

INSTALLATION ET CONFIGURATION D'UN SERVEUR WEB SÉCURISÉ

SERVEUR DNS BIND9

Préparation des fichiers systèmes

Les fichiers suivants se trouvent dans les répertoires « /etc »

- /etc/resolv.conf
- /etc/hosts
- /etc/network/interfaces

Le fichier /etc/network/interfaces

Éditer le fichier « /etc/network/interfaces » et configurer en IP statique, pour éviter le changement d'adresse IP lorsqu'il est configuré dynamiquement. Ensuite le redémarrer afin qu'ils prennent en compte les changements à l'aide de la commande :

/etc/init.d/networking restart

Ce fichier est configuré comme ci-dessous

```
## This file describes the network interfaces available on your system
## and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
#iface enp0s3 inet dhcp
    iface enp0s3 inet static
        address 192.168.0.49
        netmask 255.255.255.0
        gateway 192.168.0.254
```

NB : renseigner bien le nom de votre carte réseau. Pour le connaître taper la commande « ip a »

```
root@dns:/etc/apache2/sites-available# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 08:00:27:8a:f4:ad brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.49/24 brd 192.168.0.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 2a01:e34:eebd:b8d0:a00:27ff:fe8a:f4ad/64 scope global mngtmpaddr dynamic
        valid_lft 86351sec preferred_lft 86351sec
    inet6 fe80::a00:27ff:fe8a:f4ad/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:63:0e:c8 brd ff:ff:ff:ff:ff:ff
root@dns:/etc/apache2/sites-available#
```

Le fichier /etc/host.conf

Lorsqu'une demande de résolution de nom est demandée (c'est-à-dire lorsque l'utilisateur tape par exemple l'adresse `www.google.fr`), le système commence par regarder le fichier « `/etc/host.conf` » ouvre le et modifier comme suit



```
GNU nano 2.7.4 Fichier : /etc/host.conf
order hosts, bind
multi on
```

La première ligne du fichier indique qu'il faut commencer la recherche en regardant la table hosts locale et ensuite il faut interroger le serveur DNS (bind)

Si le résultat n'est pas trouvé dans la table hosts, le système interroge le serveur DNS indiqué dans le fichier « `/etc/resolv.conf` »

Fichier /etc/hosts

La table hosts locale est enregistrée dans le fichier « `/etc/hosts` ». Il contient une table de correspondance entre les adresses IP et des noms de domaine.

```
GNU nano 2.7.4 Fichier : /etc/hosts
127.0.0.1 localhost localhost
192.168.0.49 dns.pawoed.org dns

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

La première ligne est obligatoire pour que le système fonctionne bien même quand le réseau est désactivé. L'adresse 127.0.0.0 est toujours associé au nom localhost.

La ligne suivante peut être ajoutée manuellement pour faire la correspondance entre les adresses IP et des noms

Fichier « /etc/resolv.conf »

À défaut de connaître l'IP de la machine via le fichier de nom « /etc/hosts », le resolver DNS utilise le ou les serveurs DNS mentionnés dans le fichier /etc/resolv.conf pour effectuer ses requêtes.

```
GNU nano 2.7.4 Fichier : /etc/resolv.conf
#nameserver 192.168.0.254 #mon FAI
nameserver 192.168.0.49 #mon DNS
nameserver 8.8.8.8 #DNS google
```

Installation et configuration de bind9

Pour l'installer taper la commande :

#apt install bind9

Après l'installation redémarrer le service avec la commande

#service bind9 restart

Une fois installer vérifier avec les commandes ci-dessous :

Permet de voir la liste des processus en cours d'exécution.:

#ps -ax

```

1091 pts/0      S      0:00 bash
1119 ?          Ssl    0:00 /usr/lib/gvfs/gvfsd-metadata
1432 ?          Ssl    0:00 /usr/sbin/named -f -u bind
1433 ?          S      0:00 [kworker/0:3]
1651 ?          S      0:00 [kworker/u2:1]
1676 ?          S      0:00 [kworker/0:0]
1698 ?          S      0:00 [kworker/0:1]
1708 pts/0      R+     0:00 ps -ax

```

Permet de voir si le port du DNS est ouvert.

#netstat -atn

