

# Instalación de FTP en CentOS 7

---

**Luigi Guarino**

**02/11/2017**

# Índice

1. Introducción.....	3
¿Qué es FTP?.....	3
¿Donde usamos FTP? .....	3
¿Y el FTP seguro? .....	3
¿Y entonces, como aseguramos nuestro servidor? .....	3
2. Objetivos.....	4
3. Instalación y configuración de FTP .....	4
1. Convertirse a administrador y actualizar los paquetes del sistema .....	4
2. Instalar FTP .....	4
3. Asignar dirección IP estática .....	5
4. Preparar el entorno para los usuarios anonimos .....	6
5. Usuarios del FTP .....	6
6. Configurar FTP.....	7
4. Pruebas de contexto .....	12
1. FTP por comando (Ubuntu).....	13
2. FTP mediante Filezilla y Chrome.....	20
5. FTP Seguro (FTPS) .....	22
1. Auditar nuestro servidor antes de asegurarlo .....	22
1. Asegurar FTP con SSL.....	24
3. Configurando firewall.....	25
4. Uso del FTPS.....	27
5. Segunda auditoración.....	29
6. Conclusión.....	30

# 1. Introducción

## ¿Qué es FTP?

FTP (en inglés **File Transfer Protocol** o Protocolo de Tránsito de Archivos), es el protocolo que permite el **intercambio de archivos** entre diferentes sistemas a través de la red.

Por tanto, un servidor FTP, es el encargado de permitir el intercambio de datos entre **cliente-servidor**, además de permitir **interactuar** con los ficheros ubicados en el servidor, pudiendo **editarlos, eliminarlos y/o modificarlos** según las **directivas** establecidas en el servidor.

Debemos saber que el protocolo FTP, es un protocolo TCP, que usa los **puertos 20 y 21**, y es ofrecido por la **capa de aplicación** ([Modelo OSI](#)). Además, FTP es un servicio **no seguro**, aunque existen **alternativas** (explicadas en este post), para introducir **seguridad** a nuestro servicio.

## ¿Dónde usamos FTP?

Implementamos un servidor de FTP, en cualquier **estructura de red** en la que sea necesario **almacenar** datos como, archivos, páginas web,... Y así poder disponer de ellas en cualquier momento a través de la red, y que además otros usuarios pueden tener **acceso** a ellas (si uno quiere, claro...).

## ¿Y el FTP seguro?

Como he comentado anteriormente, FTP se trata de un protocolo **no seguro**, ya que, está optimizado para rendir a la **máxima velocidad** posible y, por tanto, las **conexiones** realizadas al servidor se transmiten **sin cifrar**. Es decir, **toda la información** (incluida usuarios y contraseñas) enviada se encuentra en texto descifrado (**texto plano**) y esto posibilita que un atacante pudiera **capturar este tráfico** y acceder al servidor y/o a la información transmitida.

## ¿Y entonces, como aseguramos nuestro servidor?

Existen **dos alternativas** para **cifrar** nuestra conexión en FTP: **SFTP** y **FTPS**.

### SFTP

La característica principal de este protocolo es que añade el mismo protocolo FTP a **SSH**. SFTP utiliza las claves de SSH para cifrar y descifrar la autenticación y opera bajo el **puerto 22**.

### FTPS

Este, utiliza el **puerto 21** y los **certificados SSL** para autenticar la conexión. **Este protocolo será el que utilizamos para cifrar la conexión.**

## 2. Objetivos

- Instalar y configurar un servidor FTP estable y seguro
- Enjaular a los usuarios
- Habilitar conexiones anónimas de uso restringido
- Configurar diferentes permisos para diferentes usuarios
- Configurar seguridad y cifrada para la conexión

## 3. Instalación y configuración de FTP

### 1. Convertirse a administrador y actualizar los paquetes del sistema

Vamos a trabajar como administradores del sistema, por tanto, ejecutamos el comando **sudo su**

### 2. Instalar FTP

La herramienta de software para el servicio FTP que vamos a instalar será: **Vsftpd**. Para ello ejecutamos el comando: **yum install vsftpd**

Una vez instalado, se creara el directorio **vsftpd**, ubicado en **/etc**, donde se encontraran los ficheros de configuración del servicio:

```
[root@luigiftip ~]# ls /etc/vsftpd
ftusers  user_list  vsftpd.conf  vsftpd_conf_migrate.sh
[root@luigiftip ~]# _
```

### 3. Asignar dirección IP estática

Vamos a asignar una **dirección IP estática** a la **tarjeta de red** que ofrecerá el servicio **FTP**.

