INSTALLATION ET CONFIGURATION D'UN SERVEUR WEB SÉCURISÉ

SERVEUR DNS BIND9

Préparation des fichiers systèmes

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

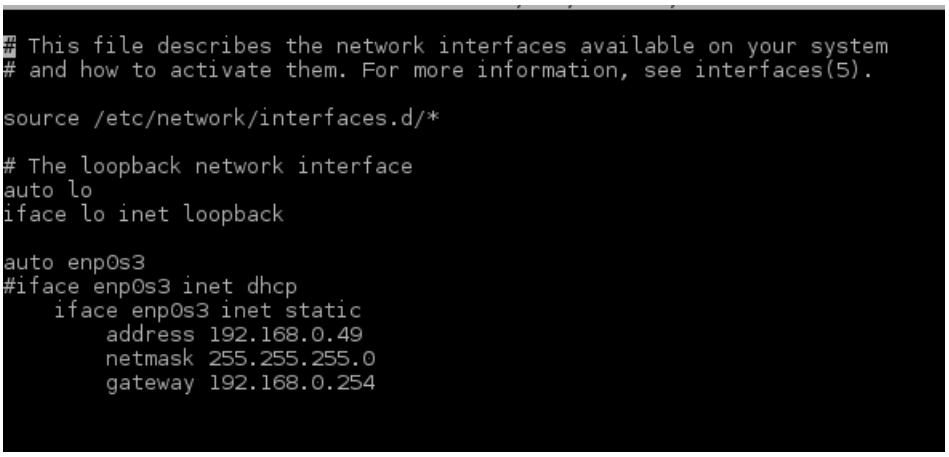
- /etc/resolv.conf
- /etc/hosts
- /etc/network/interfaces

Le fichier /etc/netwok/interfaces

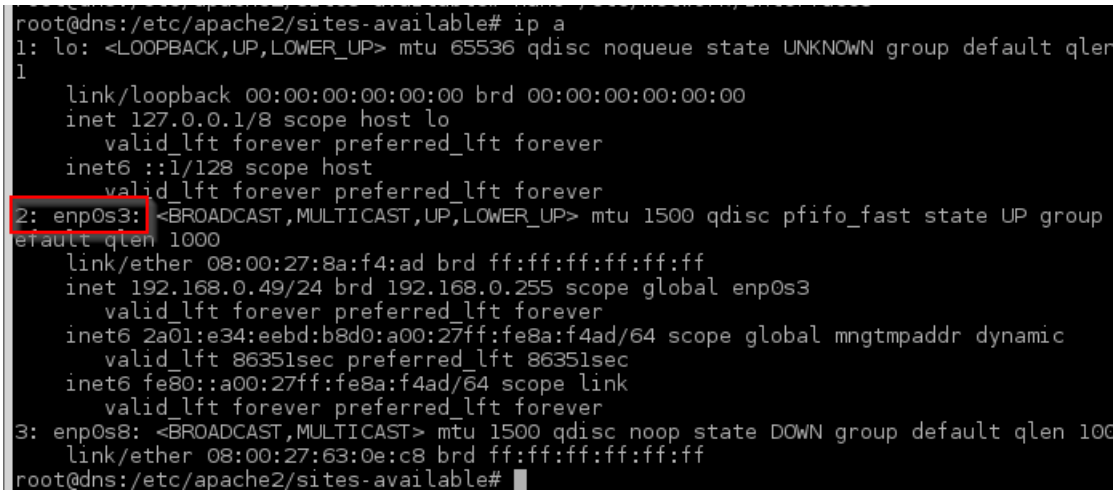
Éditer le fichier « /etc/network/interfaces » et configurer en IP statique, pour éviter le changement d'adresse IP lorsqu'il est configuré dynamiquement. Ensuite le redémarrer afin qu'ils prennent en compte les changements à l'aide de la commande :

