UNIVERSITE ABDELMALEK ESSAADI
FACULTE DES SCIENCES ET TECHNIQUES
DE TANGER

DEPARTEMENT GENIE INFORMATIQUE

Cycle Master : MBD et SIM (Semestre III)

Module : Cybersecurity

Rapport de Mini Projet :

Sécurisation du transfert de données entre deux serveurs Linux en utilisant le protocole FTP & FTPS (FTP over TLS)

Réalisé Par :

EL MSAOURI Tarik

MEDAGHRI ALAOUI Amine

Encadré Par :

Pr. EL ABDELKHALKI Jamal

Année Universitaire 2021/2022

Table des Matières

I. Introduction

Dans notre mini projet, nous avons décrit en détail comment installer et configurer un serveur FTP dans les serveurs Linux. Et nous expliquerons comment sécuriser un serveur FTP à l'aide de SSL/TLS pour activer les services de cryptage des données pour un transfert de fichiers sécurisé entre les systèmes.

D'abord on va essayer avec protocole FTP simple. Et après on va essayer avec protocole FTPs over TLS 1.2. Et on va tester le cryptage et la sécurité des données en utilisant **WireShark.**

FTP (File Transfer Protocol) est principalement utilisé pour transférer des fichiers entre ordinateurs. FTP fonctionne dans une architecture client-serveur, dans laquelle le client demande un fichier au serveur et le serveur renvoie le fichier requis au client. Sur la machine cliente, l'application cliente FTP est utilisée pour communiquer avec le serveur.

Par défaut, FTP communique sur un canal non sécurisé, mais il est possible de configurer FTP pour transférer des données sur un canal sécurisé. Dans ce mini projet, on va configurer un serveur FTP avec TLS.

II. Exigences

Pour pouvoir faire notre mini projet, il faudra suivre les exigences suivantes :

- 2 Serveurs Linux, pour notre cas, on a utilisé : Kali linux (serveur) et Ubuntu (Client) en tant que des machines virtuelles dans Hyperviseur type 2 VirtualBox.
- Install Ftp protocole dans des 2 serveurs.
- Wireshark pour analyser les données transférées.

III. Configuration coté Serveur (Kali linux)

1- Configuration le serveur en Actif Mode

En mode Actif, le client FTP démarre la session en établissant la connexion de contrôle TCP depuis n'importe quel port aléatoire de la machine cliente vers le port 21 du serveur. Ensuite, le client commence à écouter sur un port X aléatoire pour une connexion de données et informe le serveur via la connexion TCP Control que le client attend la connexion de données sur le port X. Après cela, le serveur établit une connexion de données de son port 20 à le port X sur la machine cliente.

Un problème peut survenir lorsque le client est derrière un pare-feu et que le port X est bloqué. Dans ce cas, le serveur n'est pas en mesure d'établir une connexion de données avec le client. Pour éviter ce problème, le serveur FTP est principalement utilisé en mode Passif. Par défaut, VSFTPD utilise le mode passif, nous devrons donc le changer en mode actif.

VSFTPD est un programme utilisé pour configurer FTP sur un serveur.

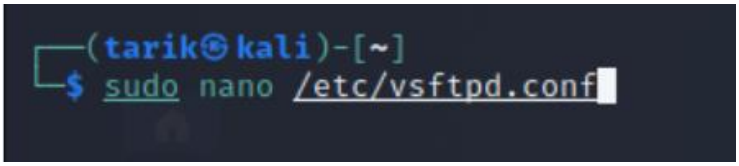
Nous l'installons à l'aide de la commande suivante :

```
┌──(tarik㉿kali)-[~]
└─$ sudo apt-get install vsftpd
[sudo] password for tarik:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
vsftpd is already the newest version (3.0.3-13).
0 upgraded, 0 newly installed, 0 to remove and 744 not upgraded.
```

Puis on commence le programme.