Ejecutamos **nano /etc/sysconfig/network-scripts/ifcfg-eth0**, donde “eth0” es nombre de la tarjeta de red mencionada (en nuestro caso “enp0s3”)

Editamos la línea **BOOTPROTO** y añadimos **IPADDR** Y **NETMASK**:

```
GNU nano 2.3.1 Fichero: ...sysconfig/network-scripts/ifcfg-enp0s3 Modificado
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6_INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s3
UUID=6a51654e-d1e0-47d7-868f-75595e9ed1d8
DEVICE=enp0s3
ONBOOT=yes
IPADDR=172.16.0.1
NETMASK=255.255.0.0_
```

Como vemos, utilizaremos la @ip **172.16.0.1/16** para el servidor. **No hemos configurado DNS ni Gateway**, para simplificar el proceso.

Por último, reiniciamos la interfaz: **ifdown enp0s3 → ifup enp0s3** y comprobamos la nueva configuración: **ip addr**

```
[root@luigiftip ~]# ifdown enp0s3
El dispositivo «enp0s3» fue desconectado correctamente.

[root@luigiftip ~]# ifup enp0s3
Conexión activada con éxito (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
[root@luigiftip ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN 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 qlen 1000
    link/ether 08:00:27:fa:02:ac brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.1/16 brd 172.16.255.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::fa2:cad3:dd78:93e9/64 scope link
        valid_lft forever preferred_lft forever
[root@luigiftip ~]# _
```

## 4. Preparar el entorno para los usuarios anónimos

Como podemos suponer, permitir el acceso a nuestro servidor **sin logeo**, es peligroso para la **seguridad** de nuestro sistema. Por tanto, nosotros seremos los encargados de **ubicar** correctamente a estos clientes y **restringirles** el uso del servicio.

Lo primero que ha haremos será crear un **directorio** donde trabajaran los anónimos. Este directorio debe estar localizado en una ruta **no crítica** para nuestro sistema. Por ejemplo, **/home**.

Por lo tanto, ejecutamos **mkdir /home/anonimos**

Una vez realizado, debemos ejecutar el siguiente comando para que CentOS **reconozca** este directorio como un **directorio de uso público**:

**chcon -R -t public\_content\_t /home/anonimo**

```
[root@luigiftftp /]# mkdir /home/anonimos
[root@luigiftftp /]# chcon -R -t public_content_t /home/anonimos
```

## 5. Usuarios del FTP

Además de logearnos como anónimo, nuestro FTP permitirá el acceso a través de usuario/password. Estos usuarios, tendrán **cuenta en el servidor** y estarán **ubicados** en el directorio **/home**.

En añadido, estos usuarios **usarán una shell fantasma**. Así **evitaremos** que puedan **acceder a la consola** de nuestros sistemas. Para la creación de este shell seguimos los siguientes pasos:

1. Creamos la shell: **nano /bin/ftp**

```
GNU nano 2.3.1          Fichero: /bin/ftp
#!bin/sh
echo "Usuario del servidor FTP de Luigi"
```

2. Añadimos nuestra nueva shell a la lista de shells del servidor. Para ello accedemos al fichero donde se encuentran las "shells" (**nano /etc/shells**) y añadimos la siguiente línea: **/bin/ftp**

```
GNU nano 2.3.1          Fichero: /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/bin/ftp
```

3. Por último, damos permisos de ejecución a todos los usuarios: **chmod a+x /bin/ftp**

*Importante: Algunos tutoriales hacen uso de la shell "nologin". Nosotros no emplearemos esta ya que existen **servicios del sistema** que ejecutan esta **shell** y, por lo tanto, **estaremos creando un grave problema de seguridad si la utilizamos**.*

Una vez creada la shell, añadimos nuestro **nuestros primeros usuarios** para el FTP, usando la siguiente comando: **useradd -g ftp -d /home/usuario -s /bin/ftp usuario**

```
[root@luigiftftp /]# useradd -g ftp -d /home/usu1 -s /bin/ftp usu1
[root@luigiftftp /]# useradd -g ftp -d /home/usu2 -s /bin/ftp usu2
[root@luigiftftp /]# useradd -g ftp -d /home/usu3 -s /bin/ftp usu3
[root@luigiftftp /]# _
```

Con el anterior comando, crearemos un **usuario**, que se añadirá al **grupo ftp**, y su **directorio** se encontraran dentro de **/ftp**. Además, usara la **shell fantasma** creada anteriormente "ftp".