 # /etc/init.d/networking restart

Ce fichier est configuré comme ci-dessous

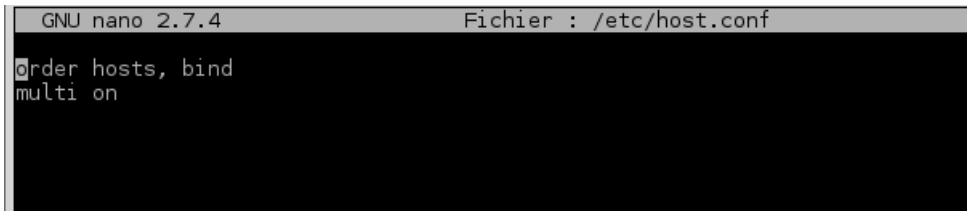


NB : renseigner bien le nom de votre carte réseau. Pour le connaitre taper la commande « ip a »



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 » ouvrez le et modifier comme suit

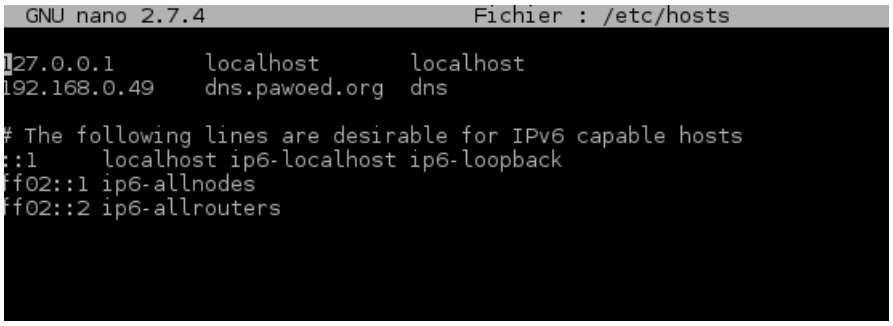


La première ligne du fichier indique qu'il faut commencer la recherche en regardant la table hosts locale et ensuite il faut interroger le serveur DNS (bind)

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.

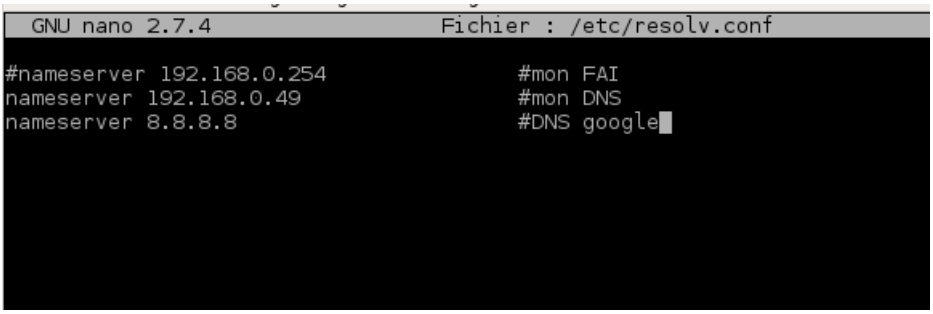


La première ligne est obligatoire pour que le système fonctionne bien même quand le réseau est désactivé. L'adresse 127.0.0.0 est toujours associé au nom localhost.

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 connaitre 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.



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 :

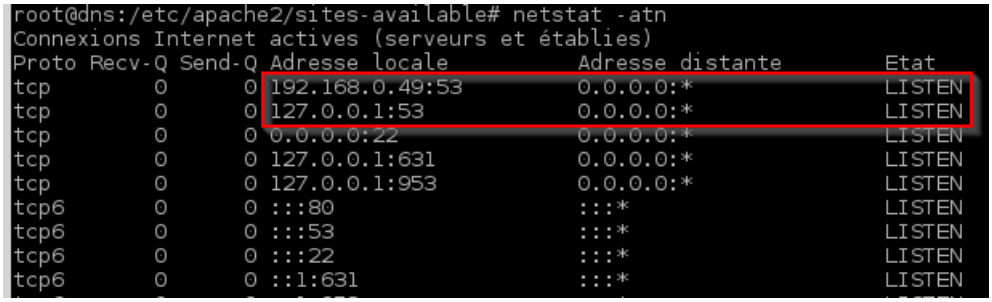
Permet de voir la liste des processus en cours d'exécution.:

