|  |  |
| --- | --- |
| Ftp Server APK 1.32 Download for Android – Download Ftp Server APK Latest  Version - APKFab.com | **Sécuriser vsFTPd** |

# Objectif

FTP est, par défaut, un protocole peu sécurisé.  
Tous les mots de passe et toutes les données sont transférés en clair.   
En utilisant TLS, toute la communication est chiffée, sécurisant ainsi le protocole FTP.

Ce tutoriel propose de créer un espace ftp l'emplacement disque /var/www/blogs.

# Installation de vsftpd et openSSL

Pour sécuriser vsFTPd, il faut installer les paquets vsftpd et openssl.  
Ces deux paquets peuvent être téléchargés depuis le référentiel de Debian.

Pour installer les paquets …  
**>> sudo apt -y install vsftpd openssl**

# Sauvegarde du fichier de configuration

Il est important de créer une sauvegarde du fichier de configuration /etc/vsftpd.conf, car en cas de problème cela peut toujours s’avérer utile ...  
**>> sudo cp --interactive /etc/vsftpd.conf /etc/vsftpd.conf\_original**

# Création du certificat SSL pour TLS

Pour utiliser TLS, on doit créer un certificat SSL.   
Ce certificat sera stocké dans le répertoire /etc/ssl-vsftpd/private.  
Ce répertoire n'existe pas et il faut le créer …  
**>> sudo mkdir --verbose --parents /etc/ssl-vsftpd/private**  
**>> sudo chmod --verbose 700 /etc/ssl-vsftpd/private**

Ensuite, il faut générer le certificat SSL …  
**>> sudo openssl req -x509 \  
 -nodes -days 365 -newkey rsa:1024 \  
 -keyout /etc/ssl-vsftpd/private/vsftpd.pem \  
 -out /etc/ssl-vsftpd/private/vsftpd.pem**

On répond aux questions d’onpenSSL …  
**Generating a RSA private key  
..........+++++  
.....................................+++++  
writing new private key to '/etc/ssl-vsftpd/private/vsftpd.pem'  
-----  
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]:CA  
State or Province Name (full name) [Some-State]:Quebec  
Locality Name (eg, city) []:Montreal  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:M9  
Organizational Unit Name (eg, section) []:Formation  
Common Name (e.g. server FQDN or YOUR name) []:srvftp.m9.info  
Email Address []:tux@m9.info**

# Activation de TLS

Afin de paramétrer vsFTPd pour qu’il soit sécurisé, on doit ajouter ou modifier les directives suivantes …  
**>> sudo nano /etc/vsftpd.conf**

|  |  |
| --- | --- |
| Directive | Fonction |
| ssl\_enable=YES | Activation du protocole SSL |
| allow\_anon\_ssl=YES | Autorisation des utilisateurs anonymes à utiliser pour les connexions SSL sécurisées |
| force\_local\_data\_ssl=YES | Obligation que toutes les connexions non anonymes utilise une connexion SSL sécurisée afin d’envoyer et de recevoir des données |
| force\_local\_logins\_ssl=YES | Obligation que toutes les connexions non anonymes utilise une connexion SSL sécurisée afin d’envoyer le mot de passe |
| ssl\_tlsv1=YES | Autorisation des connexions de protocole TLS v1 Les connexions TLS v1 sont préférées |
| ssl\_sslv2=NO | Autorisation des connexions de protocole TLS v2 Les connexions TLS v1 sont préférées |
| ssl\_sslv3=NO | Autorisation des connexions de protocole TLS v3 Les connexions TLS v1 sont préférées |
| require\_ssl\_reuse=NO | Désactivation de la réutilisation de session SSL (requise par WinSCP) |
| ssl\_ciphers=HIGH | Sélection du chiffrement SSL permettant les connexions chiffrées (requis par FileZilla) |
| rsa\_cert\_file=/etc/ssl-vsftpd/private/vsftpd.pem | Spécification de l’emplacement du certificat RSA à utiliser pour les connexions chiffrées |

Si on utilise la directive **force\_local\_logins\_ssl=YES** et **force\_local\_data\_ssl=YES**, seules les connexions TLS sont autorisées (ceci verrouille tous les utilisateurs avec d'anciens clients FTP qui ne sont pas pris en charge par TLS).  
En utilisant **force\_local\_logins\_ssl=NO** et **force\_local\_data\_ssl=NO**, les connexions TLS et non-TLS sont autorisées, en fonction de la prise en charge du client FTP.

