


Documentation Serveur Slam

Installation de openssl :

 PROJET Slam&Sisr [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

```
root@projet-ss:/# apt-get install openssl
```

Pas à pas Apache2 :

Fichier Machine Écran Entrée Périphériques Aide

```
root@projet-ss:/# apt-get install apache2_
```

Ensuite il faut aller dans **nano/etc/apache2/apache2.conf** :

```
<Directory /var/www/html>
```

```
AllowOverride All
```

```
</Directory>
```

```
Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/html>
AllowOverride All
</directory>
```

On installe PHP :

```

root@projet-ss:/# apt-get install php
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common php7.4-json php7.4-op
  php7.4-readline
Paquets suggérés :
  php-pear
Les NOUVEAUX paquets suivants seront installés :
  libapache2-mod-php7.4 php php-common php7.4 php7.4-cli php7.4-common php7.4-json php7
  php7.4-readline
0 mis à jour, 9 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 4 021 ko dans les archives.
Après cette opération, 18,0 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O
Réception de :1 http://fr.archive.ubuntu.com/ubuntu focal/main amd64 php-common all 2:7
Réception de :2 http://fr.archive.ubuntu.com/ubuntu focal-updates/main amd64 php7.4-com
.3-4ubuntu2.8 [981 kB]
2% [2 php7.4-common 0 B/981 kB 0%]_

```

Ensuite nous rentrons la commande **apt-get install -y php7.4-pdo php7.4-mysql**

Nous allons créer un répertoire certificate : **on fait mkdir/etc/apache2/certificate**

Ensuite nous rentrons dedans **cd /etc/apache2/certificate**

Puis pour finir on fait : **openssl req -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache.key**

Dans common name il faut mettre NOTRE adresse ip !
(Pour savoir votre adresse ip mettez ip addr)

Pour continuer nous allons activer le mod ssl et rewrite en faisant **a2enmod ssl** et **a2enmod rewrite**

Et nous allons modifier le fichier de configuration Apache pour le site Web par défaut :

```

Organizational Unit Name (eg, section) []:SISR et SLAM
Common Name (e.g. server FQDN or YOUR name) []:10.0.2.15
Email Address []:maxence.pau.jr@gmail.com
root@projet-ss:/etc/apache2/certificate# ls
apache-certificate.crt  apache.key
root@projet-ss:/etc/apache2/certificate# nano /etc/apache2/sites-enabled/000-default.conf

```

SSL Engine on

SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt

SSLCertificateKeyFile /etc/apache2/certificate/apache.key

```
# However, you must set it for any particular virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
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

    SSL Engine on
    SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
    SSLCertificateKeyFile /etc/apache2/certificate/apache.key
    # 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
/VirtualHost>
```

Ensuite nous allons modifier notre virtualhost en mettant en 443 pour l'HTTPS

Et en ajoutant une section VirtualHost en 80 avec

```
ServerName "10.0.2.15"
```

```
Redirect permanent / https://10.0.2.15/
```

RewriteEngine On

```
RewriteCond %{HTTPS} !=on
```

```
RewriteRule ^/?(.*) https://%{10.0.2.15} /$1 [R=301,L]
```

```
Redirect permanent / https://10.0.2.15/
```

```
<VirtualHost *:80>
```

```
ServerName www.domaine.com
```

```
Redirect permanent / https://www.domaine.com/
```

```
</VirtualHost>
```

```
GNU nano 4.8 /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
    RewriteEngine on
    RewriteCond %{HTTPS} !=on
    RewriteRule ^/?(.*) https://%{10.0.2.15}/$1 [R=301,L]
    Redirect permanent / https://10.0.2.15/
</VirtualHost>

<VirtualHost *:443>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

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

    SSL Engine on
    SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
    SSLCertificateKeyFile /etc/apache2/certificate/apache.key
    # 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
</VirtualHost>

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

Après avoir éditer le fichier il faut redémarrer :
systemctl restart apache2

Installation de FTP :

Pour commencer faite :
apt-get install proftpd

On va autoriser le port 21 :
ufw allow 21

Ensuite on va dans le fichier proftpd :
nano /etc/proftpd/proftpd.conf

Et on reproduit comme sur l'image suivante :

```
GNU nano 4.8                                proftpd.conf
#
# /etc/proftpd/proftpd.conf -- This is a basic ProFTPD configuration file.
# To really apply changes, reload proftpd after modifications, if
# it runs in daemon mode. It is not required in inetd/xinetd mode.
#
# Includes DSO modules
Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.
UseIPv6                                on
# If set on you can experience a longer connection delay in many cases.
IdentLookups                            off

ServerName                               "PROJET Slam&Sisr"
# Set to inetd only if you would run proftpd by inetd/xinetd.
# Read README.Debian for more information on proper configuration.
ServerType                               standalone
DeferWelcome                             off

MultilineRFC2228                         on
DefaultServer                             on
```

Et suite à ça nous pouvons créer des utilisateurs grâce à la commande suivante :
adduser (nom)

Installation d'un chiffrement SSL :

On créer une clé publique :

openssl req -x509 -newkey rsa:2048 -keyout /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt -nodes -days 365

Ensuite on change les permissions du certificat en 600

sudo chmod 600 /etc/ssl/certs/proftpd.crt
sudo chmod 600 /etc/ssl/private/proftpd.key

Et on uncomment l'include **/etc/proftpd/proftpd.conf**

```
# for more information.
#
Include /etc/proftpd/tls.conf
```

Après nous allons dans le fichier config tls.conf (**sudo nano /etc/proftpd/tls.conf**)

Et on enlève le # devant les lignes suivante :

