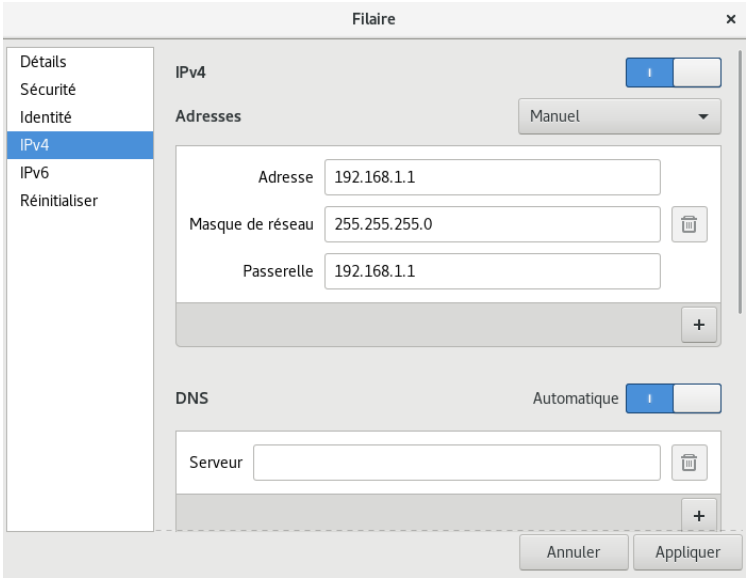
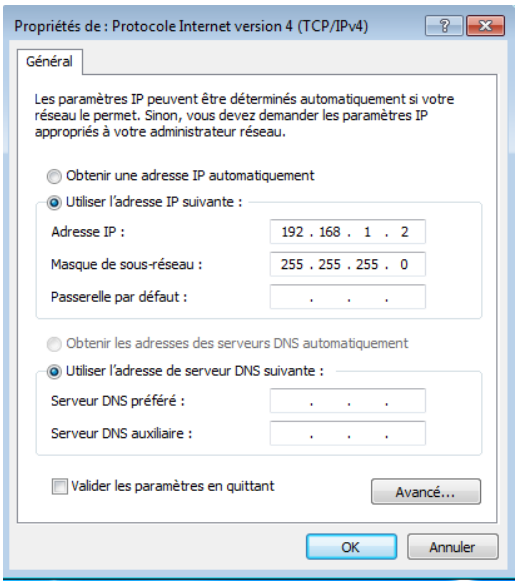
SISR5 – Certificats

I – Installation :

Pour commencer, on importe 2 VM sur VirtualBox. Une sous Windows 7 et l'autre sous Debian 9.4. On installe également PuTTy sur la VM de Windows 7.

Ensuite, on configure nos 2 VM avec des IP fixes :



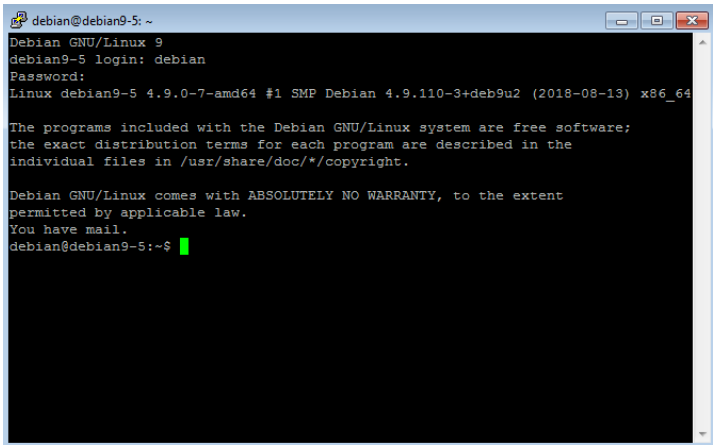


1 - Accès à un serveur à distance

Afin de pouvoir accéder à distance sur notre Debian, il faut d'abord installer telnet, qui nous permettra de nous y connecter par la suite via PuTTY.

Via une connexion par pont, exécuter la commande suivante : $ **apt-get install telnetd**

On peut à présent nous connecter via telnet via PuTTY. Entrez les identifiants debian / debian de la VM sous debian. La connexion sera ensuite établie.