Y creamos las contraseñas para los respectivos usuarios: **passwd usuario**

*Nota: Podemos comprobar que nuestros usuarios, únicamente puede iniciar sesión en FTP y en nuestro servidor NO:*

```
[root@luigiftftp /]# su usu1
Usuario del servidor FTP de Luigi
[root@luigiftftp /]# _
```

Nos queda indicarles al sistema que el **directorio de los usuarios** pueda ser **accesible** a través del **FTP**:

**setsebool -P allow\_ftpd\_full\_access on**

```
[root@luigiftftp /]# setsebool -P allow_ftpd_full_access on
```

## 6. Configurar FTP

Como buen administrador de sistemas, lo primero que haremos será realizar un **backup del fichero de configuración**, que se encuentra ubicado en **/etc/vsftpd**.

Para ello ejecutamos un **cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.backup**.

Además, instalamos la herramienta **nano**, para posteriormente editar el fichero:

```
[root@luigiftftp /]# cp etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.old
[root@luigiftftp /]# yum install nano_
```

Accedemos a él: **nano /etc/vsftpd/vsftpd.conf** :

```
GNU nano 2.3.1      Fichero: etc/vsftpd/vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
# When SELinux is enforcing check for SE bool ftp_home_dir
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
[ 127 líneas leídas ]
^G Ver ayuda  ^O Guardar   ^R Leer Fich ^Y Pág Ant  ^K CortarTxt ^C Pos actual
^X Salir      ^J Justificar ^W Buscar   ^U Pág Sig  ^U PegarTxt  ^T Ortografía
```

Como podemos comprobar, el fichero se encuentra totalmente comentado, ya que se trata de un ejemplo para tener referencia. Nos ubicamos en la ultima línea ( o borramos todo) y comenzamos a configurar nuestro servidor:

En mi caso, he borrado el archivo por completo y he creado uno nuevo, con el mismo nombre.

```
[root@luigiftip ~]# rm /etc/vsftpd/vsftpd.conf
rm: ¿borrar el fichero regular «/etc/vsftpd/vsftpd.conf»? (s/n) s
[root@luigiftip ~]# nano /etc/vsftpd/vsftpd.conf
```

Comenzamos a configurar:

## 6.1 Configuraciones generales del FTP

**listen = YES** : Así el servicio se inicia con el sistema.

**write\_enable = YES** : Dar permiso de escritura a todos los usuarios que se conecten al FTP.

**userlist\_enable=YES**: Así evitamos el uso de cuentas del sistema (root, bin,adm, daemon,...) para acceder al servicio. Estas cuentas estan recogidas en el fichero: **/etc/vsftpd/user\_list**.

**dirmessage\_enable=YES** : Activar el mensaje de bienvenida.

**ftpd\_banner="Mensaje\_de\_bienvenida"**.

**pam\_service\_name=vsftpd** : Ubicamos el fichero de configuración PAM.



**xferlog\_enable=YES:** Como administradores de nuestro servidor, nos interesa hacer un seguimiento de los logeos. Para ello, hacemos uso del fichero **xferlog**, ubicado en **/var/log/vsftpd.log**.

## 6.2 Configurar pasivo

El modo activo tiene un **grave problema de seguridad**, ya que, el cliente puede aceptar cualquier conexión de entrada lo que la vuelve susceptible. Por lo tanto, usaremos únicamente el **modo pasivo**.

**pasv\_enable = YES :** Activamos el [modo pasivo](#)

**connect\_from\_port\_20=NO:** Desactivamos la conexión por el puerto 20 (modo activo).

## 6.3 Configuración para los usuarios locales

**local\_enable = YES :** Para poder conectarse con los usuarios locales del servidor.

**user\_config\_dir=/etc/vsftpd/usuarios:** Este parámetro nos permite personalizar los permisos para cada usuario. Haremos uso de esta directiva en el **apartado 6** de este manual.

**chroot\_local\_user = YES :** Activamos el uso de cuentas locales "privilegiadas".

**chroot\_list\_enable = YES :** Sirven para que los usuarios locales puedan navegar por todo el sistema. Solo se lo permitiremos a ciertos usuarios locales con el siguiente parámetro.

**chroot\_list\_file = /etc/vsftpd/vsftpd.chroot\_list :** Indicamos el fichero donde están listados los usuarios "privilegiados".