```
TLSEngine          on
TLSLog             /var/log/proftpd/tls.log
TLSProtocol        SSLv23
TLRSACertificateFile /etc/ssl/certs/proftpd.crt
TLRSACertificateKeyFile /etc/ssl/private/proftpd.key
TLSOptions         NoCertRequest EnableDiags NoSessionReuseRequired
TLSVerifyClient    off
TLSRequired        on
```

Ce qui donnerait ceci :

```
# See http://www.castaglia.org/proftpd/doc/contrib/ProFTPd-mini-HOWTO-TLS.html
# for more information.
#
<IfModule mod_tls.c>
TLSEngine          on
TLSLog             /var/log/proftpd/tls.log
TLSProtocol        SSLv23
#
# Server SSL certificate. You can generate a self-signed certificate using
# a command like:
#
# openssl req -x509 -newkey rsa:1024 \
#             -keyout /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt \
#             -nodes -days 365
#
# The proftpd.key file must be readable by root only. The other file can be
# readable by anyone.
#
# chmod 0600 /etc/ssl/private/proftpd.key
# chmod 0640 /etc/ssl/private/proftpd.key
#
TLRSACertificateFile /etc/ssl/certs/proftpd.crt
TLRSACertificateKeyFile /etc/ssl/private/proftpd.key
#
# CA the server trusts...
# TLSCACertificateFile /etc/ssl/certs/CA.pem
# ...or avoid CA cert and be verbose
# TLSOptions          NoCertRequest EnableDiags
# ... or the same with relaxed session use for some clients (e.g. FireFtp)
TLSOptions         NoCertRequest EnableDiags NoSessionReuseRequired
#
#
# Per default drop connection if client tries to start a renegotiate
# This is a fix for CVE-2009-3555 but could break some clients.
#
# TLSOptions          AllowClientRenegotiations
#
# Authenticate clients that want to use FTP over TLS?
#
TLSVerifyClient    off
#
# Are clients required to use FTP over TLS when talking to this server?
#
TLSRequired        on
```

Et ensuite il faut faire **sudo systemctl restart proftpd** et connecter vous, normalement vous aurez le certificat qui devrait s'afficher.

Installation de SSH :

On commence par faire:

apt install openssh-server

```
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  ncurses-term openssh-sftp-server ssh-import-id
Paquets suggérés :
  molly-guard monkeysphere ssh-askpass
Les NOUVEAUX paquets suivants seront installés :
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 mis à jour, 4 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 688 ko dans les archives.
Après cette opération, 6 010 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O
```

Et on met O (pour oui)

Pour voir si il est bien en marche :

systemctl status ssh

Il devrait normalement y avoir (si il s'affiche en vert c'est que c'est bon) :

```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-02-26 22:39:24 UTC; 5min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1682 (sshd)
    Tasks: 1 (limit: 3394)
   Memory: 2.3M
   CGroup: /system.slice/ssh.service
           └─1682 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

févr. 26 22:39:24 projet-ss systemd[1]: Starting OpenBSD Secure Shell server...
févr. 26 22:39:24 projet-ss sshd[1682]: Server listening on 0.0.0.0 port 22.
févr. 26 22:39:24 projet-ss sshd[1682]: Server listening on :: port 22.
févr. 26 22:39:24 projet-ss systemd[1]: Started OpenBSD Secure Shell server.
root@projet-ss:/etc/proftpd#
```

On ouvre la connexion SSH et on fait :

sudo ufw allow ssh

Création clé privé :

On crée la clé :

ssh-keygen -t rsa

Ce qui donne :

```
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256: jJSp0vPA/u1EU615zyTR2Js2A2wpABqVCsSC9K51jCE root@projet-ss
The key's randomart image is:
+---[RSA 3072]-----+
|
|*+00.
|++0 . + *
|Eo . = X o
|o = o o o o
|. = . * S B
|.+. . o * o
|. * . o
|. = o
|. .o.o
+----[SHA256]-----+
```

Et pour terminer on ajoute la clé à l'utilisateur

ssh-id-copy slam@(ip)

Et on lui met un mot de passe

Installation de la partie client :

On tape la commande suivante :
apt install openssh-client

Puis on test si ça fonctionne en se connectant :
ssh slam@(ip)
(il faut également entrer le mot de passe mit précédemment)

Installation de mariaDB :

On commence par faire :
apt install mariadb-server

Et encore une fois, pour voir qu'il est bien en marche :

```
● mariadb.service - MariaDB 10.3.32 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-02-27 09:16:15 UTC; 20s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 2206 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 31 (limit: 3394)
    Memory: 66.4M
   CGroup: /system.slice/mariadb.service
           └─2206 /usr/sbin/mysqld

févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: information_schema
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: mysql
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: performance_schema
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: Phase 6/7: Checking and upgrading tables
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: Processing databases
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: information_schema
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: performance_schema
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: Phase 7/7: Running 'FLUSH PRIVILEGES'
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2250]: OK
févr. 27 09:16:16 projet-ss /etc/mysql/debian-start[2340]: Triggering myisam-recover for all MyISAM tables
lines 1-22/22 (END)
```

Paramétrage du proxy :

On va dans :

```
cd /etc/apt
```

Ensuite on créer le fichier suivant :

```
apt.conf
```

Puis on va dedans :

```
nano apt.conf
```

Et on met ceci dedans :

```
acquire::http::proxy "http://10.1.2.5:8080/";  
acquire::https::proxy "https://10.1.2.5:8080/";  
acquire::ftp::proxy "http://10.1.2.5:8080/";
```