#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

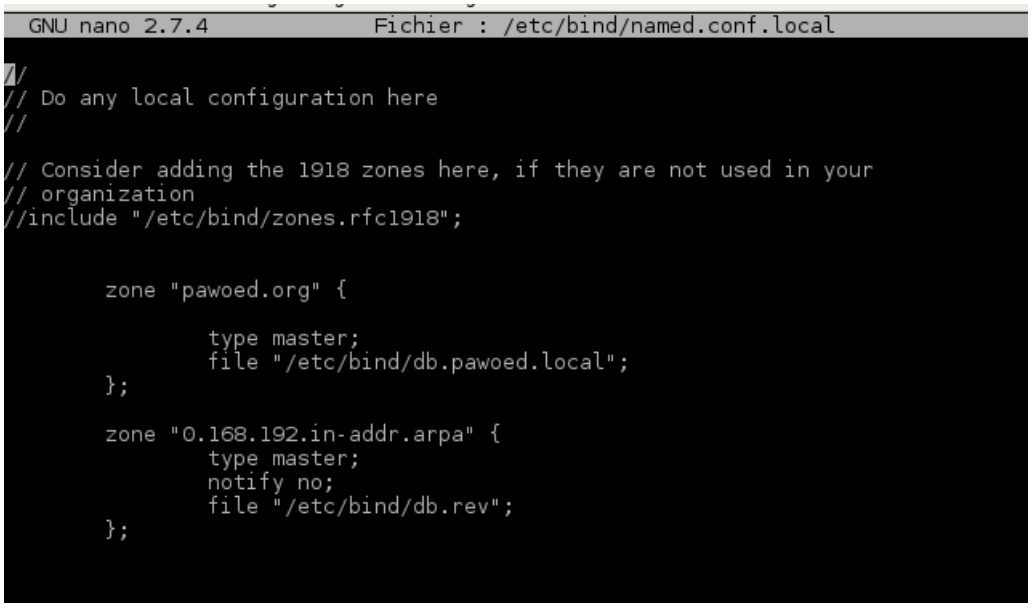


Toutes les configurations DNS sont stockées dans le répertoire /etc/bind.

- **/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



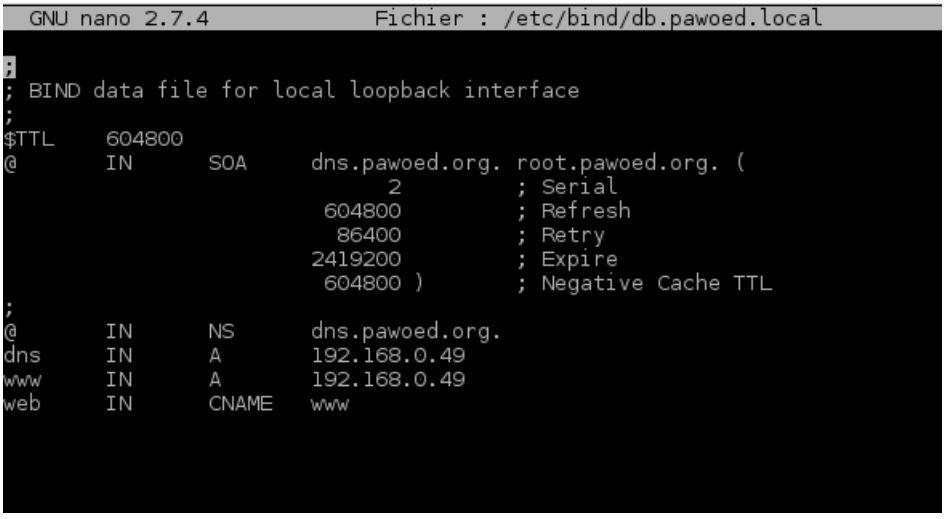
Redémarrer le service à l'aide de la commande