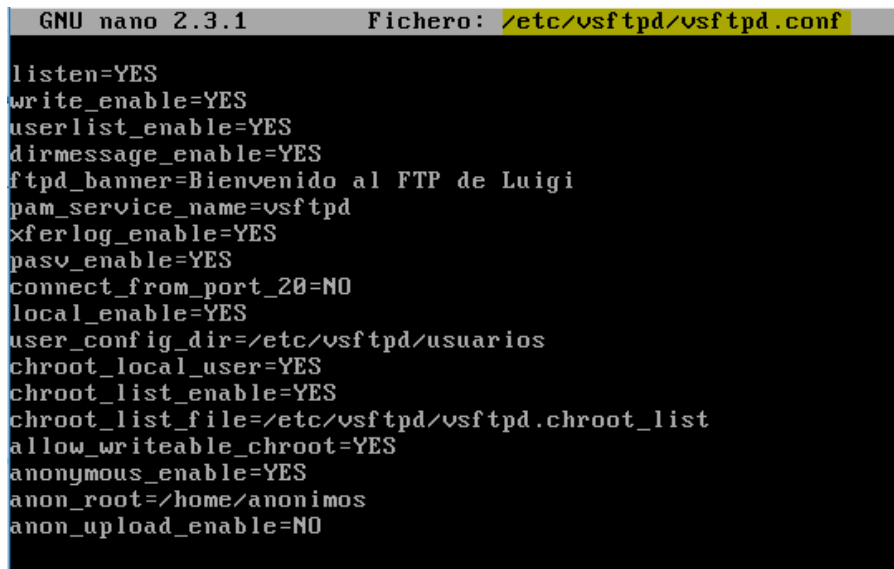
**allow\_writeable\_chroot=YES:** La siguiente directiva nos permite acceder como usuario "enjaulados" (permite la escritura del directorio).

## 6.4 Configuración para los usuarios anónimos

**anonymous\_enable=YES** : Permitimos el acceso al FTP mediante anónimo.

**anon\_root=/home/anonimo** : Indicamos el directorio para los usuarios anónimos.

**anon\_upload\_enable=NO** :No permitimos a los usuarios anónimos subir archivos al servidor.



```
GNU nano 2.3.1      Fichero: /etc/vsftpd/vsftpd.conf
listen=YES
write_enable=YES
userlist_enable=YES
dirmessage_enable=YES
ftpd_banner=Bienvenido al FTP de Luigi
pam_service_name=vsftpd
xferlog_enable=YES
pasv_enable=YES
connect_from_port_20=NO
local_enable=YES
user_config_dir=/etc/vsftpd/usuarios
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
allow_writeable_chroot=YES
anonymous_enable=YES
anon_root=/home/anonimos
anon_upload_enable=NO
```

## 6.5 Permisos de los usuarios

Como he comentado, uno de nuestros objetivos es que cada uno de nuestros usuarios del FTP, tengan diferentes permisos.

Así pues, vamos establecer los permisos para cada usuario:

**Usuario 1:** Este será el usuario de prueba para el administrador de sistemas. Por lo tanto, es el único usuario que **no se hallara enjaulado** en su directorio. Es decir, podrá navegar por el sistema a su antojo, crear ficheros/directorios, descargar/subir, editar...

**Usuario 2:** Este usuario se encontrara enjaulado en su directorio (/home/usu2), y además, tendrá permisos para **ver y descargar** lo que tenga en su directorio.

**Usuario 3:** Este último, podrá **ver, descargar y subir** ficheros/directorios a su directorio "jaula".

Antes de empezar, vamos a crear el directorio (**mkdir /etc/vsftpd/usuarios**), donde se encontraran los **ficheros de configuración de permisos** para cada usuario.

## 6.6 Establecer permisos

### Usuario 1 (usu1)

Únicamente lo añadimos a lista de usuarios "root". Para ello, ejecutamos **nano /etc/vsftpd/vsftpd.chroot\_list** y añadimos su nombre, sencillo...:

```
GNU nano 2.3.1          Fichero: vsftpd.chroot_list
usu1
```

### Usuario 2 (usu2)

Lo primero, creamos el fichero de configuración con su nombre: **nano /etc/vsftpd/usuarios/usu2**

Editamos el fichero con los siguientes parámetros:

**local\_root=/home/usu2:** Establecemos el directorio "jaula" del usuario  
**dirlist\_enable=YES** : parámetro que permite listar el contenido del directorio  
**download\_enable=YES:** Permitimos descargar contenido del directorio  
**write\_enable=NO: Denegamos** la escritura sobre el directorio, por tanto, **no podrá subir** contenido.

```
GNU nano 2.3.1          Fichero: /etc/vsftpd/usuarios/usu2
local_root=/home/usu2
dirlist_enable=YES
download_enable=YES
write_enable=NO
```