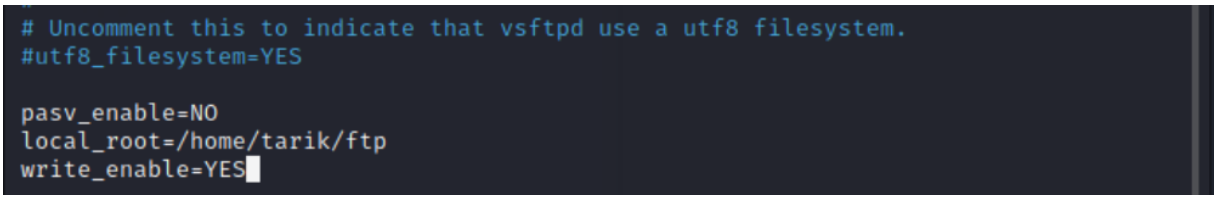
```
┌──(tarik㉿kali)-[~]
└─$ sudo systemctl start vsftpd                                    130 ×

┌──(tarik㉿kali)-[~]
└─$ systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/lib/systemd/system/vsftpd.service; disabled; vendor preset:>
     Active: active (running) since Sat 2021-11-13 07:44:42 EST; 3s ago
    Process: 9936 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, >
   Main PID: 9937 (vsftpd)
      Tasks: 1 (limit: 4633)
     Memory: 868.0K
        CPU: 11ms
     CGroup: /system.slice/vsftpd.service
             └─9937 /usr/sbin/vsftpd /etc/vsftpd.conf

Nov 13 07:44:42 kali systemd[1]: Starting vsftpd FTP server ...
Nov 13 07:44:42 kali systemd[1]: Started vsftpd FTP server.
```

Tout d'abord, ouvrez le fichier de configuration VSFTPD.

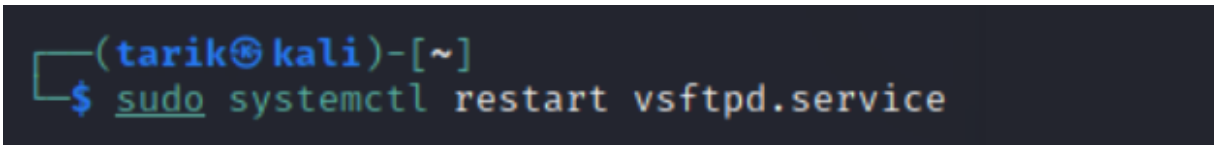


On ajoute les lignes suivantes à la fin du fichier.

L'option **'write_enable'** doit être activée pour permettre aux utilisateurs d'écrire sur le serveur FTP.



Enfin, enregistrez le fichier et fermez-le. Redémarrez ensuite le service VSFTPD :



Ensuite, créez un répertoire que le serveur FTP utilisera pour stocker les fichiers. Nous allons configurer **'/home/tarik/ftp/'** comme chemin racine pour le serveur FTP.

Et nous spécifions déjà ce répertoire dans le fichier de configuration en modifiant l'option **'local_root'**. Cela configurera le chemin racine du serveur.

Le serveur est configuré en mode actif

2- Configuration du pare-feu pour le mode actif

Si FTP est utilisé en mode actif, le serveur FTP utilisera deux ports pour communiquer avec le client, les ports 21 et 20. Le port 21 est utilisé pour transmettre des commandes au client et le port 20 est utilisé pour transférer des données vers n'importe quel port aléatoire du client. Nous utiliserons **ufw** pour configurer le pare-feu sur le serveur.

```
┌──(tarik㉿kali)-[~]
└─$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)

┌──(tarik㉿kali)-[~]
└─$ sudo ufw allow 20/tcp
Rule added
Rule added (v6)
```