#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



Redémarrer le service à l'aide de la commande

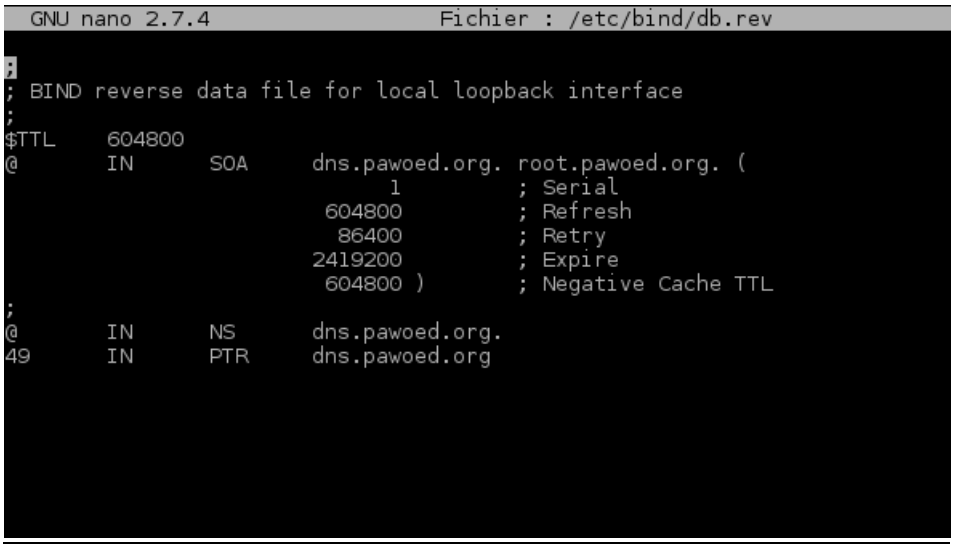
#service bind9 restart

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

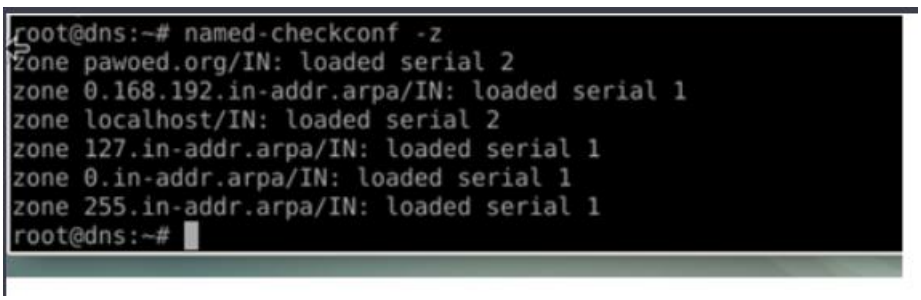
Le modifier comme suit :



Les commandes utiles

La commande ci-dessous vérifie le bon fonctionnement du service

#named-checkconf -z



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
49.0.168.192.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 :

Configuration des hôtes virtuels ou VirtualHost

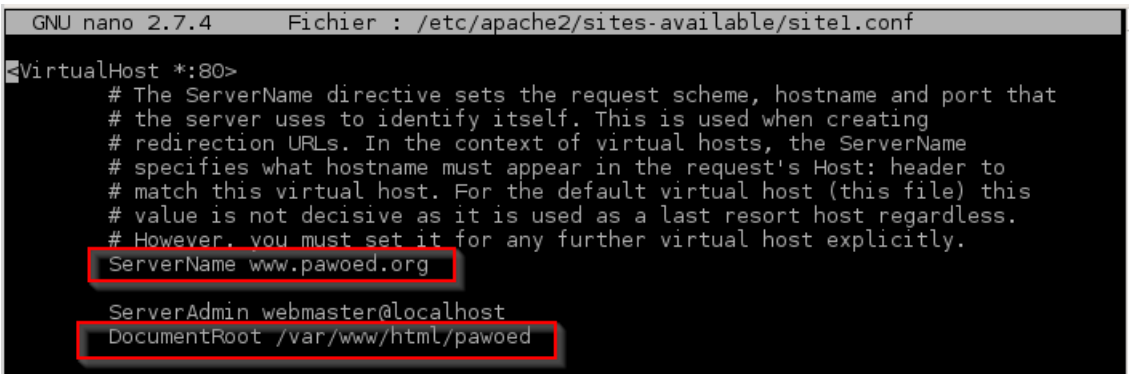
Chaque hôte est défini dans un fichier de configuration indépendant que l'on trouve et que l'on crée dans le répertoire « /etc/apache2/sites-available/ », dans ce répertoire se trouve le premier hôte virtuel « 000-default.conf »

