



Université Paris-Est Créteil
Faculté Des Sciences Et Technologies
Année Universitaire 2015-2016
Master 2 Sécurité Des Systèmes Informatiques
Ier Semestre

COMPTE RENDU DU TP OPENVPN

Etudiants :

DIALLO Mamadou Aliou

LOUE Boureima Arnaud

ALLALI Sara

LAKHILI Safae

Chargé du cours et des TP's :

Mme SOVANNA TAN

INTRODUCTION

Le but de ce TP est de mettre en place une autorité de certification et un serveur VPN avec openvpn.

Un VPN Vitruel Private Network est une interconnexion de réseaux locaux via une technique de tunnel sécurisé, généralement à travers internet.

Un **VPN** repose sur un protocole, appelé protocole de tunnelisation, c'est-à-dire un protocole permettant aux données passant d'une extrémité à l'autre du **VPN** d'être sécurisées par des algorithmes de cryptographie.

Installation de OPENVPN

La commande d'installation de openvpn sur une distribution ubuntu de linux est la suivante :

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

Installation de easy-rsa

« easy-rsa » est un ensemble de scripts qui facilitent l'administration d'une autorité de certification.

La commande d'installation de esay-rsa sur une distribution ubuntu de linux est la suivante :

```
sudo apt-get install esay-rsa
```

Après installation de ces deux outils on procède à des manipulations suivantes :

- Copie du contenu du repertoire easy-rsa dans /etc/openvpn/easy-rsa

```
aliou@aliou4ever: /etc/openvpn
aliou@aliou4ever:/etc/openvpn$ sudo cp -R /usr/share/doc/easy-rsa/* /etc/openvpn/easy-rsa
aliou@aliou4ever:/etc/openvpn$
```

- Ensuite :

```
aliou@aliou4ever:/etc/openvpn/easy-rsa$ sudo make-cadir CA
```

```
aliou@aliou4ever:/etc/openvpn/easy-rsa$ cd CA
aliou@aliou4ever:/etc/openvpn/easy-rsa/CA$ ls
build-ca      build-key-pass  build-req-pass  openssl-0.9.6.cnf  revoke-full
build-dh      build-key-pkcs12 clean-all      openssl-0.9.8.cnf  sign-req
build-inter   build-key-server inherit-inter    openssl-1.0.0.cnf  vars
build-key     build-req       list-crl        pkitool            whichopensslcnf
```

- Puis on crée un répertoire keys (par exemple) dans le répertoire /etc/openvpn/esay-rsa, ce répertoire est destiné à contenir les clés et certificats.

- Enfin on procède à l'initialisation de la PKI (on doit se placer dans le répertoire /etc/openssl/easy-rsa)

Initialisation des variables : `./vars`

Nettoisement de tous les clés et certificats existants : `./clean-all`

Génération de l'autorité de certification (certificat et clé privée du CA)