```

root@dns:/etc/apache2/sites-available# netstat -atn
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat
tcp 0 0 192.168.0.49:53 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:53 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:631 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:953 0.0.0.0:* LISTEN
tcp6 0 0 :::80 :::* LISTEN
tcp6 0 0 :::53 :::* LISTEN
tcp6 0 0 :::22 :::* LISTEN
tcp6 0 0 :::1:631 :::* LISTEN

```

Toutes les configurations DNS sont stockées dans le répertoire /etc/bind.

- /etc/bind/named.conf
- /etc/bind/named.conf.options
- /etc/bind/named.conf.default-zones
- /etc/bind/named.conf.local
- /etc/bind/named.conf.log
- /etc/bind/db.mondomaine(FQDN)
- /etc/bind/db.mondomaine.inv

Fichier /etc/bind/named.conf.local

Ce fichier contient la configuration locale du serveur DNS, on y déclare les zones associées au domaine. On peut configurer autant de zones si nécessaire. Y sont également indiqués les différents fichiers qui seront configurés plus bas : le fichier /etc/bind/db.pawoed.local et le fichier /etc/bind/db.rev

```
GNU nano 2.7.4 Fichier : /etc/bind/named.conf.local
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "pawoed.org" {
    type master;
    file "/etc/bind/db.pawoed.local";
};

zone "0.168.192.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.rev";
};
```

Redémarrer le service à l'aide de la commande

#service bind9 restart

Fichier /etc/bind/db.pawoed.local

Pour configurer ce fichier, le copier à partir du fichier local

#cp /etc/bind/db.local /etc/bind/db.pawoed.local

```
GNU nano 2.7.4 Fichier : /etc/bind/db.pawoed.local
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA dns.pawoed.org. root.pawoed.org. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS dns.pawoed.org.
dns IN A 192.168.0.49
www IN A 192.168.0.49
web IN CNAME www
```

Redémarrer le service à l'aide de la commande

#service bind9 restart

Fichier /etc/bind/db.rev

Pour configurer ce fichier, le copier à partir du fichier local à l'aide de la commande :

```
#cp /etc/bind/db.local /etc/bind/db.rev
```

Le modifier comme suit :

```
GNU nano 2.7.4 Fichier : /etc/bind/db.rev
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      dns.pawoed.org. root.pawoed.org. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       dns.pawoed.org.
49        IN      PTR      dns.pawoed.org
```

[Les commandes utiles](#)

La commande ci-dessous vérifie le bon fonctionnement du service

```
#named-checkconf -z
```

```
root@dns:~# named-checkconf -z
zone pawoed.org/IN: loaded serial 2
zone 0.168.192.in-addr.arpa/IN: loaded serial 1
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
root@dns:~#
```

Les commandes ci-dessous permettent de vérifier la validité des fichiers de zone :

#named-checkzone pawoed.org /etc/bind/db.pawoed.local

```
root@dns:/etc/bind# named-checkzone pawoed.org /etc/bind/db.pawoed.local
zone pawoed.org/IN: loaded serial 2
OK
root@dns:/etc/bind# █
```

La commande « nslookup » permet de vérifier la résolution de nom DNS.

#nslookup dns.pawoed.org

J'ai bien la réponse de mon serveur qui me dit le nom de domaine « dns.pawoed.org » a bien l'adresse IP '192.168.0.49'

```
root@dns:/etc/bind# nslookup dns.pawoed.org
Server:      192.168.0.49
Address:     192.168.0.49#53

Name:   dns.pawoed.org
Address: 192.168.0.49

root@dns:/etc/bind# █
```

La commande « host » permet de tester la résolution du nom e la résolution inverse.

