#option si on veut autoriser les échanges directs entre les clients
client-to-client

#option si on veut permettre plusieurs connexions avec un même certificat
client
duplicate-cn

# à utiliser si on a généré le secret
tls-auth <chemin_du_fichier_secret>/<nom_fichier_secret>.key 0

#algorithme d'encryption
cipher BF-CBC #Blowfish, le plus basique
; cipher AES-128-CBC #AES
;cipher DES-EDE3-CBC #triple-DES
```

- Test du fichier.conf

aller dans /etc/openvpn lancer la commande

```
openvpn <nom_fichier>.conf
```

Si tous fonctionne bien il vous demande un password entrez le mot de passe PEM crée précédemment.

```
Il vous demandera d'utiliser la commande <systemd-tty-ask-password-agent>
```

ensuite entrez le mot de passe PEM une seconde fois.

Paramétrage du client

Le poste client doit disposer des éléments suivants :

- Certificat de l'autorité de certification (ca.crt en général)
- Paramètre de sécurité Diffie-Hellman (fichier .pem)
- Clé secrète pré-partagée (ta.key généralement)

- Clé privée du client générée précédemment (<nom_du_fichier_client>.key)
- Certificat du client généré précédemment (<nom_du_fichier_client>.crt)
- Fichier de configuration *.ovpn* ou *.conf*

Le fichier de configuration du client comportera les éléments suivants :

Remarque : Le nom du fichier .conf ou .ovpn correspondra au nom de la connexion dans le logiciel

```
#indique le mode d'exécution du logiciel : ici en mode client
client
#interface virtuelle pour la communication : ici en mode tunnel
dev tun
#protocole d'échange : doit être le même que pour le serveur
proto udp
#informations du serveur VPN
remote <ip_ou_fqdn_serveur>  <port_du_service>
#mode d'établissement de la connexion
resolv-retry infinite
nobind
#éléments de conservation de la connexion
persist-key
persist-tun
#éléments de sécurité : tout ce qui est utilisé côté serveur doit l'être ici aussi
ca <fichier_autorité_certification>.crt
dh <fichier_diffie_hellman>.pem
#tls-auth <clé_secrète_prépartagée>.key
#éléments de sécurité propre au client
cert <fichier_certificat_client>.crt
key <clé_privée_client>.key
#Compression et affichage
verb 3
compress lz4-v2
pull
```

Révocation d'une clé et de son certificat

Une fois qu'un élément est périmé ou que le compte doit être supprimé, on doit révoquer le certificat pour s'assurer d'une suppression propre.

On doit d'abord se placer dans le dossier de configuration du easyrsa (à adapter à votre configuration)

```
cd /etc/openvpn/easy-rsa/easyrsa
```

On procède à la révocation

```
./easyrsa revoke <nom_du_compte>
```

L'outil propose aussi de générer une nouvelle validation (CRL : certificate revocation list) pour publier sur les sites vérifiant la confiance (optionnel)

```
./easyrsa gen-crl
```

. Cela génère un **.pem** qu'il faudra publier sur les sites appropriés pour empêcher la reconnaissance des objets révoqués.