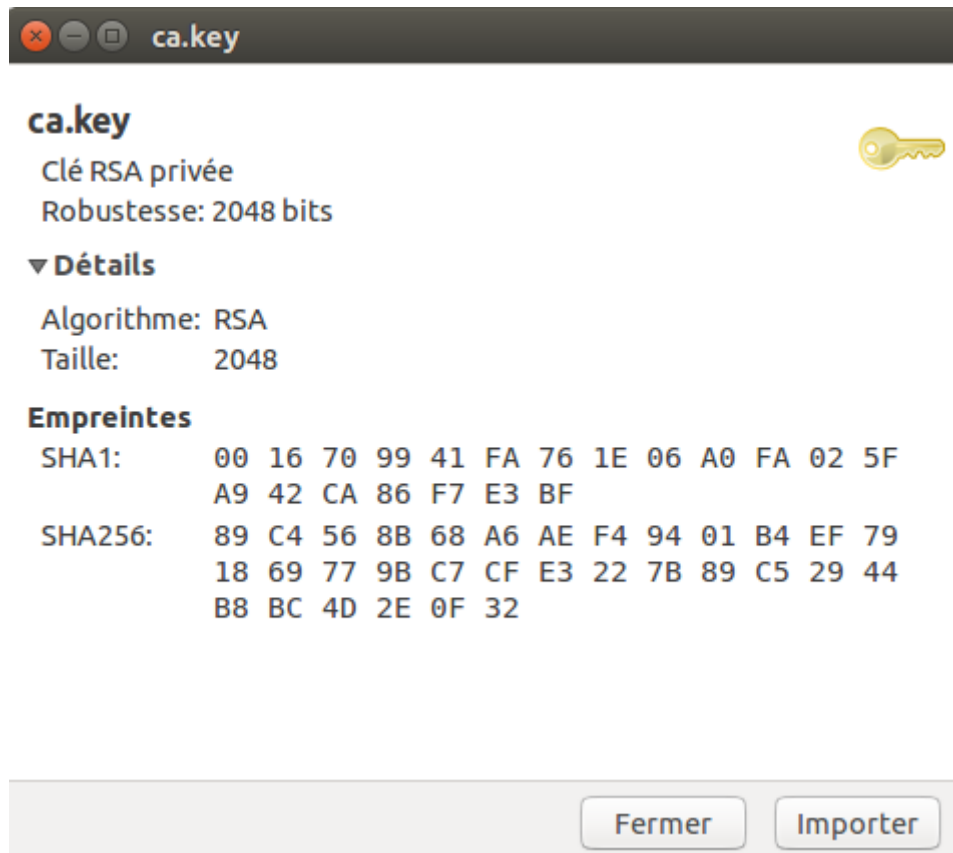
Le CA est en fait l'entité de confiance qui certifie les autres entités notamment le serveur et les clients en signant leurs certificats.

Pour générer une autorité de certification CA c'est à dire générer un certificat et une clé privée pour le CA on tape la commande suivante : `./build-ca`

```
aliou@aliou4ever:/etc/openssl/easy-rsa/CA$ ./clean-all
aliou@aliou4ever:/etc/openssl/easy-rsa/CA$ ./build-ca
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:FR
State or Province Name (full name) [CA]:IDF
Locality Name (eg, city) [SanFrancisco]:Créteil
Organization Name (eg, company) [Fort-Funston]:UPEC
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:UPEC Sciences
Common Name (eg, your name or your server's hostname) [Fort-Funston CA]:UPEC CA
Name [EasyRSA]:CA
Email Address [me@myhost.mydomain]:ca@u-pec.fr
aliou@aliou4ever:/etc/openssl/easy-rsa/CA$
```



Certificat du CA



Clé privée du CA

Génération du serveur VPN

La commande de génération du certificat et de la clé publique du serveur est la suivante :


`./build-key-server aliou-server`


Nous avons appelé le serveur « aliou-server »

On renseigne les informations pour le certificat du serveur.



Certificat du serveur



server-aliou.key

Clé RSA privée
Robustesse: 2048 bits

▼ Détails

Algorithme: RSA
Taille: 2048

Empreintes

SHA1: 10 A9 B0 07 60 68 39 7D D0 D4 75 68 34
F7 DF 85 D1 CD F4 85

SHA256: B2 41 0B 68 7D 31 79 59 0A 6F 73 F5 22
7B 21 88 49 60 80 69 CC 01 B6 96 72 AE
25 38 3A E7 2D 5C

Clé privée du serveur

Génération de clients VPN

Nous allons générer trois clients VPN.

Pour générer un client VPN on tape la commande suivante :

```
./build-key client-arnaud
```