```
root@dns:/etc/bind# host 192.168.0.49
192.168.0.49.in-addr.arpa domain name pointer dns.pawoed.org.0.168.192.in-addr.arpa.
root@dns:/etc/bind# █
```

La commande « dig » permet également de tester la résolution du nom et la résolution inverse, mais surtout d'interroger directement le serveur bind9 et obtenir de nombreuses autres informations. Le paramètre -x est obligatoire pour obtenir une réponse « ANSWER SECTION »

#dig dns.pawoed.org

#dig -x 192.168.0.49

Si je fais « ping dns (dns étant le nom de mon serveur), le nom de domaine pawoed.org, sera ajouté avant d'effectuer la résolution du nom

```
root@dns:/etc/bind# ping dns
PING dns.pawoed.org (192.168.0.49) 56(84) bytes of data:
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=1 ttl=64 time=0.030 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=3 ttl=64 time=0.068 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=5 ttl=64 time=0.068 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=6 ttl=64 time=0.066 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from dns.pawoed.org (192.168.0.49): icmp_seq=8 ttl=64 time=0.058 ms
█
```


SERVEUR APACHE2 ET VIRTUALHOST

Pour installer le serveur

Matériel nécessaire une machine virtuel Debian ou ubuntu

#apt update && install apache2

Après redémarrer le service avec la commande

#service apache2 restart

La commande « netstat -atn » permet de vérifier les ports à l'écoute

Création de l'arborescence

Nous allons par la suite créer un répertoire à la racine du répertoire « /var/www/html »

#mkdir /var/www/html/nom_répertoire

Ensuite se déplacer dans le répertoire nouvellement crée (**cd**

/var/www/html/nom_répertoire (dans mon cas **cd /var/www/html/pawoed**)) et éditer le fichier **index.html** (**nano index.html** ou **vi index.html**)

Construire votre site web

Ex :



```
GNU nano 2.7.4                                Fichier : index.html
<title>Bonjour!!!</title>
<h1>Bienvenue sur mon premier site Web</h1>
```

Configuration des hôtes virtuels ou VirtualHost

Chaque hôte est défini dans un fichier de configuration indépendant que l'on trouve et que l'on crée dans le répertoire « /etc/apache2/sites-available/ », dans ce répertoire se trouve le premier hôte virtuel « 000-default.conf »

Pour cela copier le fichier du premier virtualhost défini dans le répertoire « /etc/apache2/sites-available/ » vers le fichier de chaque hôte à l'aide de la commande suivante :

#cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/site1.conf »

NB : dans ce répertoire on peut créer autant d'hôtes virtuels que nécessaire

Éditer le fichier et modifier les lignes suivantes :

```
GNU nano 2.7.4 Fichier : /etc/apache2/sites-available/site1.conf
VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName www.pawoed.org

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/pawoed
```

Après la configuration du fichier « /etc/apache2/sites-available/site1.conf », activer l'hôte virtuel en créant un lien entre le répertoire « /etc/apache2/sites-available » (répertoire des sites disponibles) et le répertoire « /etc/apache2/sites-enabled » (sites activés)

Pour se faire se déplacer dans le répertoire de sites activés (« cd /etc/apache2/sites-enabled ») : et taper la commande :

#a2ensite site1.conf

Vous devez voir apparaître un message comme ci-dessous :

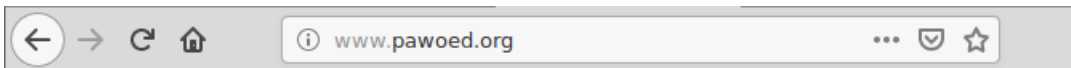
```
root@srv_web1:/etc/apache2/sites-enabled# a2ensite [redacted].conf
Enabling site [redacted].
To activate the new configuration, you need to run:
  systemctl reload apache2
root@srv_web1:/etc/apache2/sites-enabled#
```

Redémarrer le service après toutes ces configurations

#service apache2 restart

Test

Aller sur le navigateur et taper l'URL de votre site <http://www.pawoed.org>, il vous redirige bien sur la page pawoed



Bienvenue sur mon site Web de test

SERVEUR WEB SÉCURISÉ

Installer le paquet openssl à l'aide de la commande

```
#apt install openssl
```

Création des certificats

Connectez-vous sous root et aller dans le répertoire de configuration de votre serveur Apache2 /etc/apache2(on peut évidemment choisir un autre répertoire) et créez un répertoire appelé « ssl ». Vous vous placez dans ce répertoire afin que les clés et les certificats soient créés à l'intérieur avant d'effectuer les manipulations.

- **Génération de la clé privée**

On génère la clé privée avec la commande suivante en définissant un nom de fichier :

