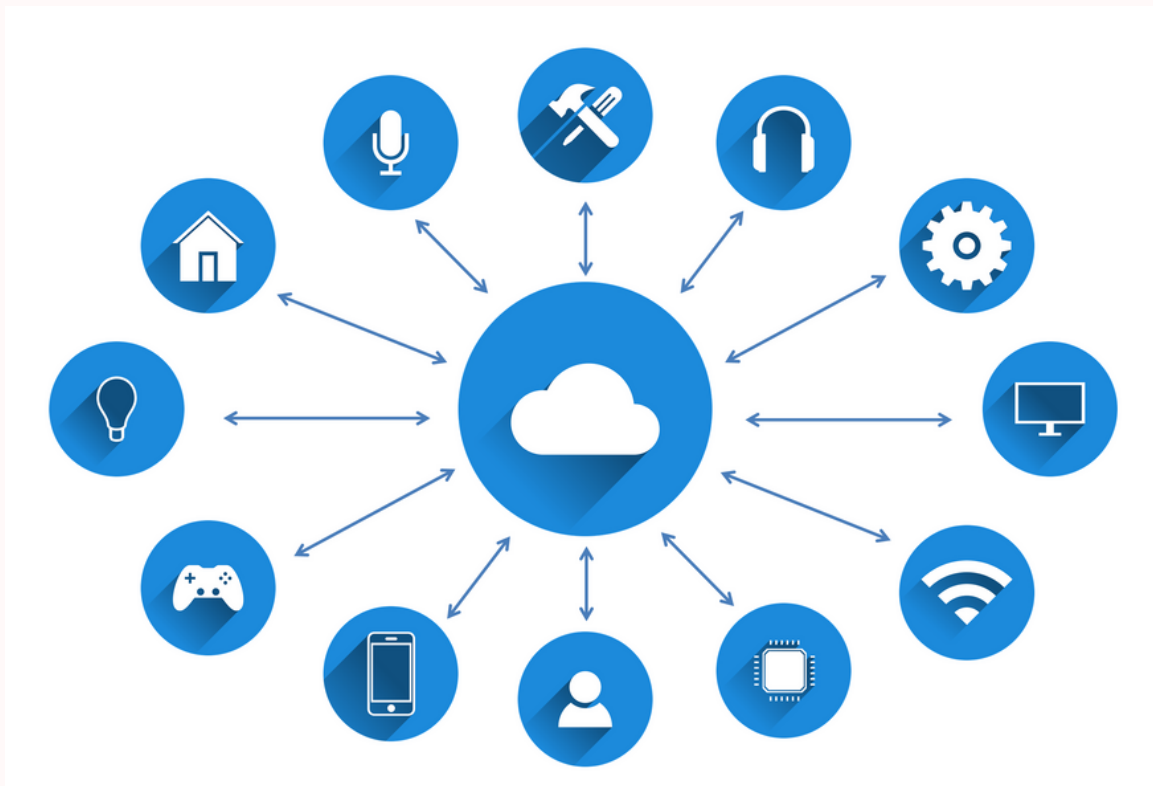
TOULOUSE ynov CAMPUS

PROJET VPN

DOSSIER TECHNIQUE



Nolan Cano
Axelle Refeyton

SOMMAIRE

TOULOUSE ynov CAMPUS

PRÉAMBULE

OBJECTIFS

Le but de ce projet est de donner un accès distant et protégé à un réseau privé. Il faudra donc répondre aux besoins suivants :

- Donner un accès externe au réseau
- Donner une IP publique au réseau
- Sécuriser les interactions avec :
  - Un firewall
  - Une gestion de certificats
  - Une authentification des utilisateurs
  - Un journal de logs

Nous avons choisi d'utiliser OpenVPN, un projet open source offrant beaucoup de fonctionnalités.

PRÉ-REQUIS

Pour mener à bien nos objectifs nous avons besoin :

- 2 postes Linux opérationnel avec un accès administrateur constamment allumés
- Un poste client
- Un accès internet

TOULOUSE ynov CAMPUS

PRÉAMBULE

VPN : INTÉRÊT ET UTILITÉ

Les VPN (Virtual Private Network) offrent une multitude d'avantages et d'utilisations essentiels : contournement des restrictions géographiques, anonymat en ligne, confidentialité et sécurité des données, sécurité sur les réseaux Wi-Fi publics...
Les aspects qui vont en particulier nous intéresser sont la préservation de l'anonymat des utilisateurs ainsi que l'accès à distance aux réseaux privés.