Voici les trames lorsque l'on exécute la commande « ls » via telnet sur la VM Debian :

88	38.279692	aa:bb:cc:00:01:00	CDP/VTP/DTP/PAgP/UD…	DTP	60 Dynamic Trunk Protocol
89	38.279731	aa:bb:cc:00:01:00	CDP/VTP/DTP/PAgP/UD…	DTP	90 Dynamic Trunk Protocol
90	38.808990	192.168.1.2	192.168.1.1	TELNET	55 Telnet Data ...
91	38.809705	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
92	38.904960	192.168.1.2	192.168.1.1	TELNET	55 Telnet Data ...
93	38.905713	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
94	39.118348	192.168.1.2	192.168.1.1	TCP	54 49158 → 23 [ACK] Seq=95 Ack=615 Win=65024 Len=0
95	39.160974	192.168.1.2	192.168.1.1	TELNET	56 Telnet Data ...
96	39.161595	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
97	39.368119	192.168.1.2	192.168.1.1	TCP	54 49158 → 23 [ACK] Seq=97 Ack=617 Win=65024 Len=0
98	39.368765	192.168.1.1	192.168.1.2	TELNET	280 Telnet Data ...
99	39.571591	192.168.1.2	192.168.1.1	TCP	54 49158 → 23 [ACK] Seq=97 Ack=843 Win=64768 Len=0
100	40.164843	aa:bb:cc:00:01:00	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 …
101	42.173742	aa:bb:cc:00:01:00	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 …

2 - SSH avec mot de passe

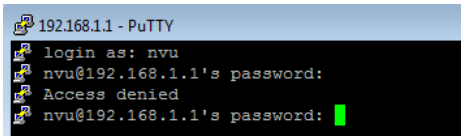
On se connecte à présent en SSH à notre Debian, on répond NON à la question qu'on nous pose. Ensuite il suffit de se connecter avec les identifiants du serveur distant comme en telnet afin de se connecter. Voici à présent les trames que l'on peut observer avec l'accès à distance en SSH :

4	6.031453	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…
5	6.720107	PcsCompu_84:7a:9f	Broadcast	ARP	42 Who has 192.168.1.1? Tell 192.168.1.2
6	6.720288	PcsCompu_e6:7f:84	PcsCompu_84:7a:9f	ARP	60 192.168.1.1 is at 08:00:27:e6:7f:84
7	6.720314	PcsCompu_84:7a:9f	Broadcast	ARP	42 Who has 192.168.1.1? Tell 192.168.1.2
8	6.720447	PcsCompu_e6:7f:84	PcsCompu_84:7a:9f	ARP	60 192.168.1.1 is at 08:00:27:e6:7f:84
9	6.720811	192.168.1.2	192.168.1.1	SSH	118 Client: Encrypted packet (len=64)
10	6.736219	192.168.1.1	192.168.1.2	SSH	118 Server: Encrypted packet (len=64)
11	6.800044	192.168.1.2	192.168.1.1	SSH	118 Client: Encrypted packet (len=64)
12	6.800359	192.168.1.1	192.168.1.2	SSH	118 Server: Encrypted packet (len=64)
13	7.005256	192.168.1.2	192.168.1.1	TCP	54 49159 → 22 [ACK] Seq=129 Ack=129 Win=253 Len=0
14	7.264062	192.168.1.2	192.168.1.1	SSH	118 Client: Encrypted packet (len=64)
15	7.264527	192.168.1.1	192.168.1.2	SSH	118 Server: Encrypted packet (len=64)
16	7.265308	192.168.1.1	192.168.1.2	SSH	294 Server: Encrypted packet (len=240)
17	7.265396	192.168.1.1	192.168.1.2	SSH	118 Server: Encrypted packet (len=64)
18	7.265748	192.168.1.1	192.168.1.2	SSH	150 Server: Encrypted packet (len=96)
19	7.265752	192.168.1.2	192.168.1.1	TCP	54 49159 → 22 [ACK] Seq=193 Ack=433 Win=252 Len=0
20	7.266165	192.168.1.2	192.168.1.1	TCP	54 49159 → 22 [ACK] Seq=193 Ack=593 Win=251 Len=0
21	8.039970	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…

On créer à présent un nouvel utilisateur « nvu » avec les commandes suivantes directement sur le terminal de Debian :

$ su root puis *$ useradd nvu*

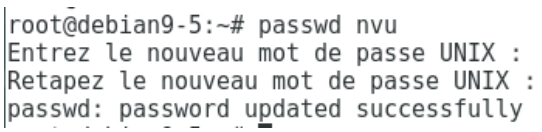
On constate donc qu'en effet il est impossible de se connecter via SSH avec notre nouvel utilisateur : « Access denied ».