```
#openssl genrsa 1024 > serveur.key
```

La sortie attendue est la suivante :

```
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
```

Si vous souhaitez que cette clé ait un mot de passe(qui vous sera demandé à chaque démarrage d'apache, donc à éviter !), ajoutez "-des3" après "genrsa".

Pour observer son contenu taper la commande:

```
#less serveur.key
```

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDQG9wvnuLC4aqzaJCAWGA1AxFzg00hjP0bhq1mukzsGyuuWBFg
vj/k9vFNYX55DHctb/4cXtsZRWwvgcjYnCVwRu+DAjFsk//kOMfhp1miv9xQ+ZL
8w/Xrnm8JWdSS+S4LCMnsuIiQtLbhMrQnUV02hAtbIZiSM3k60jShEZhdQIDAQAB
AoGAHi0cBW+1k+qjFPbB1Uq7UJSMUEKmyYm1vSPCKlTZB0gfVxZzPdDTpEcNks/
yo+rLFSD9Vsvy/9LGmLoXruadWlK67PCUnpM5/oRRGgy8t73YKrxflAU5Gtymjvc
ZCF0cAs6wBft3yLU31Qc4WqVM2vTyUH76jebVhxEw8k630UCQQD/10mAXV+TxBPG
ZTPFbzUeAE5rQqQ0W4aoMNVm61Yn/19h6SzY2MfSQvF1BNns/efCRrqOMeyvPWUG
g1okfogTAkEA0D7pDf/D2Yu5msb0AGF4QBU1erLzpi/s6Rv6VEPYCGnHQ1o3jbg9
FZbjHJ4UcYyYaA8jIrkY+FIJM88Y1GbWxJBAILEdvJ5R/CFcKkF2j2yIWmLaIo1
En8fw43XI5L0PB7Hxx6KDLVu4XzVYQyahTZBdqR0eMlUNZJBhJE2t03wi2cCQQCp
JkCFd3es0BrNrxqfz1ThozRFofcz88za7TldydL0YcFtC4Sb4vWsYizwktZ6jcPEm
rQz8G19W7MO+ynwLptB/AkEA1tsnFXoYzI71enmT dugGxbv0RqAd5iQpDYQkDSdn
2LImp/3YnXNJ9qpY91j87tKthh/Oetu6SH1mLg1LOYNI dw==
-----END RSA PRIVATE KEY-----
```

Pour protéger votre fichier taper la commande :
#chmod 400 serveur.key

- **Création d'un fichier de demande de signature de certificat**

Ce fichier contient la clé publique à certifier

#openssl req -new -key serveur.key > serveur.csr

Le système va vous demander de saisir des champs ; remplissez-les en adaptant sauf le champ « Common Name » qui doit être identique au nom de votre serveur virtuel :

```
Country Name (2 lettre code) [AU] :FR
State or Province Name (full name) [Some-State] :PARIS
Locality Name (eg, city) [ ] :PARIS
Organization Name (eg, company) [Internet Widgits Pty Ltd] :pawoed
Organizational Unit Name (eg, section) [ ] : dns.pawoed.org
Common Name (eg, YOUR name) [ ] :dns.pawoed.org
Email Address [ ] :
```

Ce n'est pas la peine de saisir d'autres « extra attributes »...

Ceci a pour effet de créer le formulaire de demande de certificat (fichier serveur.csr) à partir de notre clé privée préalablement créée.

Ce fichier contient la clé publique à certifier.

Deux possibilités :

- Envoyer le fichier serveur.csr à un organisme (le tiers de confiance ou **l'autorité de certification (CA)**) et ainsi obtenir le certificat dûment signé par la clé privée de l'organisme (après avoir payé),
- **Ou bien signer vous-même le certificat.**

- **Création de son propre certificat de l'autorité de certification**

Pour signer un certificat, vous devez devenir votre propre autorité de certification, cela implique donc de réaliser une clé et un certificat auto-signé.