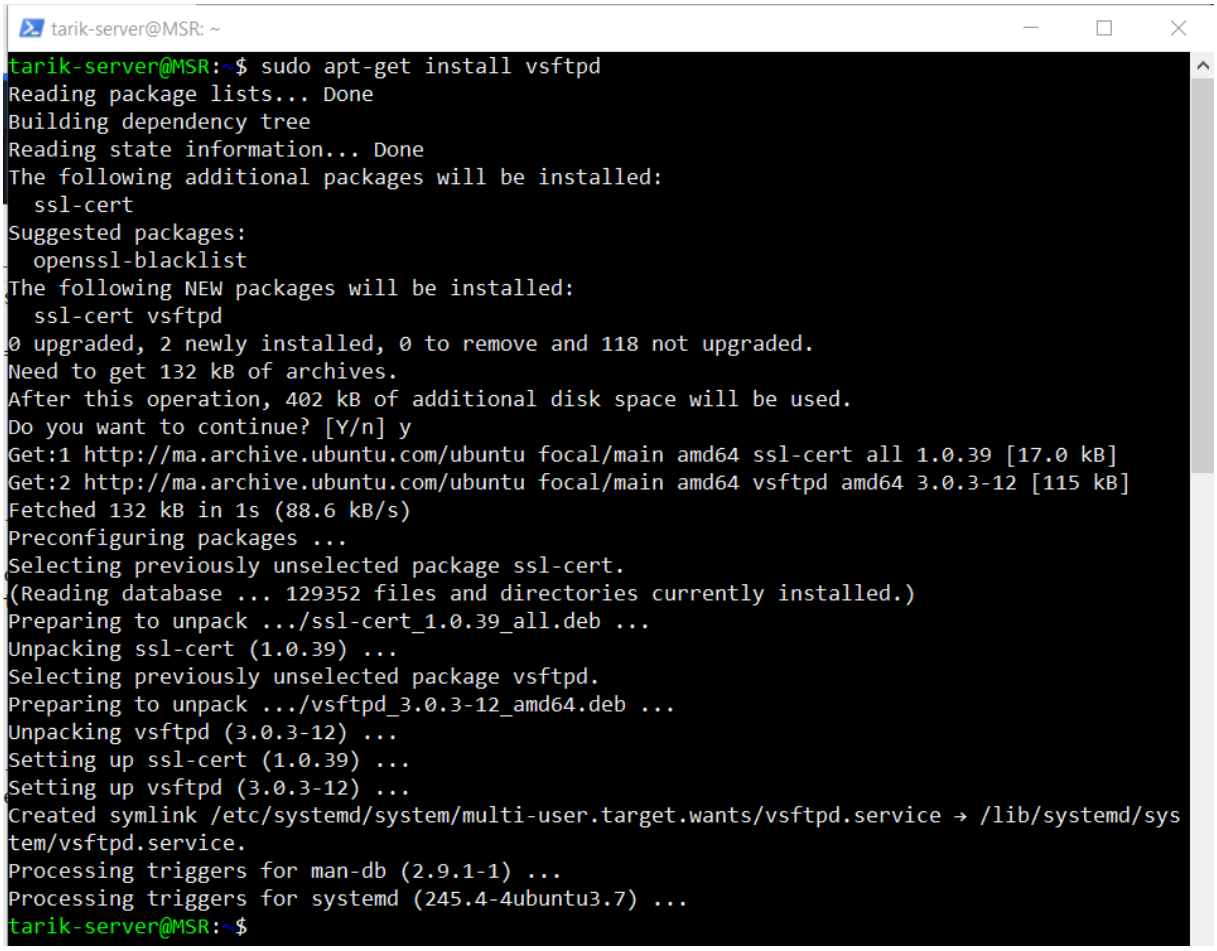
Ufw enable et vérifiez l'état de ufw à l'aide des commandes suivantes :

```
┌──(tarik㉿kali)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup

┌──(tarik㉿kali)-[~]
└─$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
5432                       ALLOW       Anywhere
5432/tcp                   ALLOW       Anywhere
443                        ALLOW       Anywhere
21                         ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
20/tcp                     ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
5432 (v6)                  ALLOW       Anywhere (v6)
5432/tcp (v6)              ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
21 (v6)                    ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
20/tcp (v6)                ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)
```

III. Configuration coté Client (Ubuntu)

La machine Ubuntu 20.04 qui a le nom de tarik-server.