Afin de mettre un mot de passe sur notre utilisateur nvu, on exécute la commande suivante :

$ passwd nvu

Ensuite on tape le mot de passe voulu 2 fois.



On réessaie à présent de se connecter avec noter utilisateur sur PuTTy. On peut voir que cette fois-ci la connexion s'établie avec succès. On peut donc en conclure qu'il faut obligatoirement un mot de passe pour se connecter à distance sur un serveur.

3 - Etablissement d'une connexion SSH avec clé asymétrique

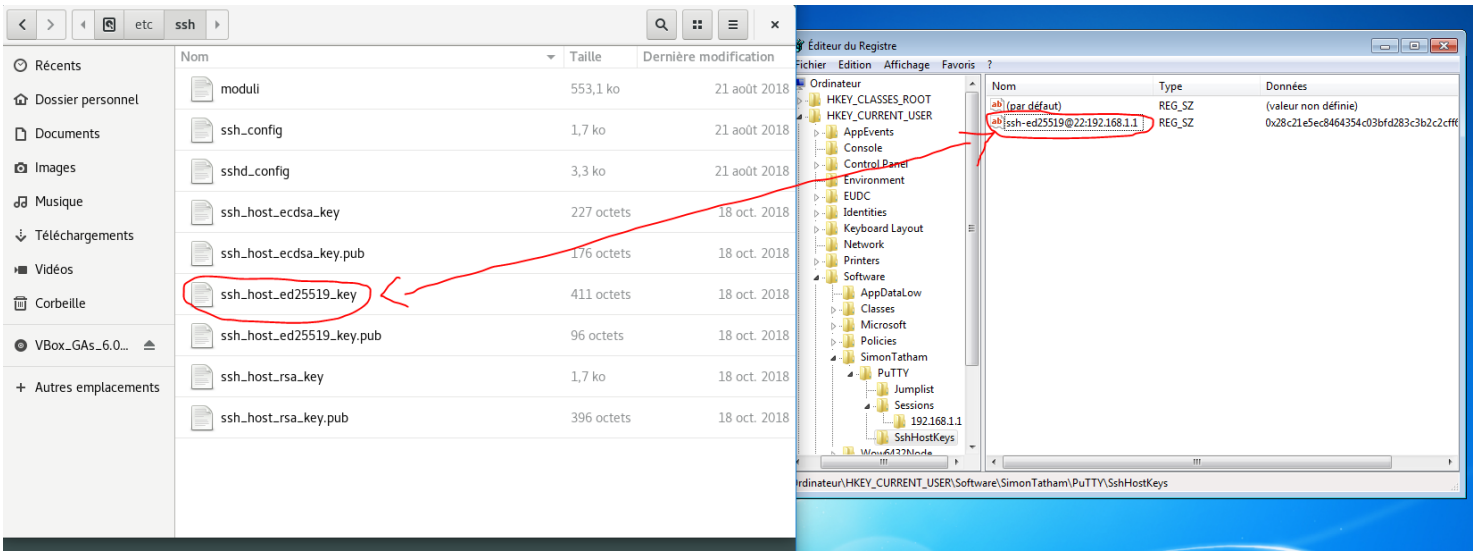
Nous créons à présent un nouvel utilisateur sur Debian appelé nvu2 et un mot de passe (ici nvu2 également).

```
root@debian9-5:~# useradd nvu2
root@debian9-5:~# passwd nvu2
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
```

La connexion à distance via PuTTy s'effectue correctement avec le nouvel utilisateur. On clique bien sur OUI à présent lorsque que l'on pose la question.

On ferme à présent la session, maintenant on rouvre la session, on constate que le message avec la question n'est effectivement pas reposé.

En se rendant dans */etc/ssh* sur Debian et dans le registre *HKEY_CURRENT_USER\Software\SimonThatam\PuTTy* sur Windows, on constate qu'ils ont la même clé publique dans les noms de fichier. En cliquant sur OUI dans le message précédent, on a donc accepté que la clé soit enregistrer afin de ne plus la redemander à chaque fois.