La création de la clé privée de l'autorité de certification se fait comme précédemment :

#openssl genrsa -des3 1024 > ca.key

Ce qui a pour effet de créer la clé privée de l'autorité de certification. Dans ce cas, il vaut mieux rajouter l'option -des3 qui introduit l'usage d'une "passphrase" (mot de passe long qui peut même contenir du blanc) car c'est cette clé privée qui signera tous les certificats que l'on émettra ; cette "passphrase" sera donc demandée à chaque utilisation de la clé.

Ensuite, à partir de la clé privée, on crée un certificat x509 pour une durée de validité d'un an auto-signé

```
#openssl req -new -x509 -days 365 -key ca.key > ca.crt
```

Il faut saisir la passphrase... puisqu'on utilise "ca.key" que l'on a protégé. Et on donne les renseignements concernant cette fois-ci l'autorité de certification (c'est à dire nous-même).

Attention : le nom de "Common Name" doit être différent de celui qui a été donné pour la clé (ici cert_CA).

```
Country Name (2 lettre code) [AU] :FR
State or Province Name (full name) [Some-State] :PARIS
Locality Name (eg, city) [ ] :PARIS
Organization Name (eg, company) [Internet Widgits Pty Ltd] :pawoed
Organizational Unit Name (eg, section) [ ] : dns.pawoed.org
Common Name (eg, YOUR name) [ ] :cert_CA
Email Address [ ] :
```

C'est notre certificat d'autorité de certification qui va permettre de signer les certificats créés.

- **La signature du certificat serveur par le CA (Certificate Authority)**

La commande qui signe la demande de certificat est la suivante :

```
#openssl x509 -req -in serveur.csr -out serveur.crt -CA ca.crt -CAkey ca.key -CAcreateserial -CAserial ca.srl
```

Il faut saisir la passphrase...

L'option « CAcreateserial » n'est nécessaire que la première fois.

Le certificat signé est le fichier "**serveur.crt**". La sortie de la commande attendue est la suivante :

```
Signature ok
Subject=/FR/ST=PARIS/L=PARIS/O=pawoed/OU=pawoed/CN=dns.pawoed.org
Getting CA Private Key
Enter pass phrase for ca.key :
```

Vous avez maintenant un certificat pour votre serveur qui se nomme « **serveur.crt** »

La commande ci-dessous permet de voir le certificat :

```
#less serveur.crt
```

```

-----BEGIN CERTIFICATE-----
MIICVDCCAb0CAQEWdQYJKoZIhvcNAQEEBQAwdDELMakGA1UEBhMCRlIxFTATBgNV
BAGTDENvcnNIIGR1IFN1ZDEQMA4GA1UEBxMHQWphY2NpbzEMMAoGA1UEChMDTEwC
MREwDwYDVQQLLWVhCVFmG5U5GTzEhMBkGA1UEAxMsY2VydWV1ci5idHNpbmZvLmZy
MB4XDTA0MDIwODE2MjQyNl0XDTA0MDMwOTE2MjQyNl0wTELMAkGA1UEBhMCRlIx
FTATBgNVBAGTDENvcnNIIGR1IFN1ZDEQMA4GA1UEBxMHQWphY2NpbzEMMAoGA1UE
ChMDTEwCMREwDwYDVQQLLWVhCVFmG5U5GTzEYMBYGA1UEAxMPcHJvZi5idHNpbmZv
LmZyMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDSUagxPSv3LtgDV5sygt12
kSbn/NWP0QUiPlksOkF2NkPfwW/mf55dD1hSndlOM/5kLbSB05ieE3TgikF0Iktj
Bwm5xSqewM5QDYzXFt031DrPX63Fvo+tCKTQoVItDeuJPMahVsXnDyYHeUURRWLW
wc0BzEgFZGGw7wiMF6wt5QIDAQABMA0GCSqGSIb3DQEBAQUAA4GBALD640iwKPMf
pqdYtvmLnA7CiEuao60i/pzVJE2LIXXbwYjNAM+7Lov+dFT+b5FcOUGqLymSG3
kSK600auBHI+giGI7C87u4EJaHDvGIUxHxQQGsUM0SCIIVGK7Lwm+8e9I2X0G2GP
9t/rrbdgzXXOC13up99naL5XAzCIp6r5
-----END CERTIFICATE-----

```

- **Installer le certificat de l'autorité de certification dans chaque navigateur client.**

C'est ce dernier qui va valider le certificat reçu par le client lors de la requête https://.

