OpenVPN et PKI

Présentation

Projet :

- installer un **serveur VPN** ;
- Installer et utiliser une **autorité de certification (CA)** sur un serveur différent pour la gestion des certificats nécessaires à l'établissement des tunnels VPN des clients (importation et signature des demandes de certificats).
- Le logiciel **EasyRSA** permettra de créer une infrastructure de clé publique sur le CA, ainsi que la gestion des clés et des demandes de signature des certificats serveur et client.

Ressource :

- https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-debian-10

Installer OpenVPN sur le serveur OpenVPN

```
# apt update
# apt install openvpn
```

Installer une autorité de certification sur un serveur CA dédié (autre serveur que OpenVPN)

Par sécurité il est préférable d'utiliser un serveur dédié, déconnecté du réseau quand il n'est pas utilisé, et qui sera l'autorité de certification.

Téléchargez EasyRSA

```
# wget wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.8/EasyRSA-3.0.8.tgz
# tar xvf EasyRSA-3.0.8.tgz
```

Configurer le CA

- copier le fichier vars.example en **vars**

```
# cd EasyRSA-3.0.8
# cp vars.example vars
```

- éditer le fichier **vars** pour modifier ces lignes en les décommentant

```
#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"
```

- lancer le script **easyrsa** pour initialiser la PKI sur le CA serveur <code shell> # ./easyrsa init-pki </code>
- génération du CA en indiquant le **common name** et en supprimant la nécessité d'utiliser une passphrase pour chaque action sur le CA avec le paramètre **no-pass** : <code shell> # ./easyrsa build-ca nopass </code>

> Le script crée 2 clés importantes :
>
> - **ca.crt** : le **certificat public du CA** qui permettra au serveur et aux clients d'informer qu'ils appartiennent à la même organisation. Les **clients et le serveur** ont donc besoin d'une **copie** de ce certificat.
> - **ca.key** : c'est la **clé privée du CA** qui permet de signer les clés et les certificats des serveurs et des clients. Il est IMPORTANT que cette clé ne soit **jamais communiquée** car celui qui la possède pourra signer des certificats. Cette clé ne doit être présente que sur le CA et celui **hors ligne** quand il n'y a pas de certificat à signer.
>
> :

Le CA est maintenant prêt pour singer des demandes de signature de certificat

Création du certificat signé pour le serveur OpenVPN sur le serveur VPN

- Téléchargez EasyRSA

```
# wget wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.8/EasyRSA-3.0.8.tgz
# tar xvf EasyRSA-3.0.8.tgz
```

- lancer le script **easyrsa** pour initialiser le dossier PKI sur le client OpenVPN <code shell> # cd EasyRSA-3.0.8 # ./easyrsa init-pki </code>
- génération des clés avec l'option **nopass** en précisant le nom du serveur OpenVPN <code shell> # ./easyrsa gen-req OpenVPN nopass </code>

  Création :

  - de la clé privée du serveur OpenVPN.key qu'il faudra copier dans le dossier du logiciel OpenVPN /etc/openvpn/
  - du fichier de demande de certificat OpenVPN.req pour signature auprès du CA.

  - copie de la clé privée du serveur <code shell> # cp pki/private/OpenVPN.key /etc/openvpn/ </code>
  - transfert du fichier OpenVPN.req du serveur OpenVPN dans le dossier /tmp du serveur CA (avec SCP ou Winscp) <code shell> # scp root@ip*serveur*OpenVPN:/root/EasyRSA-3.0.8/pki/reqs/OpenVPN.req /tmp </code >
  - sur le serveur CA, importer le fichier de demande <code shell> # ./easyrsa import-req /tmp/OpenVPN.req OpenVPN </code>
  - signer la demande en précisant le type de client qui est ici un serveur OpenVPN. Il faudra confirmer la demande avec le mot yes <code shell> # ./easyrsa sign-req server OpenVPN </code>
  - transfert du certificat signé du CA vers le serveur OpenVPN <code shell> # scp pki/issued/OpenVPN.crt root@ip*serveur*OpenVPN:/tmp </code>
- transfert du certificat public du CA vers le serveur OpenVPN <code shell> # scp pki/ca.crt root@ip*serveur*OpenVPN:/tmp </code>
- copie sur le serveur OpenVPN des fichier OpenVPN.crt et ca.crt du dossier /tmp dans le dossier /etc/openvpn <code shell> # cp /tmp/{OpenVPN.cet,ca.crt} /etc/openvpn </code>
  - création sur le serveur OpenVPN d'une clé Diffie-Hellman qui sera utilisée pour les échanges de clés <code shell> # ./easyrsa gen-dh </code>