II – Mise en place d'un serveur web sécurisé

Afin de mettre en place le serveur web, on définit tout d'abord un nouveau nom DNS au serveur. Pour se faire, nous devons apporter des modifications aux fichiers `/etc/hosts` et `/etc/hostname`.

```
 GNU nano 2.7.4                Fichier : /etc/hosts

127.0.0.1       localhost
#127.0.1.1      debian9-5.lan   debian9-5
127.0.1.1       serverlilian.home serverlilian

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

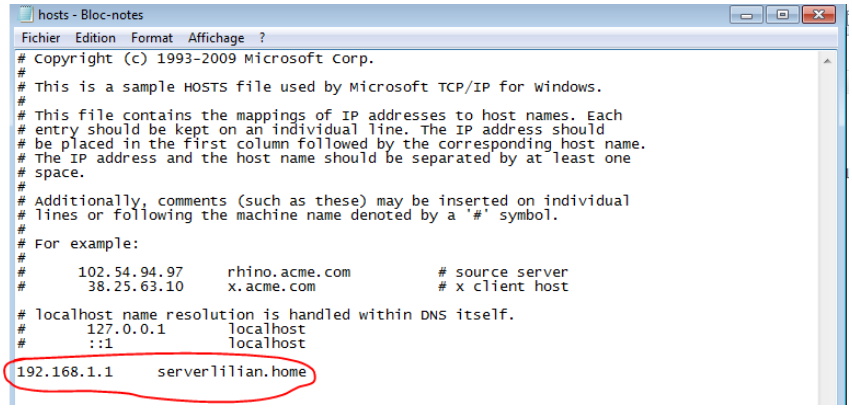
```
 GNU nano 2.7.4                Fichier : /etc/hostname

serverlilian
```

Ensuite, on exécute la commande : `sudo hostname nouveau_nom` afin de l'appliquer.

Sur le poste en Windows 7 à présent, nous devons ajouter une entrée DNS afin qu'il puisse se rendre sur « serverlilian.home ». Pour se faire, se rendre dans le fichier `C:\windows\system32\drivers\etc\hosts` (à ouvrir en tant qu'administrateur). Ajouter une ligne comme ceci :

Ajouter l'adresse IP du serveur distant puis ensuite le FQDN que nous avons entrée dans les fichiers de la Debian.



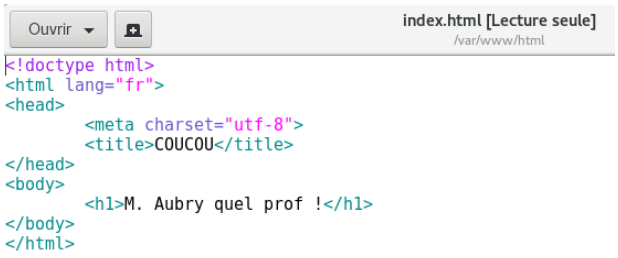
Maintenant que cela est fait, on vérifie que nos changements fonctionnent :

On pingue serverlilian.home et on voit que le retour fonctionne sur l'adresse IP 192.168.1.1.

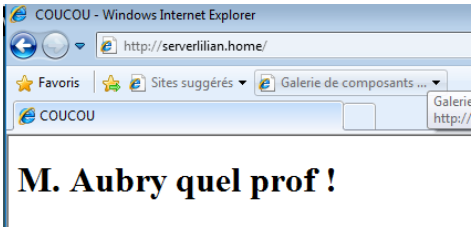
1 - Communication HTTP

Afin de tester sur notre serveur Web correctement après l'installation, nous allons modifier le fichier index.html dans /var/www/html/.

Remplaçons le contenu comme ceci :

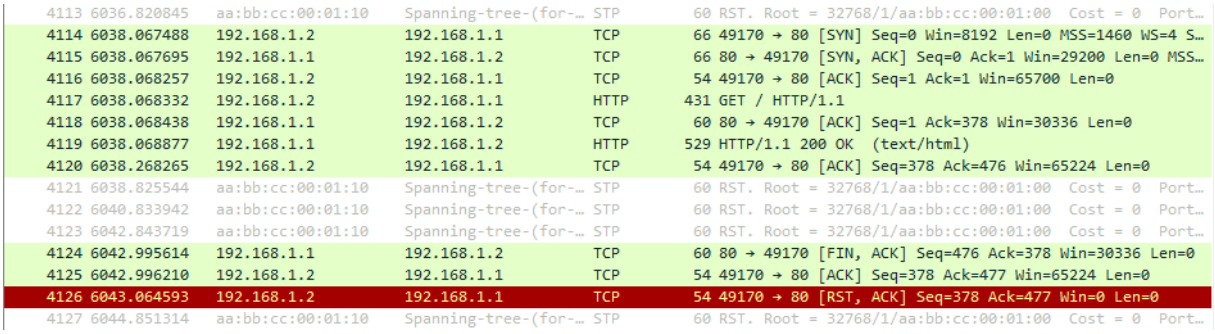


Maintenant rendez-vous depuis notre VM en Windows 7 sur http://serverlilian.home/ depuis Internet Explorer 8. Nous devrions arriver au résultat suivant :



On voit par la même occasion que notre serveur web fonctionne correctement.

Voici à présent la trame de l'échange pour la requête de la page web :



4113 6036.820845	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…
4114 6038.067488	192.168.1.2	192.168.1.1	TCP	66 49170 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 S…
4115 6038.067695	192.168.1.1	192.168.1.2	TCP	66 80 → 49170 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS…
4116 6038.068257	192.168.1.2	192.168.1.1	TCP	54 49170 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4117 6038.068332	192.168.1.2	192.168.1.1	HTTP	431 GET / HTTP/1.1
4118 6038.068438	192.168.1.1	192.168.1.2	TCP	60 80 → 49170 [ACK] Seq=1 Ack=378 Win=30336 Len=0
4119 6038.068877	192.168.1.1	192.168.1.2	HTTP	529 HTTP/1.1 200 OK (text/html)
4120 6038.268265	192.168.1.2	192.168.1.1	TCP	54 49170 → 80 [ACK] Seq=378 Ack=476 Win=65224 Len=0
4121 6038.825544	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…
4122 6040.833942	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…
4123 6042.843719	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…
4124 6042.995614	192.168.1.1	192.168.1.2	TCP	60 80 → 49170 [FIN, ACK] Seq=476 Ack=378 Win=30336 Len=0
4125 6042.996210	192.168.1.2	192.168.1.1	TCP	54 49170 → 80 [ACK] Seq=378 Ack=477 Win=65224 Len=0
4126 6043.064593	192.168.1.2	192.168.1.1	TCP	54 49170 → 80 [RST, ACK] Seq=378 Ack=477 Win=0 Len=0
4127 6044.851314	aa:bb:cc:00:01:10	Spanning-tree-(for-…	STP	60 RST. Root = 32768/1/aa:bb:cc:00:01:00 Cost = 0 Port…

2 - Communication HTTPS

Pour se préparer à activer le protocole HTTPS, suivre les commandes suivantes :

$ cd /etc/apache2

$ mdkir ssl

$ cd ssl

III – Créations des certificats

1 - Création du certificat serveur

Génération de la clé privée : Afin de la générer, tapez la commande suivante afin d'obtenir une clé privée : *$ openssl genrsa 1024 > serverlilian.key*.