On installe le service vsftpd.




```
tarik-server@MSR:~$ sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2021-11-13 12:30:53 UTC; 2min 52s ago
   Main PID: 3384 (vsftpd)
      Tasks: 1 (limit: 2250)
     Memory: 588.0K
     CGroup: /system.slice/vsftpd.service
             └─3384 /usr/sbin/vsftpd /etc/vsftpd.conf

Nov 13 12:30:53 MSR systemd[1]: Starting vsftpd FTP server...
Nov 13 12:30:53 MSR systemd[1]: Started vsftpd FTP server.
```

On configure le pare-feu comme la cote du serveur.

```
tarik-server@MSR:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)

tarik-server@MSR:~$ sudo ufw allow 21/tcp
Rule added
Rule added (v6)
tarik-server@MSR:~$ sudo ufw allow 20/tcp
Rule added
Rule added (v6)
```

Après avoir autorisé tous les ports sur le pare-feu, on active ufw en exécutant la commande suivante.

```
tarik-server@MSR:~$ sudo ufw enable
[sudo] password for tarik-server:
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

```
tarik-server@MSR:~$ sudo ufw status
[sudo] password for tarik-server:
Status: active

To                         Action      From
--                         ------      ----
20/tcp                     ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
20/tcp                     ALLOW       192.168.100.234
20/tcp (v6)                ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
```

IV. Test de la connexion et la transmission des fichiers en utilisant FTP

Maintenant, notre serveur est configuré en mode actif, et nous pouvons y accéder du côté client. On peut utiliser une application client comme Filezilla ou on peut y accéder d'après ligne des commandes.

Nb : l'adresse IP du notre serveur est : 192.168.100.234

```
tarik-server@MSR: ~
tarik-server@MSR:~$ ftp 192.168.100.234
Connected to 192.168.100.234.
220 (vsFTPd 3.0.3)
Name (192.168.100.234:tarik-server): tarik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 1000     1000           16 Nov 13 12:08 test_kali.txt
226 Directory send OK.
```

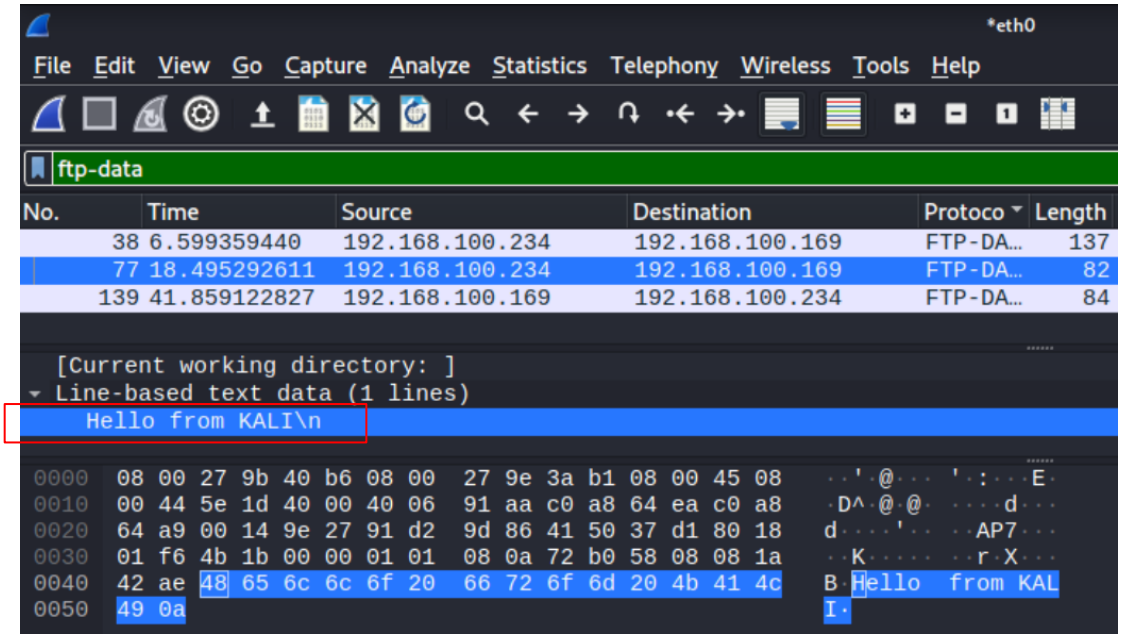
On va essayer quelques commandes de la transmission du fichier notamment réception et l'envoie d'un fichier.