Arnaud

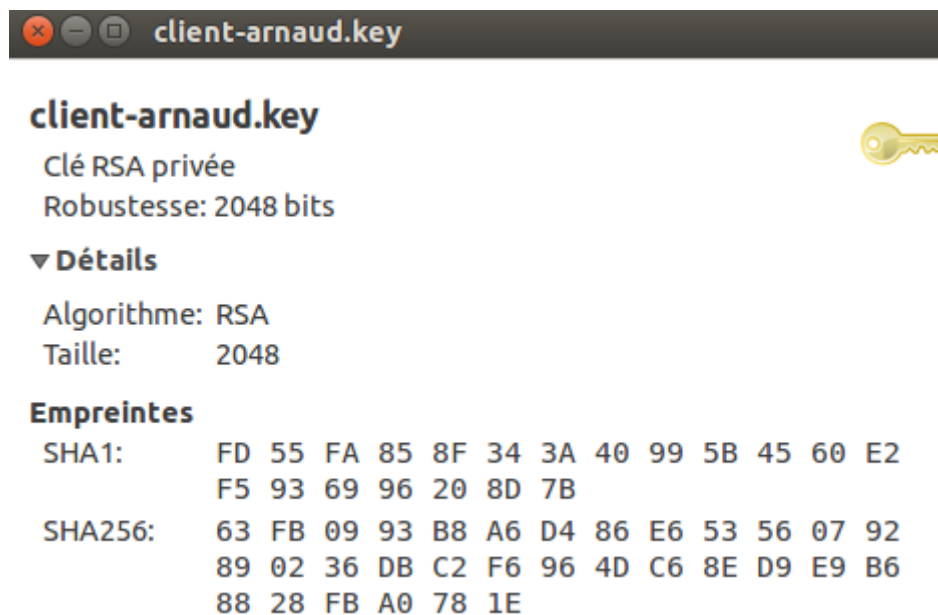
Identité: Arnaud
Vérifié par: UPEC CA
Expire: 16/01/2026

▼ Détails

Nom du sujet

C (Pays): FR
ST (État): IDF
L (Localité): Paris 19
O (Organisation): Arnaud
OU (Unité d'organisation): ArnaudUPEC
CN (Nom courant): Arnaud
2.5.4.41: #130661726E617564
EMAIL (Adresse électronique): arnaud@u-pec.fr

Certificat du client « client-arnaud »



Clé privée du client « client-arnaud »

Pour les deux autres clients :

```
./build-key client-sara
```

```
./build-key client-safae
```

Génération des paramètres de Diffie-Hellman

Diffie-Hellman est un protocole d'échange de clés.

Pour générer les paramètres de Diffie-Hellman on tape la commande suivante :

```
./build-dh
```

```
root@aliou4ever:/etc/openvpn/easy-rsa/CA# ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....
.....+.....+.....
.....
```

cette commande va créer une clé « dh2048.pem »

```

root@aliou4ever:/etc/openvpn/easy-rsa/CA/keys# cat dh2048.pem
-----BEGIN DH PARAMETERS-----
MIIBCAKCAQEA7+gqstLwH9adzp4Qa7q9i5Ayr1ywFrDWPwBkwlybgDQoVoEGpkf
I0dtH255Gp0VICiW+jE4wxq9E86A9zEUTHxTPK1cs6QY7Ao3AQDW7FMPjdKBNC00
cVvKAN/hJmtW37zZUt1azUbYv1lZ+nQk54jQNioAhoA/Fe5nU5W2hA0IZ6eVw2c
LFBIsq4PnZCE0fMpWzp0Wt81tqUAWf1YLSH2D29hvSeXg1toN77vbwKlZoeZpLdq
YKA2eNnQW5SJFDtV2D7kSUu4aYSWqvw0FT2B0CrTLHigpibueNCsMUKzXPFHhIla
9+8afYtINp+09N4am7HtyZLzKTHMGudpSwIBAg==
-----END DH PARAMETERS-----

```

Visualisation de la clé

Configuration du serveur VPN

La configuration du serveur consiste à éditer le fichier de configuration du serveur qui s'appelle « **server.conf** » en modifiant certaines lignes.

```

#emplacement de l'autorité CA
ca /etc/openvpn/easy-rsa/CA/keys/ca.crt
#emplacement du certificat du serveur
cert /etc/openvpn/easy-rsa/CA/keys/server-aliou.crt
#emplacement de la clé du serveur
key /etc/openvpn/easy-rsa/CA/keys/server-aliou.key # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
# openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.

#emplacement du fichier Diffie-Hellman
dh /etc/openvpn/easy-rsa/CA/keys/dh2048.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

```

- A la ligne 2 on donne l'emplacement du certificat de l'autorité de certification.
- A la ligne 4 on donne l'emplacement du certificat du serveur
- A la ligne 6 on donne l'emplacement de la clé privée du serveur
- A la ligne 14 on donne l'emplacement de la clé Diffie-Hellman
- A la dernière on donne l'adresse de réseau du serveur, ici 10.8.0.0/24. Le serveur prendra la première adresse de ce réseau c'est à dire l'adresse **10.8.0.1**. A noter qu'on ne peut pas attribuer n'importe quelle adresse à ce réseau, il existe une plage d'adresse à respecter.