Pour installer sur votre navigateur le certificat de l'autorité de certification, fichier ca.crt :

- Sur Mozilla :

Aller sur **préférence > confidentialité et Sécurité ou Vie Privée et Sécurité > Certificats**

Click sur Afficher les certificats > Autorités, aller tout en bas et cliquer sur « **Importer** » chercher le dossier où se trouve le certificat ca.crt (/etc/apache2/ssl) valider et fermer la fenêtre

Configuration du serveur apache2

Vérifions tout d'abord si le serveur apache2 a déjà le module ssl chargé à l'aide de la commande :

#apache2ctl -M | grep ssl

Le module est chargé si le résultat de la commande renvoie :

```

root@dns:/etc/apache2/sites-available# apache2ctl -M | grep ssl
ssl_module (shared)
root@dns:/etc/apache2/sites-available# █

```

Sinon nous rechargeons le module avec la commande :

#a2enmod ssl

La sortie de cette commande devrait ressembler à ça :

```

root@dns:/etc/apache2/sites-available# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@dns:/etc/apache2/sites-available# █

```

Dans un premier temps il est nécessaire que le serveur apache écoute sur le port 443. Pour cela il suffit d'éditer le fichier /etc/apache2/ports.conf et vérifier que la ligne Listen 443 y soit présente. La meilleure manière de l'intégrer est de l'écrire de la sorte :

```
GNU nano 2.7.4      Fichier : /etc/apache2/ports.conf
## If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 443

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_ssl.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Ainsi, lors de la désactivation (volontaire ou non) du module SSL d'Apache, la configuration restera correcte et ne plantera pas le serveur

Dans un second temps il est nécessaire de créer un virtualhost, Apache prenant en compte notre certificat ssl. Debian et Ubuntu en fournissent un par défaut (/etc/apache2/sites-available/default-ssl) que nous allons modifier pour tester.

Ouvrir le fichier /etc/apache2/sites-available/default-ssl.conf

Chercher les sections suivantes ou rajouter celles qui manquent :

VirtualHost _default_ :443

ServerAdmin webmaster@localhost

Servename dns.pawoed.org :443

Vérifier que les variables suivantes sont correctement définies dans le même fichier :

SSLEngine on

SSLCertificateFile /etc/apache2/ssl/serveur.crt # chemin du fichier du certificat signé

SSLCertificateKeyFile /etc/apache2/ssl/serveur.key #chemin du fichier de la clé privée du certificat

```
GNU nano 2.7.4 Fichier : /etc/apache2/sites-available/default-ssl
IfModule mod_ssl.c>
<VirtualHost _default_:443>
  ServerName dns.pawoed.org:443

  ServerAdmin webmaster@localhost

  DocumentRoot /var/www/html/pawoed

  # Available loglevels: trace8, ..., tracel, 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

  SSLEngine on

  # A self-signed (snakeoil) certificate can be created by installing
  # the ssl-cert package. See
  # /usr/share/doc/apache2/README.Debian.gz for more info.
  # If both key and certificate are stored in the same file, only the
  # SSLCertificateFile directive is needed.
  SSLCertificateFile /etc/apache2/ssl/serveur.crt
  SSLCertificateKeyFile /etc/apache2/ssl/serveur.key
  ErrorLog /var/log/apache2/error_ssl.log
  LogLevel warn
```

NB :

1-attention à bien renseigner le bon répertoire du DocumentRoot (c'est dans ce répertoire que se trouve la configuration de votre site)

2-penser à créer le répertoire ErrorLog renseigner dans ce fichier

Activation du VirtualHost et redémarrage du serveur apache2

À l'aide de la commande ci-dessous nous allons activer l'hôte virtuel (en étant dans le répertoire /etc/apache2/sites-available) :

#a2ensite default-ssl

Redémarrer avec la commande

#service apache2 restart

Test

À partir d'un navigateur sur votre poste client renseigner votre url :

<https://dns.pawoed.org> ou <https://www.pawoed.org>