```
ftp> get test_kali.txt
local: test_kali.txt remote: test_kali.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for test_kali.txt (16 bytes).
226 Transfer complete.
16 bytes received in 0.00 secs (190.5488 kB/s)
ftp>
ftp>
ftp>
ftp> put /home/tarik-server/FTP_ubuntu/test_ubuntu.text test_ubuntu.txt
local: /home/tarik-server/FTP_ubuntu/test_ubuntu.text remote: test_ubuntu.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
18 bytes sent in 0.00 secs (390.6250 kB/s)
ftp>
```

Puis, On va essayer de détecter le contenue d'un fichier en utilisant Wireshark.

- ## Tester la sécurité du protocole FTP sans Sécurisé

On peut remarquer facilement le contenue du fichier transmis en utilisant un programme d'analyse de données.



V. La transmission des fichiers en utilisant FTP over TLS 1.2

1- Génération du certificat SSL/TLS et de la clé privée dans le Serveur Kali

Par défaut, le serveur FTP établit la connexion entre le client et le serveur via un canal non sécurisé. Ce type de communication ne doit pas être utilisé si vous souhaitez partager des données sensibles entre le client et le serveur. Pour communiquer sur un canal sécurisé, il est nécessaire d'utiliser des certificats SSL.

Nous allons générer ces certificats en utilisant **openssl**. La commande suivante générera des certificats SSL pour notre serveur.

```
sudo openssl req -x509 -nodes -day 365 -newkey rsa:2048 -keyout
/etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

Lorsque nous exécutons la commande ci-dessus, nous aurons des questions. Après avoir répondu à ces questions, le certificat sera généré.

```
┌──(tarik㉿kali)-[~/Desktop]
└─$
sudo openssl req -x509 -nodes -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/priva
te/vsftpd.pem -days 365 -newkey rsa:2048
Generating a RSA private key
.....................................................................+++++
.....................................+++++
writing new private key to '/etc/ssl/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]:MA
string is too long, it needs to be no more than 2 bytes long
Country Name (2 letter code) [AU]:Morocco
string is too long, it needs to be no more than 2 bytes long
Country Name (2 letter code) [AU]:ma
State or Province Name (full name) [Some-State]:Tangier
Locality Name (eg, city) []:Tangier
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FST
Organizational Unit Name (eg, section) []:FST
Common Name (e.g. server FQDN or YOUR name) []:Tarik
Email Address []:tarik.elmsaouri@etu.uae.ac.ma
```

Nous pouvons vérifier les certificats dans le terminal.

```
┌──(tarik㉿kali)-[~]
└─$ sudo ls /etc/ssl/private/
ssl-cert-snakeoil.key  vsftpd.pem
```

2- Utilisation de certificats dans le fichier de configuration

Maintenant, nos certificats sont prêts à être utilisés. Nous allons configurer le fichier 'vsftpd.conf' pour utiliser les certificats SSL pour la communication. Ouvrez le fichier de configuration avec la commande suivante. *sudo nano /etc/vsftpd.conf*