Dans ce document, nous verrons comment mettre en place et configurer les infrastructures suivantes :

- Un serveur d'accès OpenVPN
- Un serveur AC

Ce document indique seulement comment configurer les infrastructures réseau nécessaires au bon fonctionnement du service OpenVPN. Pour se connecter au service en tant que client, veuillez vous référer à la documenation utilisateur.

TOULOUSE
ynov
CAMPUS

LEXIQUE

- **Serveur VPN** : c'est un protocole de communication qui permet de créer une connexion sécurisée et chiffrée entre deux points ou plus. Un VPN utilise des techniques de cryptage pour protéger les données transitant entre les points de connexion, assurant ainsi la confidentialité, l'intégrité et l'authenticité des informations échangées. Un VPN crée un tunnel virtuel entre l'appareil de l'utilisateur et le serveur VPN distant. Lorsqu'une connexion est établie, tout le trafic réseau de l'appareil est encapsulé dans des paquets sécurisés qui sont ensuite transmis via le tunnel VPN. Ces paquets sont protégés par un chiffrement fort, ce qui signifie que même si quelqu'un intercepte le trafic, il ne pourra pas lire les données.

- **Serveur AC** : Serveur de certification, également appelé autorité de certification (AC). C'est un composant central d'une infrastructure à clés publiques qui génère, émet et gère les certificats numériques. Il garantit l'authenticité et la confiance des certificats utilisés pour sécuriser les communications et les transactions en ligne.

TOULOUSE
ynov
CAMPUS

SERVEUR OPENVPN

PRÉPARATION DE LA MACHINE

Sous Ubuntu, mettre à jour l'appareil avec la commande suivante : (https://www.hostinger.fr/tutoriels/comment-configurer-un-serveur-vpn-linux-avec-openvpn)

```
sudo apt update
```

OpenVPN utilise le package net_tools. Pour l'installer sous ubuntu/debian, il faut taper la commande suivante :

```
sudo apt install net-tools
```

La machine est désormais prête à télécharger OpenVPN.

INSTALLATION D'OPEN VPN

(Expliquer pourquoi et ce qu'est OpenVPN)

L'installation diffère selon les systèmes d'exploitation.
Nous utilisons Ubuntu 22, l'installation sera donc celle requise pour Ubuntu 22.
Info User : admin -> mdp : axelleAdmin

Téléchargement du client OpenVPN :

Une fois le binaire téléchargé, il faut l'installer :
https://as-portal.openvpn.com/get-access-server/ubuntu

TOULOUSE ynov CAMPUS

SERVEUR OPENVPN
CRÉATION D'UNE ICP

Dans un premier temps, nous allons créer un répertoire local de l'infrastructure à clé publique sur notre serveur OpenVPN. Nous utiliserons ce répertoire pour gérer les demandes de certificats du serveur et des clients.

Pour créer un répertoire ICP sur notre serveur OpenVPN, nous devons remplir un fichier appelé vars avec quelques valeurs par défaut. Tout d'abord, nous allons cd dans le répertoire easy-rsa, puis nous allons créer et modifier le fichier vars.

Dans le fichier nous allons coller les deux lignes suivante:

```
set_var EASYRSA_ALGO "ec"
set_var EASYRSA_DIGEST "sha512"
```

Puis, nous allons créer le répertoire ICP en utilisant cette commande: "./easyrsa init-pki"

TOULOUSE ynov CAMPUS

SERVEUR OPENVPN

GÉNÉRATION DE CLÉ

Maintenant que notre serveur OpenVPN a installé toutes les conditions préalables, l'étape suivante consiste à générer une clé privée et une demande de signature de certificat (CSR) sur notre serveur OpenVPN.

Pour commencer dans le répertoire ~/easy-rsa et nous allons entrer la commande suivante: "./easyrsa gen-req server"

Puis nous allons utiliser la commande suivante:

sudo cp /home/sammy/easy-rsa/pki/private/server.key /etc/openvpn/server/

Après avoir suivi ces étapes, nous avons créé avec succès une clé privée pour notre serveur OpenVPN. Nous avons également généré une demande de signature de certificat pour le serveur OpenVPN.

le serveur de l'AC doit connaître le serveur et le valider. Une fois que l'AC a validé et relayé le certificat vers le serveur OpenVPN, les clients qui font confiance à votre AC pourront également faire confiance au serveur OpenVPN. Pour ce faire, nous allons utilisé la commande suivante:

scp /home/admin/easy-rsa/pki/reqs/server.req admin@your_ca_server_ip:/tmp

TOULOUSE ynov CAMPUS

SERVEUR CERTIFICATS

A QUOI ÇA SERT ?