    La clé Diffie-Hellman dh.pem est mise dans le sous-dossier pki

  - génération d'une signature HMAC pour renforcer les capacités de vérification de l'intégrité TLS du serveur: <code shell> # openvpn –genkey –secret ta.key </code>
  - copie de la clé dh.pem et de la signature ta.key dans le dossier /etc/openvp: <code shell> # cp ta.key /etc/openvpn # cp pki/dh.pem /etc/openvpn </code>

    Le serveur OpenVPN est maintenant prêt avec tous les fichiers nécessaires.

    Il ne reste plus qu'à créer les certificats et clés nécessaires pour les clients qui souhaitent utiliser le serveur VPN.

Gestion des certificats signés et des clés pour les clients du serveur OpenVPN

La démarche à suivre est la suivante :

- création d'un script sur le serveur OpenVPN qui va automatiquement générer les fichier de configuration du client VPN contenant tous les clés requises et les certificats. Cela évite d'avoir à transférer les clés, les certificats et les fichiers de configuration aux clients et de rationaliser le processus d'adhésion au VPN.
- Création d'une arborescence sur le serveur OpenVPN pour stocker les certificats et clés des clients

```
# mkdir -p ~/client-configs/keys
```

- Définition des droits pour sécuriser le dossier

```
# chmod -R 700 ~/client-configs
```

- génération des clés avec l'option **nopass** en précisant le nom du client

```
# ./easyrsa gen-req client1 nopass
```

Création :

- de la clé privée du client **client1.key** qu'il faudra copier dans le dossier /client-configs/keys
- du fichier de demande de certificat **client1.req** pour **signature par le CA**.

- copie de la clé privée du client1 dans le dossier /client-configs/keys

```
# cp pki/private/client1.key ~/client-configs/keys
```

- **transfert** de la requête client1.req sur le serveur CA :

```
Depuis le serveur CA
# scp root@ip_serveur_OpenVPN:/root/EasyRSA-3.0.8/pki/reqs/client1.req /tmp
```

- **import** de la demande de certificat à signer :

```
Depuis le serveur CA
# ./easyrsa import-req /tmp/client1.req client1
```

- **signature** du certificat de client1 en précisant qu'il s'agit d'une requête de type client et en confirmant le **common name client1** avec le mot **yes**:

```
Depuis le serveur CA
# ./easyrsa sign-req client client1
```

Un **certificat client appelé client1.crt** est généré et doit être copier dans le dossier /client-configs/keys du serveur OpenVPN :

- copie du certificat du client1.crt dans le dossier /client-configs/keys du serveur OpenVPN

```
Depuis le serveur CA
# scp pki/issued/client1.crt root@ip_serveur_OpenVPN:/root/client-configs/keys
```

- récupération de la clé ta.key du serveur OpenVPN pour la mettre dans le dossier /client-configs/keys du serveur OpenVPN

```
Depuis le serveur OpenVPN
# cd EasyRSA-3.0.8
# cp ta.key ~/client-configs/keys
```

- copie du fichier ca.crt du CA dans le dossier /client-configs/keys

```
Depuis le serveur OpenVPN
# cp /etc/openvpn/ca.crt ~/client-configs/keys
```