Ajoutez les lignes suivantes à la fin des fichiers. Ces modifications garantiront que le serveur FTP utilise les certificats SSL nouvellement générés pour communiquer en toute sécurité avec le client**.**

```
ssl_enable=YES
force_local_data_ssl=NO
force_local_logins_ssl=NO
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

On a ajouté l'option **ssl_enable=YES** pour activer l'utilisation de SSL, encore une fois, car TLS est plus sécurisé que SSL, nous allons restreindre VSFTPD à utiliser TLS à la place, en activant l'option ssl_tlsv1.

Et nous ajoutons les lignes pour définir l'emplacement du certificat SSL et du fichier de clé :

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

```
# encrypted connections.
#    rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
#    rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES

pasv_enable=NO
local_root=/home/tarik/tarikftp
write_enable=YES

#configuration ssl

ssl_enable=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO

force_local_data_ssl=NO
force_local_logins_ssl=NO

rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

Enregistrez le fichier et fermez-le. Redémarrez ensuite le service VSFTPD

```
┌──(tarik㉿kali)-[~/Desktop]
└─$ sudo systemctl restart vsftpd
```

3- Vérifier FTP avec les connexions SSL/TLS

D'abord on installe dans le coté client l'utile **lftp** pour se connecter au serveur en utilisant SSL certificat.

```
tarik-server@MSR:~$ sudo apt-get install lftp
[sudo] password for tarik-server:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  lftp
```

On peut se connecter par deux façons, la méthode simple est de faire une seule commande :

```
lftp -c 'set ftp:ssl-allow true ; set ssl:verify-certificate no;set ftp:ssl-
protect-data true; open -u tarik,2020 -e "ls; quit" 192.168.1.9'
```

On a ajouté "**set ssl:verify-certificate no**" car le serveur utilise des certificats auto-signés.

```
tarik-server@MSR:~$ lftp -c 'set ftp:ssl-allow true ; set ssl:verify-certificate no;
open -u tarik,2020 -e "ls; quit" 192.168.1.9'
-rw-r--r--    1 1000     1000           16 Nov 13 12:08 test_kali.txt
-rw-------    1 1000     1000           18 Nov 13 12:10 test_ubuntu.txt
tarik-server@MSR:~$
```

Exemple de recevoir d'un fichier.

```
tarik-server@MSR: ~
tarik-server@MSR:~$ tarik-server@MSR:~$ lftp -c 'set ftp:ssl-allow true ; set ssl:verify-certificate no;
set ftp:ssl-protect-data true; open -u tarik,2020 -e "get test_ssl.txt;quit" 192.168.1.9'
tarik-server@MSR:~$ ls | grep ss
test_ssl.txt
tarik-server@MSR:~$
```

La deuxième méthode :