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 :

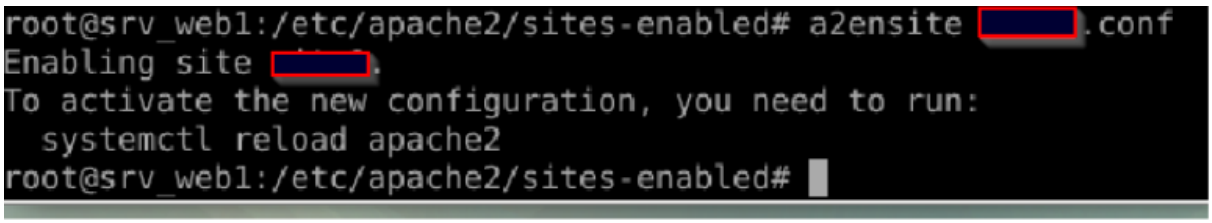


Après la configuration du fichier « /etc/apache2/sites-available/site1.conf », activer l'hôte virtuel en créant un lien entre le répertoire « /etc/apache2/sites-available »(répertoire des sites disponibles) et le répertoire « /etc/apache2/sites-enabled » (sites activés)

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

#a2nsite site1.conf

Vous devez voir apparaitre un message comme ci-dessous :



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


www.pawoed.org
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-----
MIICXgIBAAKBgQDQG9wvnuLC4aqzaJCAWGA1AxFzg00hjPObhq1mukzsGyuuWBFG
vj/k9vFNYX55DHctb/4cXtsZRWWvgcjtYnCVwRu+DAjFsk//kOMfhplmiv9xQ+ZL
8w/Xrnm8JWdSS+S4LCMnsuIiQtLbhMrQnUV02hAtbIZiSM3k6OjShEZhDQIDAQAB
AoGAHi0cBW+1k+qjFPbBlUq7UJSMUEKmyYmlvVSPCklTZB0gfVxZzPdDTpEcNks/
yo+rLFSD9Vsvy/9LGmLoXruadWlK67PCUnpM5/oRRGgy8t73YKrxflAU5Gtymjvc
ZCf0CAs6wBft3yLU31Qc4WqVM2vTyUH76jebVhxEw8k63OUCQQD/1OmAXV+TxBPG
ZTPFbzUeAE5rQqqOW4aoMNvM61Yn/19h6SzY2MfSQvF1BNns/efCRrqOMeyvPWUG
g1okfogTAkEA0D7pDf/D2Yu5msbOAGF4QBU1erLzpi/s6Rv6VEPYCGnHQlo3jbg9
FZbjHJ4UcYyYaA8jIrkY+FIJM88YlGbWXwJBAILEdvJ5R/CFCkKf2j2yIWmLaIol
En8fw43XI5L0PB7Hxx6KDLVu4XzVYQyahTZBdqR0eMlUNZJBhJE2tO3wi2cCQQCp
JkCFd3es0BrNxqfzlThozRFofcz88za7TldydL0YcFtC4Sb4vWsYizwktZ6jcPEm
rQz8Gl9W7MO+ynwLptB/AkEA1tsnFXoYzI71enmTdugGxbv0RqAd5iQpDYQkDSdn
2LImp/3YnXNJ9qpY91j87tKthh/Oetu6SHlmLg1LOYNIdw==
-----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 Autority)**