Configuration du serveur OpenVPN

Configuration du serveur OpenVPN avec les clés et certificats du serveur et du client

- copier-coller l'exemple de configuration

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

- modification du fichier de configuration /etc/openvpn/server.conf

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

- faites les modifications suivantes :
  - changer **dh dh2048.pem** par **dh dh.pem**
  - ajouter **auth SHA256** après la ligne **cipher AES-256-CBC**

- décommentez (enlever le ;) les lignes suivantes :
  - **user nobody**
  - **group nogroup**
- adapter le nom des clés et certificat au nom du serveur VPN :
  - changer **cert server.crt** par **cert OpenVPN.crt**
  - changer **key server.key** par **cert OpenVPN.key**

Ces modifications permettent la création d'un tunnel VPN entre deux ordinateurs mais n'oblige pas à utiliser le tunnel. Pour router tout le trafic à travers le VPN, il faut pousser la configuration du DNS vers les clients en ajoutant les modifications suivantes au fichier server.conf.

- décommenter la ligne redirect-gateway

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

- indiquer les DNS public fournis par opendns.com. La reconfiguration du DNS fera utiliser le tunnel VPN comme passerelle par défaut

```
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
```

Par défaut OpenVPN utilise le port UDP 1194. Pour autoriser ce port avec **ufw**

```
# ufw allow 1194/udp
```

Lien : https://doc.ubuntu-fr.org/ufw

- configurer le routage en modifiant le fichier /etc/sysctl.conf (décommentez la ligne) <code shell> net.ipv4.ip_forward=1 </code>
  - appliquer la modification <code shell> # sysctl -p </code>
  - modification du parefeu UFW pour le NAT en modifiant le fichier /etc/ufw/before.rules pour ajouter les règles suivantes: <code shell> # # 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

</code>

```
* configurer UFW pour autoriser par défaut forwarded packets en modifiant le fichier /etc/default/ufw
pour accept à l apalce de DROP.
```

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

- activer les nouvelles règles de ufw

```
# ufw disable
```

```
# ufw enable
```

Démarrer le service OpenVPN

- comme le fichier de configuration est /etc/openvpn/server.conf, ajouter @server à l afin du nom

```
# systemctl start openvpn@server
```

- visualiser le status du service openvpn

```
# systemctl status openvpn@server
```

- visualiser l'interface **tun0** du service openvpn

```
# systemctl status openvpn@server
```

- activer le démarrage automatique du service

```
# systemctl enable openvpn@server
```

Création d'un script de génération de la configuration des clients

- création d'un dossier de stockage des configuration clients

```
# mkdir -p ~/client-configs/files
```

- copie d'un exemple de configuration client

```
# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf
```

- modification du fichier ~/client-configs/base.conf

```
# nano ~/client-configs/base.conf
```

- faites les modifications suivantes :
  - changer **remote my-server-1 1194** par **remote @ip_openvpn 1194**
  - ajouter **auth SHA256** après la ligne **cipher AES-256-CBC**
  - décommentez (enlever le ;) les lignes suivantes :
    - **user nobody**
    - **group nogroup**
  - ajouter à la fin du fichier **key-direction 1**
  - ajouter à la fin du fichier les lignes suivantes qui seront utilisées par l'utilitaire resolvconf des clietns Linux :

```
# script-security 2
# up /etc/openvpn/update-resolv-conf
# down /etc/openvpn/update-resolv-conf
```

> Si votre client fonctionne sous Linux et possède un fichier **/etc/openvpn/update-resolv-conf**, décommentez ces lignes du fichier de configuration du client après qu'il ait été généré.