```
root@serverlilian:/etc/apache2/ssl# openssl genrsa 1024 > serverlilian.key
Generating RSA private key, 1024 bit long modulus
...++++++
..++++++
e is 65537 (0x010001)
root@serverlilian:/etc/apache2/ssl#
```

Le fichier est donc stocké dans *serverlilian.key*.

A présent, on génère la demande de certificat avec la commande : *$ openssl req -new -key serverlilian.key > serverlilian.csr*.

On obtient quelque chose comme ceci, il suffit d'entrer les informations demandées à chaque ligne :

```
root@serverlilian:/etc/apache2/ssl# openssl req -new -key serverlilian.key > serverlilian.csr
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) [AU]:FR
State or Province Name (full name) [Some-State]:DOUBS
Locality Name (eg, city) []:Besak
Organization Name (eg, company) [Internet Widgits Pty Ltd]:LLB
Organizational Unit Name (eg, section) []:BTSSIO
Common Name (e.g. server FQDN or YOUR name) []:serverlilian.home
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

2 - Création du certificat de l'autorité de certification

A présent, nous devons générer un certificat d'autorité de certification afin d'un reconnu comme une autorité ayant le droit de délivrer des certificats SSL. Pour se faire, on entre la commande suivante : *$ `openssl genrsa -des3 1024 > ca.key`* qui va générer le certificat.

Maintenant, on tape la commande : $ *`openssl req -new -x509 -days 365 -key ca.key > ca.crt`* afin de générer un certificat d'une durée d'un an.

```
root@serverlilian:/etc/apache2/ssl# openssl req -new -x509 -days 365 -key ca.key > ca.crt
Enter pass phrase for 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) [AU]:FR
State or Province Name (full name) [Some-State]:DOUBS
Locality Name (eg, city) []:Besak
Organization Name (eg, company) [Internet Widgits Pty Ltd]:LLB
Organizational Unit Name (eg, section) []:BTSSIO
Common Name (e.g. server FQDN or YOUR name) []:cert_CA
Email Address []:
root@serverlilian:/etc/apache2/ssl#
```

3 - Signature du certificat serveur par le CA (Certificate Autority)

Nous devons maintenant faire signer notre certificat serveur par notre autorité de certification. Pour se faire, on exécute la commande suivante : *$ `openssl x509 -req -in serverlilian.csr -out serverlilian.crt -CA ca.crt -CAkey ca.key -CAcreateserial -CAserial ca.srl`*.