Pour notre VPN, nous allons avoir besoin d'un serveur de certificats. Ce dernier présente plusieurs avantages importants:

- Un serveur de certificat permet d'authentifier les clients VPN. Chaque client possède un certificat unique, émis par le serveur de certificat, qui atteste de son identité. Cela garantit que seuls les clients légitimes et approuvés peuvent accéder au réseau VPN.

- L'utilisation de certificats dans un VPN renforce sa sécurité en se basant sur une infrastructure à clé publique (PKI) utilisant des paires de clés cryptographiques (clé publique/privée). Lorsqu'un client se connecte au serveur VPN, son certificat est présenté et vérifié à l'aide de la clé publique correspondante, assurant ainsi une communication sécurisée entre le client et le serveur VPN tout en protégeant les données contre l'interception ou la falsification.

TOULOUSE
ynov
CAMPUS

SERVEUR CERTIFICATS
CONFIGURATION INITIALE

Lors de la première configuration d'un nouveau serveur, il est essentiel d'effectuer certaines étapes de configuration importantes dans le cadre du processus initial. Ces étapes visent à renforcer la sécurité et l'accessibilité de votre serveur, vous offrant ainsi une base solide pour les futures actions à entreprendre.

Pour ce faire, nous allons commencer par créer un nouvel utilisateur avec la commande: adduser <nom de l'utilisateur>

```
nolan@nolan-virtual-machine:~$ adduser nolan
```

Par la suite, pour effectuer des actions nécessitant des privilèges, nous utiliserons la commande "sudo" avec un mot de passe fort afin d'éviter toute utilisation non autorisée des privilèges d'administrateur.

Ensuite, nous allons installé OpenSSH avec la commande:

```
nolan@nolan-virtual-machine:~$ sudo apt install openssh-client
```

a présent nous allons pouvoir configurer un par feu de basse

TOULOUSE ynov CAMPUS

SERVEUR CERTIFICATS

PARE-FEU DE BASE

Pour commencer, nous allons utilisé la commande "ufw app list" pour vérifier que le service qui nous permet de nous connecter à notre serveur, a un profil enregistré auprès de UFW.

```
nolan@nolan-virtual-machine:~$ sudo ufw app list
[sudo] Mot de passe de nolan :
Applications disponibles :
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
```

Ensuite, Nous devons nous assurer que le pare-feu autorise les connexions SSH

```
nolan@nolan-virtual-machine:~$ sudo ufw allow OpenSSH
```

Puis nous allons activé le pare-feu

```
nolan@nolan-virtual-machine:~$ sudo ufw enable
```

TOULOUSE ynov CAMPUS

SERVEUR CERTIFICATS
EASY - RSA

Easy-RSA est un script open source qui simplifie la gestion et la configuration d'un serveur d'autorité de certification (CA). En utilisant Easy-RSA, vous pouvez facilement générer des clés cryptographiques, créer et signer des certificats numériques, ainsi que gérer les révocations de certificats.

Pour l'installer, nous allons utiliser la commandes suivante:

```
nolan@nolan-virtual-machine:~$ sudo apt install easy-rsa
```

Maintenant, nous allons créer un répertoire d'infrastructure à clé public avec la commade "mkdir ~/easy-rsa" puis nous allons faire un lien symbolique:

```
nolan@nolan-virtual-machine:~$ ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

Pour restreindre l’accès à notre nouveau répertoire PKI, afin que seul le propriétaire puisse y accéder, nous allons utiliser la commande "chmod 700 /home/nolan/easy-rsa". Puis on va initialisez l’ICP à l’intérieur du répertoire : easy-rsa avec la commande "./easyrsa init-pki"

TOULOUSE
ynov
CAMPUS

SERVEUR CERTIFICATS
AUTORITÉ DE CERTIFICATATION

Avant de pouvoir créer la clé privée et le certificat de notre autorité de certification, on doit créer et remplir un fichier appelé avec des valeurs par défaut. Pour se faire, nous allons nous rendre dans dans le répertoire easy-rsa

```
nolan@nolan-virtual-machine:~$ cd ~/easy-rsa
nolan@nolan-virtual-machine:~/easy-rsa$ nano vars
```

Puis nous allons modifier les informations du fichier

Sélection nolan@nolan-virtual-machine: ~/easy-rsa