- création du script de génération du fichier de configuration client <code shell> # nano ~/client-configs/make_config.sh </code> Voici le contenu du fichier utilisant l'utilisateur compte qui peut utiliser sudo : <code> #!/bin/bash

First argument: Client identifier

KEY*DIR=/home/compte/client-configs/keys OUTPUT*DIR=/home/compte/client-configs/files BASE_CONFIG=/home/compte/client-configs/base.conf

cat ${BASE*CONFIG}* \ *<(echo -e '<ca>') \ ${KEY*DIR}/ca.crt \

```
  <(echo -e '</ca>\n<cert>') \
  ${KEY_DIR}/${1}.crt \
  <(echo -e '</cert>\n<key>') \
  ${KEY_DIR}/${1}.key \
```

```
  <(echo -e '</key>\n<tls-auth>') \
  ${KEY_DIR}/ta.key \
  <(echo -e '</tls-auth>') \
  > ${OUTPUT_DIR}/${1}.ovpn
```

</code>

```
* rendre le script exécutable
```

```
# chmod 700 ~/client-configs/make_config.sh
```

- générer un fichier de configuration client

```
compte$ cd ~/client-configs
compte$ sudo ./make_config.sh client1
```

> Après exécution du script le fichier de configuration client **client1.ovpn** est créé dans le dossier **/client-configs/files directory**.
>
> Ce fichier doit être transféré sur le client VPN avec scp, sftp ou Winscp.

```
$ sftp sammy@your_server_ip:client-configs/files/client1.ovpn ~/
```

Installation de OpenVPN sur un client Linux

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

- vérifier la présence du script update-resolv-conf <code shell> client$ ls /etc/openvpn </code>
- Transférer le script client1.ovpn situé sur le serveur OpenVPN sur le client <code shell> client$ # scp compte@ip*serveur*OpenVPN:client-configs/files/client1.opvn / </code>
- modifier le fichier client1.ovpn si le script update-resolv-conf existe en décommentant les lignes suivantes : <code shell> client$ nano client1.ovpn </code> <code shell> script-security 2 up /etc/openvpn/update-resolv-conf down /etc/openvpn/update-resolv-conf </code>
- pour éviter le DNS leak, ajouter la lignea suivante à la fin du fichier. Pour en savoir plus DNS leak <code shell> # eviter DNS leak block-outside-dns </code>

> Pour éventuellement **tester la connexion VPN** :
>
> ```
> client$ sudo openvpn --config client1.ovpn
> ```

- pour démarrer automatiquement le client VPN au lancement du client :
  - **copier** le fichier de configuration du client VPN **client1.ovpn** dans **/etc/openvpn** en changeant l'extension ovpn par conf ce qui donne **client1.conf** :

<code shell> client$ sudo cp client1.ovpn /etc/openvpn/client1.conf </code>

- modifier le fichier de configuration **/etc/default/openvpn** en décommentant et en renseignant le nom du fichier de configuration (ne pas mettre l'extension .conf): <code shell> AUTOSTART=“client1” </code>
- activer le lancement automatique du lien VPN au démarrage du client <code shell> client$ sudo systemctl enable openvpn </code>

> Pour **tester le bon fonctionnement du tunnel VPN**, utiliser le site DNSLeakTest pour visualiser l'**adresse IP publique** qui vous permet de naviguer sur Internet. Cette adresse IP doit être celle **serveur VPN** et non plus celle du **routeur de votre réseau** (Box Internet).
>
> Pour visualiser les serveurs DNS utilisés, cliquez sur **Extended Test**.

Révoquer des certificats client

- sur le serveur CA exécuter le script easyrsa et confirmant avec le mot yes :

```
Depuis le serveur CA
# cd  cd EasyRSA-v3.0.8
# ./easyrsa revoke client1
```

- création d'un liste de révocation de certificats (CRL)

```
Depuis le serveur CA
# ./easyrsa gen-crl
```

Un fichier **crl.pem** a été généré.