On saisie ensuite notre passphrase (ici « passphrase ») et c'est bon, notre certificat est à présent signé sous « serverlilian.crt ».

```
root@serverlilian:/etc/apache2/ssl# openssl x509 -req -in serverlilian.csr -out
serverlilian.crt -CA ca.crt -CAkey ca.key -CAcreateserial -CAserial ca.srl
Signature ok
subject=C = FR, ST = DOUBS, L = Besak, O = LLB, OU = BTSSIO, CN = serverlilian.h
ome
Getting CA Private Key
Enter pass phrase for ca.key:
root@serverlilian:/etc/apache2/ssl#
```

IV – Installation du certificat d'autorité de certification

1 – Installation sur Internet Explorer et Mozilla

Il faut à présent installer notre certificat SSL sur nos clients afin qu'ils sachent que le site serverlilian.home doit être sécurisé avec notre certificat.

Pour Internet Explorer, il suffit de double cliquer dessus et de l'installer, il sera automatiquement importer pour Internet Explorer.

Pour Mozilla, il faut se rendre dans les paramètres, dans « Vie privée et sécurité », descendre tout en bas puis cliquer sur « Afficher les certificats… ». Maintenant, onglet « Autorités », il faut cliquer sur Importer… puis choisir notre certificat. A présent, notre certificat est importé sur nos deux navigateurs web.

2 – Configuration d'Apache2

Enfin de configurer Apache2, il faut d'abord activer le module SSL avec la commande suivante : *$ `a2enmod ssl`*.

Ensuite il faut ajouter l site SSL par défaut via cette commande : *$ `a2ensite default-ssl`*.
On redémarre ensuite notre serveur web : *$ `service apache2 restart`*.

Il faut à présent éditer le fichier de configuration de SSL qui se situe ici : /etc/apache2/sites-enabled/default-ssl.conf. Suivre les informations entourées en rouge ci-dessous.

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                NameVirtualHost *:443
                ServerName serverlilian.home
                DocumentRoot /var/www/html

                # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                ErrorLog /var/log/apache2/error_ssl.log
                LogLevel warm

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example the
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on
                SSLCertificateFile /etc/apache2/ssl/serverlilian.crt
                SSLCertificateKeyFile /etc/apache2/ssl/serverlilian.key
```

Voici à présent l'analyse de trame permettant de voir que les trames envoyées et réceptionnées sont bien sécurisées.

17	21.343251	192.168.1.2	192.168.1.1	TCP	66 49201 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
18	21.343372	192.168.1.1	192.168.1.2	TCP	66 443 → 49201 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
19	21.343803	192.168.1.2	192.168.1.1	TCP	54 49201 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
20	21.344918	192.168.1.2	192.168.1.1	TLSv1.2	571 Client Hello
21	21.345055	192.168.1.1	192.168.1.2	TCP	60 443 → 49201 [ACK] Seq=1 Ack=518 Win=30336 Len=0
22	21.346197	192.168.1.1	192.168.1.2	TLSv1.2	1235 Server Hello, Certificate, Server Key Exchange, Server Hello Done
23	21.349755	192.168.1.2	192.168.1.1	TLSv1.2	180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
24	21.350109	192.168.1.1	192.168.1.2	TLSv1.2	328 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
25	21.351126	192.168.1.2	192.168.1.1	TLSv1.2	85 Encrypted Alert
26	21.351251	192.168.1.2	192.168.1.1	TCP	54 49201 → 443 [FIN, ACK] Seq=675 Ack=1456 Win=64000 Len=0
27	21.351299	192.168.1.1	192.168.1.2	TLSv1.2	85 Encrypted Alert
28	21.351378	192.168.1.1	192.168.1.2	TCP	60 443 → 49201 [FIN, ACK] Seq=1487 Ack=676 Win=30336 Len=0
29	21.351783	192.168.1.2	192.168.1.1	TCP	54 49201 → 443 [RST, ACK] Seq=676 Ack=1487 Win=0 Len=0
30	21.352579	192.168.1.2	192.168.1.1	TCP	54 49201 → 443 [RST] Seq=676 Win=0 Len=0