```
 GNU nano 6.2                                                    vars
~/easy-rsa/vars
set_var EASYRSA_REQ_COUNTRY    "FR"
set_var EASYRSA_REQ_PROVINCE   "Toulouse"
set_var EASYRSA_REQ_CITY       "Toulouse"
set_var EASYRSA_REQ_ORG        "YnovCampus"
set_var EASYRSA_REQ_EMAIL      "nolan.cano@ynov.com"
set_var EASYRSA_REQ_OU         "Community"
set_var EASYRSA_ALGO           "ec"
set_var EASYRSA_DIGEST         "sha512"
```

Une fois le fichier modifier, nous allons pouvoir utilisé la commande suivante "./easyrsa build-ca" pour créer la paire de clés publique et privée racine

Nous allons choisir un mot de passe sécurisé que nous entrerons à chaque fois que nous interagirons avec l'autorité de certification afin de renforcer la sécurité.

TOULOUSE ynov CAMPUS

SERVEUR CERTIFICATS

DISTRIBUTION DU CERTIFICAT

Pour se procurer une copie du fichier à partir de notre serveur d’autorité de certification nous allons exécutez la commande suivante en non root "cat ~/easy-rsa/pki/ca.crt"

On va ensuite pouvoir récupérer le contenue du fichier et sur le deuxième système Linux on va pouvoir importer le certificat avec les commandes suivante:

```
nolan@nolan-virtual-machine:~$ sudo cp /tmp/ca.crt /usr/local/share/ca-certificates/
nolan@nolan-virtual-machine:~$ sudo update-ca-certificates
```

Maintenant, votre deuxième système Linux fera confiance à tout certificat qui a été signé par le serveur d’autorité de certification.

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

DEMANDE DE CERTIFICAT

L'étape suivante consiste à nous connecter au serveur de l'AC en tant qu'utilisateur non root que nous avons créé pour gérer notre AC. Avec la commande suivante ous allons importer la demande de certificat:

```
$ ./easyrsa import-req /tmp/server.req server
```

Ensuite il faut signez la demande en exécutant le script:

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

Pour terminer la configuration des certificats, il faut copiez les fichiers server.crt et ca.crt du serveur AC vers le serveur OpenVPN :

```
$ scp pki/issued/server.crt sammy@your_vpn_server_ip:/tmp
$ scp pki/ca.crt sammy@your_vpn_server_ip:/tmp
```

De retour sur votre serveur OpenVPN nous allons copier les fichiers:

```
$ sudo cp /tmp/{server.crt,ca.crt} /etc/openvpn/server
```

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

MATÉRIEL CRYPTOGRAPHIQUE

Pour générer la clé tls-crypt pré-partagée, exécutez les opérations suivantes sur le serveur OpenVPN:

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

Le résultat sera un fichier appelé ta.key. Nous allons le copier dans le répertoire /etc/openvpn/server/ :

```
$ sudo cp ta.key /etc/openvpn/server
```

Nous allons commencez par créer une structure de répertoire dans notre répertoire d'origine pour stocker les fichiers du certificat et de la clé client :

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

Nous allons verrouiller ses autorisations dès maintenant par mesure de sécurité :

```
$ ./easyrsa gen-req client1
```

Ensuite nous allons copiez le fichier client1.key dans le répertoire ~/client-configs/keys/ que vous avez créé précédemment, puis le transférer à notre serveur d'autorité de certification :

```
$ cp pki/private/client1.key ~/client-configs/keys/
$ scp pki/reqs/client1.req admin@your_ca_server_ip:/tmp
```

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

GÉNÉRATION CERTIFICAT

Connectons-nous maintenant à notre serveur AC. Naviguons vers le répertoire EasyRSA et importons la demande de certificat.