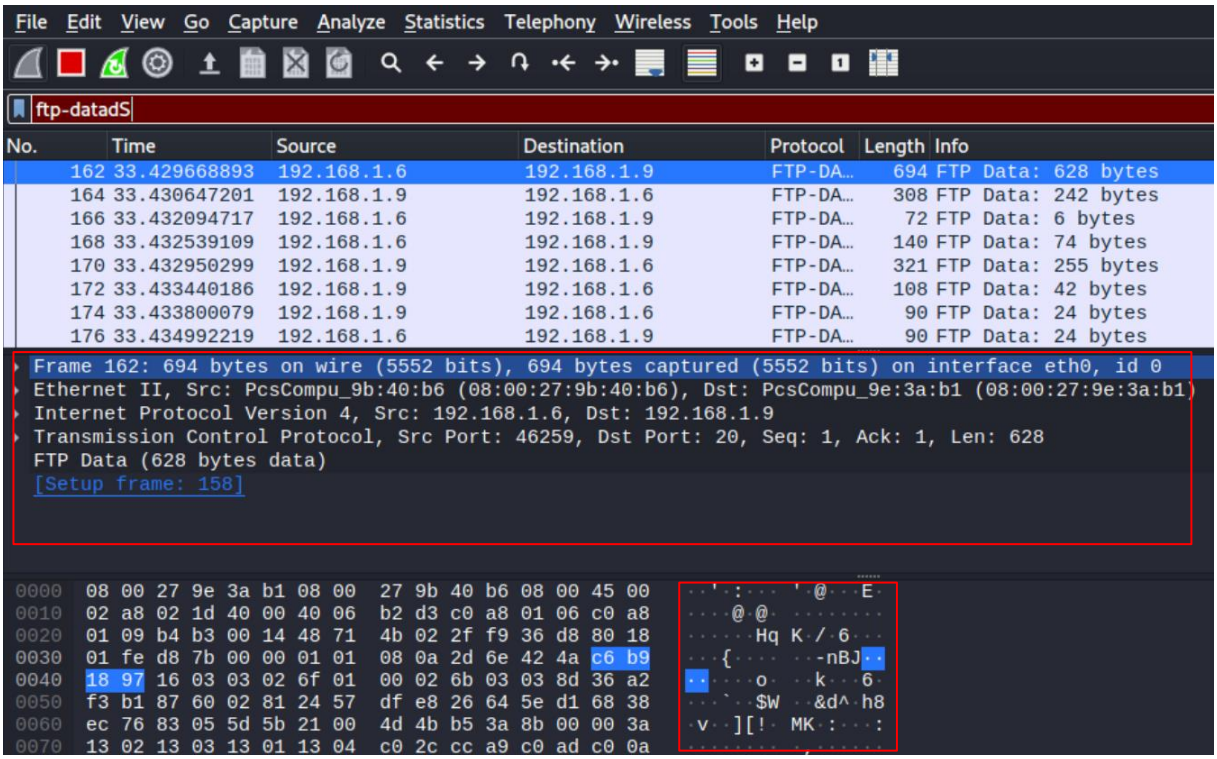
```
tarik-server@MSR: ~
tarik-server@MSR:~$ lftp tarik@192.168.100.234
Password:
lftp tarik@192.168.100.234:~> set ftp:ssl-allow true
lftp tarik@192.168.100.234:~> set ssl:verify-certificate no
lftp tarik@192.168.100.234:~> set ftp:ssl-protect-data true
lftp tarik@192.168.100.234:~>
```

```
.168.43.156:~>
lftp tarik@192.168.43.156:~> set ssl:ca-file "/etc/ssl/certs/ca-certificates.crt"
```

```
lftp tarik@192.168.100.234:~> ls
-rw-r--r--    1 1000     1000           16 Nov 13 12:08 test_kali.txt
-rw-r--r--    1 1000     1000           20 Nov 13 17:18 test_ssl.txt
-rw-------    1 1000     1000           18 Nov 13 12:10 test_ubuntu.txt
lftp tarik@192.168.100.234:~>
lftp tarik@192.168.100.234:~> get test_kali.txt
16 bytes transferred in 30 seconds
lftp tarik@192.168.100.234:~>
lftp tarik@192.168.100.234:~>
lftp tarik@192.168.100.234:~> put /home/tarik-server/FTP_ubuntu/ftp_ssl.txt
25 bytes transferred in 30 seconds
lftp tarik@192.168.100.234:~>
```

4- Tester la sécurité du protocole FTP Sécurisé par SSL/TLS

On peut remarquer que le contenue du fichier transmis n'est pas visible et donc sécurisés.



Conclusion

Le protocole de transfert de fichiers est utilisé depuis de nombreuses années pour transférer des fichiers et des documents sur Internet. VSFTPD est l'un des packages utilisés comme serveur FTP sur notre machine. VSFTPD contient diverses configurations que nous pouvons utiliser pour personnaliser votre serveur FTP.

Ce mini projet nous a montré comment configurer un serveur FTP avec TLS pour une sécurité renforcée.

Réferences

https://www.tecmint.com/secure-ftp-server-using-ssl-tls-on-ubuntu/

https://linuxhint.com/configure-ftp-tls-ubuntu/

https://fr.wikipedia.org/wiki/File_Transfer_Protocol