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
BAgTDENvcnNlIGR1IFN1ZDEQMA4GA1UEBxMHQWphY2NpbzEMMAoGA1UEChMDTExC
MREwDwYDVQQLEwhCVFMgSU5GTzEbMBkGA1UEAxMSc2VydmV1ci5idHNpbmZvLmZy
MB4XDTA0MDIwODE2MjQyNloXDTA0MDMwOTE2MjQyNlowcTELMAkGA1UEBhMCRlIx
FTATBgNVBAgTDENvcnNlIGR1IFN1ZDEQMA4GA1UEBxMHQWphY2NpbzEMMAoGA1UE
ChMDTExCMREwDwYDVQQLEwhCVFMgSU5GTzEYMBYGA1UEAxMPcHJvZi5idHNpbmZv
LmZyMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDSUagxPSv3LtgDV5sygt12
kSbN/NWP0QUiPlksOkF2NkPfwW/mf55dD1hSndlOM/5kLbSBo5ieE3TgikF0Iktj
BWm5xSqewM5QDYzXFt031DrPX63Fvo+tCKTQoVItdEuJPMahVsXnDyYHeUURRWLW
wc0BzEgFZGGw7wiMF6wt5QIDAQABMA0GCSqGSIb3DQEBBAUAA4GBALD640iwKPMf
pqdYtfvmLnA7CiEuao60i/pzVJE2LIXXXbwYjNAM+7Lov+dFT+b5FcOUGqLymSG3
kSK6OOauBHItgiGI7C87u4EJaHDvGIUxHxQQGsUM0SCIIVGK7Lwm+8e9I2X0G2GP
9t/rrbdGzXXOCl3up99naL5XAzCIp6r5
-----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 Mozzila :

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 ficher /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 :

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

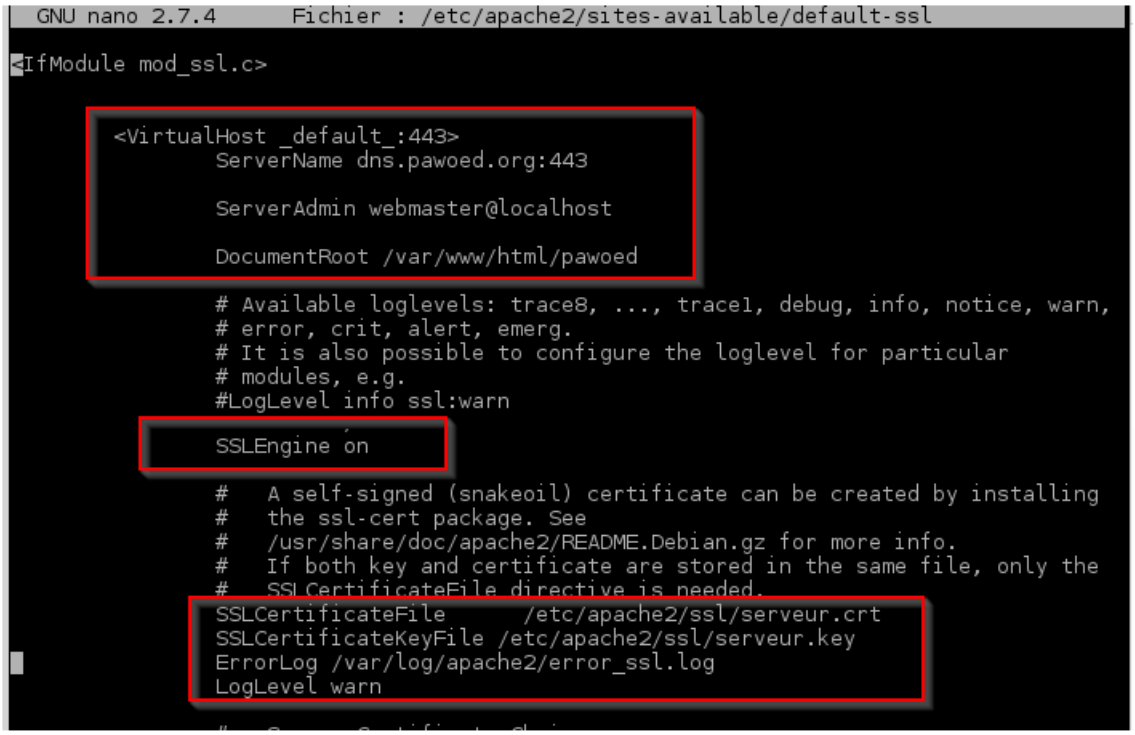
Servername dns.pawoed.org :443

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



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