```
$ ./easyrsa import-req /tmp/client1.req client1
```

Ensuite, nous allons signer la demande de la même manière que nous l'avons fait pour le serveur à l'étape précédente. Cette fois, cependant, nous veillerons à préciser le type de demande client.

Cela créera un fichier de certificat client nommé client1.crt. Transférez ce fichier vers le serveur :

```
$ scp pki/issued/client1.crt sammy@your_server_ip:/tmp
```

De retour sur notre serveur OpenVPN, copiez le certificat du client sur le répertoire ~/client-configs/keys/ :

```
$ cp /tmp/client1.crt ~/client-configs/keys/
```

Ensuite, copiez le fichier ca.crt et ta.key

```
$ sudo cp /etc/openvpn/server/ca.crt ~/client-configs/keys/
$ sudo chown sammy.sammy ~/client-configs/keys/*
```

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

CONFIGURER OPENVPN

Tout d'abord, nous allons copier l'échantillon server.conf comme point de départ pour notre propre fichier de configuration:

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

```
$ sudo gunzip /etc/openvpn/server/server.conf.gz
```

Nous allons devoir modifier quelques lignes dans le nouveau fichier comme suit:

```
;tls-auth ta.key 0 # This file is secret
tls-crypt ta.key
...
;cipher AES-256-CBC
cipher AES-256-GCM
...
auth SHA256
...
;dh dh2048.pem
dh none
...
user nobody
group nogroup
```

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

CONFIGURATION PARE-FEU

Pour ajuster le paramètre de transfert IP par défaut de notre serveur OpenVPN, nous ouvrirons le fichier /etc/sysctl.conf en utilisant nano ou notre éditeur de texte préféré.

$ net.ipv4.ip_forward = 1

Avant d'ouvrir le fichier de configuration du pare-feu pour ajouter les règles de masquage, nous devons d'abord trouver l'interface de réseau public de notre machine. Pour ce faire, nous taperons :

$ ip route list default

Lorsque l'interface est associée à notre itinéraire par défaut, nous ouvrirons le fichier /etc/ufw/before.rules pour ajouter la configuration appropriée:

$ sudo nano /etc/ufw/before.rules

En haut du fichier, ajoutez les lignes suivante:

```
# 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
```

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

CONFIGURATION PARE-FEU

Ensuite, nous devons indiquer à UFW d'autoriser également les paquets transmis par défaut. Pour ce faire, nous ouvrirons le fichier /etc/default/ufw.

$ sudo nano /etc/default/ufw

À l'intérieur, trouvez la directive DEFAULT_FORWARD_POLICY et changez la valeur de DROP à ACCEPT

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

DÉMARRER OPENVPN

OpenVPN fonctionne comme un service systemd, nous pouvons donc utiliser systemctl pour le gérer. Nous configurerons OpenVPN pour qu'il se lance au démarrage afin que nous puissions nous connecter à notre VPN à tout moment tant que notre serveur est en marche. Pour ce faire, activons le service OpenVPN en l'ajoutant à systemctl.

```
$ sudo systemctl -f enable openvpn-server@server.service
```

Ensuite, on lance le service OpenVPN :

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

Nous avons maintenant terminé la configuration côté serveur pour OpenVPN. Ensuite, Nous allons configurer notre machine cliente et nous connecter au serveur OpenVPN.

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

CONFIGURER MACHINE CLIENTE

Nous allons copier un exemple de fichier de configuration client dans le répertoire client-configs pour l'utiliser comme configuration de base:

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

Nous allons ouvrir ce nouveau dossier et le modifier comme suit:

```
...
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote your_server_ip 1194
...
proto udp
...
# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup
...
```

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

CONFIGURER MACHINE CLIENTE

```
. . .
# SSL/TLS parms.
# See the server config file for more
# description. It's best to use
# a separate .crt/.key file pair
# for each client. A single ca
# file can be used for all clients.
;ca ca.crt
;cert client.crt
;key client.key
. . .
# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1
. . .
cipher AES-256-GCMauth SHA256
. . .
key-direction 1
. . .
; script-security 2; up /etc/openvpn/update-resolv-conf; down
/etc/openvpn/update-resolv-conf
. . .
; script-security 2; up /etc/openvpn/update-systemd-
resolved; down /etc/openvpn/update-systemd-resolved;
down-pre; dhcp-option DOMAIN-ROUTE .
```

Enregistrez et fermez le fichier lorsque vous avez terminé.

TOULOUSE ynov CAMPUS

MISE EN PLACE VPN

CONFIGURER MACHINE CLIENTE

Nous allons pour finir créer un script qui compilera notre configuration de base avec les fichiers de certificat , de clé et de cryptage pertinents, puis nous placerons la configuration générée dans le répertoire~/client-configs/files:

$ nano ~/client-configs/make_config.sh

Dans ce fichier, nous allons ajouter le contenu suivant :

```
#!/bin/bash

# First argument: Client identifier

KEY_DIR=~/client-configs/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/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-crypt>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-crypt>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

TOULOUSE ynov CAMPUS