- transfert du fichier crl.pem sur le serveur OpenVPN dans le dossier /etc/openvpn

```
Depuis le serveur CA
# scp pki/crl.pem root@ip_serveur_OpenVPN:/etc/openvpn
```

- modifier la configuration du serveur OpenVPN pour prendre en compte la liste de certificat révoqué :

```
Depuis le serveur OpenVPN
# nano /etc/openvpn/server.conf
```

- Ajouter à la fin du fichier /etc/openvpn/server.conf a ligne suivante : <code shell> crl-verify crl.pem </code>
    - redémarrer le serveur OpenVPN <code shell> # systemctl restart openvpn@server </code>

Résumé

<uml>

Préparation du CA : création de la PKI

CA → CA : Installer EasyRSA CA → CA : Compléter les variables du CA \ndans le fichier vars CA → CA : Initier la PKI \n→ easyrsa init-pki sur CA CA → CA : Création de la CA avec common name\n→ easyrsa build-ca nopass \n→ création de ca.crt & ca.key

Préparation du serveur OpenVPN

OpenVPN→OpenVPN : Installer OpenVPN OpenVPN → OpenVPN : Installer EasyRSA OpenVPN → OpenVPN : Initier la PKI \n→easyrsa init-pki OpenVPN → OpenVPN : création clé et requête avec common name\n→ easyrsa gen-req \n→ création OpenVPN.key & OpenVPN.req OpenVPN → OpenVPN : copie clé privée OpenVPN.key \ndans le dossier du logiciel OpenVPN /etc/openvpn OpenVPN → CA : transfert de OpenVPN.req CA → CA : Import de fichier OpenVPN.req \n→ easyrsa import-reg CA → CA : Signer le certificat en précisant le type server \n→ easyrsa sign-req server CA → OpenVPN : transfert certificat signé OpenVPN.crt \ndans le dossier du logiciel OpenVPN CA → OpenVPN : transfert certificat public du CA ca.crt \ndans le dossier du logiciel OpenVPN OpenVPN → OpenVPN : Création d'un clé Diffie-Hellman dh.pem \n→ easyrsa gen-dh \n→ mise dans le dossier du logiciel OpenVPN OpenVPN → OpenVPN : génération d'une signature HMAC ta.key \n→ openvpn - -genkey - -secret \n→ mise dans le dossier du logiciel OpenVPN

Préparation du serveur OpenVPN pour gérer les clients

OpenVPN → OpenVPN : Création du dossier ~/client-configs/keys OpenVPN → OpenVPN : Sécurisation du dossier \n→ chmod -R 700 ~/client-configs

Gestion de chaque nouveau client sur le serveur OpenVPN

OpenVPN → OpenVPN : Génération des clés pour un client\n→ easyrsa gen-req client1 nopass\n→ création de client1.key & clienty1.req OpenVPN → OpenVPN : Copie de la clé privée client1.key\ndans le dossier ~/client-configs/keys OpenVPN → CA : transfert de clienty1.req CA → CA : Import de fichier clienty1.req \n→ easyrsa import-reg CA → CA : Signer le certificat en précisant le type client \n→ easyrsa sign-req client CA → OpenVPN : transfert certificat signé client1.crt \ndans le dossier ~/client-configs/keys OpenVPN → OpenVPN : Copie de la clé ta.key de EasyRSA\ndans le dossier ~/client-configs/keys OpenVPN → OpenVPN : Copie de la clé ca.crt de /etc/opernvpn \ndans le dossier

~/client-configs/keys

Configuration du serveur OpenVPN avec clés et certificats serveur et client

OpenVPN → OpenVPN : récupération et modification de l'exemple de configuration serveur.conf OpenVPN → OpenVPN : configuration de ufw pour\n- accepter les connexions sur le port 1194\n- activer le NAT sur eth0\n- autoriser par défaut le forwarding OpenVPN → OpenVPN : configurer le routage \ndans le fichier /etc/sysctl.conf OpenVPN → OpenVPN : démarrer le service openvpn avec le fichier server.conf\n→ systemctl start openvpn@server OpenVPN → OpenVPN : activer le démarrage automartique du service\n→ systemctl enable openvpn@server </uml>