On doit également s’assurer de préciser les paramètres suivants dans le fichier /etc/vsftpd.conf pour permettre les connexions authentifiées (non anonymes) …  
Afin de paramétrer vsFTPd pour qu’il soit sécurisé, on doit ajouter ou modifier les directives suivantes …  
**>> sudo nano /etc/vsftpd.conf**

|  |  |
| --- | --- |
| Directive | Fonction |
| local\_enable=YES | Autorisation afin de permettre aux utilisateurs locaux de se connecter |
| write\_enable=YES | Activation de n’importe quelle forme de commande d’écriture FTP |
| local\_umask=YES | Obligation que toutes les connexions non anonymes utilise une connexion SSL sécurisée afin d’envoyer et de recevoir des données |
| force\_local\_logins\_ssl=022 | Spécification de 077 pour l’umask par défaut pour les utilisateurs locaux  on peut modifier cette valeur pour 022, si les utilisateurs s’attendent à ce que  (022 est utilisé par la plupart des autres serveurs FTP) |

On doit redémarrer le service vsftpd …  
**>> sudo systemctl restart vsftpd.service**

# Ajout d’un utilisateur pour vsftpd

On ajoute un utilisateur Linux local auquel on pourra se connecter.   
On ajoute l’utilisateur **tux** avec le mot de passe **MonMotDePasse**.   
Cet utilisateur fera partie du groupe **usagers\_sftp**.

On ajoute un groupe d'utilisateur …  
**>> sudo groupadd usagers\_sftp**

On ajoute un utilisateur …  
**>> sudo adduser tux**

On ajoute l’utilisateur **tux** au groupe **usagers\_sftp** et on lui assigne le répertoire par défaut …  
**>> sudo usermod --gid usagers\_sftp tux  
>> sudo usermod --home /var/ftp tux**

Il est maintenant possible de se connecter au serveur vsFTPd.

Le serveur affiche une alerte comme quoi le certificat est auto-signé.   
Cela est normal.

Il est désormais possible d’avoir un accès FTP au serveur.  
Toutefois, on doit limiter accès à l’arborescence du serveur.

# Emprisonnement (chroot) d’un utilisateur

Afin de limiter l'accès de l'utilisateur à son répertoire et qu'il ne puisse pas accéder aux répertoires parents …  
**>> sudo nano /etc/vsftpd.conf**

On modifie la directive …  
**chroot\_local\_user=YES**

De plus, il faut emprisonner (chrooter) le groupe de l’utilisateur en modifiant la directive appropriée dans le fichier /etc/ssh/sshd\_config …  
**>> sudo nano /etc/ssh/sshd\_config**

On doit commenter la directive …  
**#Subsystem sftp /usr/libexec/openssh/sftp-server**et on ajoute …  
**Subsystem sftp internal-sftp  
Match Group usagers\_sftp  
X11Forwarding no  
AllowTcpForwarding no  
ChrootDirectory %h  
ForceCommand internal-sftp**

**Remarque** …  
L’utilisateur défini èa la directive Match User doit se trouver après le UsePAM.   
Dans le cas contraire, on recevra une erreur lors de la relance du service ssh.

On redémarre le service SSH …  
**>> sudo systemctl restart sshd.service**

L’utilisateur tux peut maintenant se connecter à l’aide ssh au serveur, mais si la connexion SFTP échoue c'est parce que le répertoire /var/www/blogs doit appartenir à root et uniquement éditable par le propriétaire …  
**>> sudo chown --verbose root:root /var/www/blogs/  
>> sudo chmod 777 /var/www/blogs/**

# vsFTPd et iptables

Pour le mode de transfert FTP actif, on doit autoriser les connexions TCP entrantes sur le port 21 et sortantes sur le port 20

Pour le mode de transfert FTP passif, on doit autoriser les connexions TCP entrantes sur le port 21 et les connexions entrante sur des ports aléatoires.

Pour ce faire …  
**>> sudo nano /etc/vsftpd.conf  
pasv\_enable=YES  
pasv\_min\_port=13450  
pasv\_max\_port=13500**

Et d'ouvrir les ports dans iptables …  
**>> sudo iptables -t filter -A INPUT -p tcp --dport 13450:13500 -j ACCEPT  
>> sudo iptables -t filter -A OUTPUT -p tcp --dport 13450:13500 -j ACCEPT**