Enfin nous pouvons lancer le serveur en tapant la commande :

```
openvpn server.conf
```

```
aliou@aliou4ever:/etc/openvpn$ sudo openvpn server.conf
[sudo] password for aliou:
Thu Jan 21 18:31:34 2016 OpenVPN 2.3.2 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO]
Thu Jan 21 18:31:34 2016 Diffie-Hellman initialized with 2048 bit key
Thu Jan 21 18:31:34 2016 Socket Buffers: R=[212992->131072] S=[212992->131072]
Thu Jan 21 18:31:34 2016 ROUTE: default_gateway=UNDEF
Thu Jan 21 18:31:34 2016 TUN/TAP device tun0 opened
Thu Jan 21 18:31:34 2016 TUN/TAP TX queue length set to 100
Thu Jan 21 18:31:34 2016 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Thu Jan 21 18:31:34 2016 /sbin/ip link set dev tun0 up mtu 1500
Thu Jan 21 18:31:34 2016 /sbin/ip addr add dev tun0 local 10.8.0.1 peer 10.8.0.2
Thu Jan 21 18:31:34 2016 /sbin/ip route add 10.8.0.0/24 via 10.8.0.2
Thu Jan 21 18:31:34 2016 UDPv4 link local (bound): [undef]
Thu Jan 21 18:31:34 2016 UDPv4 link remote: [undef]
Thu Jan 21 18:31:34 2016 MULTI: multi_init called, r=256 v=256
Thu Jan 21 18:31:34 2016 IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
Thu Jan 21 18:31:34 2016 IFCONFIG POOL LIST
Thu Jan 21 18:31:34 2016 Initialization Sequence Completed
```

Configuration du serveur VPN

La configuration du client consiste à éditer le fichier de configuration du client qui s'appelle « **client.conf** » en modifiant certaines lignes.

```
#emplacement de l'autorité CA
ca /etc/openvpn/easy-rsa/CA/keys/ca.crt
#emplacement du certificat client
cert /etc/openvpn/easy-rsa/CA/keys/client-arnaud.crt
#emplacement de la clé privée du client
key /etc/openvpn/easy-rsa/CA/keys/client-arnaud.key

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 10.8.0.1 1194
```

On précise l'adresse du serveur VPN (10.8.0.1) et son numéro de port (1194) du protocole **UDP** utilisé par le serveur.

On procède maintenant au démarrage du client en tapant la commande :

```
openvpn client.conf
```

```
Fri Jan 22 16:51:05 2016 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Fri Jan 22 16:51:05 2016 /sbin/ip link set dev tun1 up mtu 1500
Fri Jan 22 16:51:05 2016 /sbin/ip addr add dev tun1 local 10.8.0.6 peer 10.8.0.5
Fri Jan 22 16:51:05 2016 /sbin/ip route add 10.8.0.1/32 via 10.8.0.5
Fri Jan 22 16:51:05 2016 Initialization Sequence Completed
```


On va vérifier la connectivité à partir du client en faisant un ping vers l'interface du serveur.

```
aliou@aliou4ever:/etc/openvpn$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.012 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=0.027 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=0.019 ms
64 bytes from 10.8.0.1: icmp_seq=5 ttl=64 time=0.018 ms
64 bytes from 10.8.0.1: icmp_seq=6 ttl=64 time=0.016 ms
64 bytes from 10.8.0.1: icmp_seq=7 ttl=64 time=0.045 ms
64 bytes from 10.8.0.1: icmp_seq=8 ttl=64 time=0.019 ms
64 bytes from 10.8.0.1: icmp_seq=9 ttl=64 time=0.018 ms
64 bytes from 10.8.0.1: icmp_seq=10 ttl=64 time=0.030 ms
64 bytes from 10.8.0.1: icmp_seq=11 ttl=64 time=0.031 ms
64 bytes from 10.8.0.1: icmp_seq=12 ttl=64 time=0.023 ms
64 bytes from 10.8.0.1: icmp_seq=13 ttl=64 time=0.013 ms
64 bytes from 10.8.0.1: icmp_seq=14 ttl=64 time=0.018 ms
```

On voit bien que le ping fonctionne et donc le VPN fonctionne.