### Usuario 3 (usu3)

Creamos el fichero de configuración: **nano /etc/vsftpd/usuarios/usu3**

Redactamos los parámetros:

**local\_root=/home/usu3**

**dirlist\_enable=YES**

**download\_enable=YES**

**write\_enable=YES:** Permitimos la escritura sobre el directorio, por tanto, podrá subir contenido.

```
GNU nano 2.3.1 Fichero: /etc/vsftpd/usuarios/usu3
local_root=/home/usu3
dirlist_enable=YES
download_enable=YES
write_enable=YES
```

## 4. Pruebas de contexto

Antes de comenzar con las pruebas, debemos reiniciar el servicio para que aplique todos los cambios que hemos realizado. Para ello, ejecutamos: **systemctl restart vsftpd**.

Y además, vamos a **desactivar el firewall de forma temporal**, ya que, no hemos creado las reglas y excepciones para que el firewall **no bloquee** nuestro servicio FTP, **paso que realizaremos en la 2º parte de este manual (junto con el cifrado)**. Ejecutamos el comando: **systemctl stop firewalld**

```
[root@luigiftip /]# systemctl restart vsftpd
[root@luigiftip /]# systemctl stop firewalld
[root@luigiftip /]# _
```

**Listo nuestro servidor**, ahora, vamos a realizar las pruebas de contexto. Utilizaremos un sistema **Ubuntu** para realizar los ejemplos mediante comandos. Posteriormente utilizaremos el programa **Filezilla** en **Windows 7** y también utilizaremos FTP mediante el **navegador**:

## 1. FTP por comando (Ubuntu)

Como bien supondremos, nuestros equipos deben estar en red con el servidor. En este caso, nuestro equipo Ubuntu tendrá asignada la @ip **172.16.0.3/16**:

```
luigi@luigi-VirtualBox:~$ ifconfig
enp0s3  Link encap:Ethernet  direcciónHW 08:00:27:f5:40:0f
        Direc. inet:172.16.0.3  Difus.:172.16.255.255  Másc:255.255.0.0
        Dirección inet6: fe80::ff78:1576:b823:342a/64  Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
        Paquetes RX:709 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:690 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:66168 (66.1 KB)  TX bytes:55183 (55.1 KB)

lo      Link encap:Bucle local
        Direc. inet:127.0.0.1  Másc:255.0.0.0
        Dirección inet6: ::1/128  Alcance:Anfitrión
        ACTIVO BUCLE FUNCIONANDO  MTU:65536  Métrica:1
        Paquetes RX:388 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:388 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1
        Bytes RX:32016 (32.0 KB)  TX bytes:32016 (32.0 KB)
```

Comenzamos abriendo una consola y lanzamos el siguiente comando: **ftp 172.16.0.1**

### Usuario 1

Iniciamos sesión con usu1:

```
luigi@luigi-VirtualBox:~$ ftp 172.16.0.1
Connected to 172.16.0.1.
220 Bienvenido al FTP de Luigi
Name (172.16.0.1:luigi): usu1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Podemos **comprobar el logeo** en nuestro servidor:

```
GNU nano 2.3.1      Fichero: /var/log/vsftpd.log
Sat Oct 28 00:16:28 2017 [pid 100091] CONNECT: Client "172.16.0.3"
Sat Oct 28 00:16:31 2017 [pid 100081] [usu1] OK LOGIN: Client "172.16.0.3"
```

Como hemos mencionado anteriormente, usu1 sera el usuario de prueba para el administrador del servidor, es decir, nosotros. Por tanto, tendrá total libertad de uso del servicio...

Dicho esto, si lanzamos el comando `pwd`, nos dirá donde nos encontramos:

```
ftp> pwd
257 "/home/usu1"
ftp> █
```

Como observamos, nos encontramos en `/home/usu1`. Esto quiere decir que nos encontramos en la **raíz del servidor FTP, dentro del directorio `/home/usu1`**. Por tanto, tenemos **acceso al resto del sistema**. Vamos a comprobarlo, `cd /` :

```
ftp> cd /
250 Directory successfully changed.
ftp> pwd
257 "/"
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
lrwxrwxrwx   1 0      0              7 Oct 27 14:46 bin -> usr/bin
dr-xr-xr-x   5 0      0            4096 Oct 27 14:51 boot
drwxr-xr-x  19 0      0           3060 Oct 27 15:16 dev
drwxr-xr-x   7 0      0           8192 Oct 27 19:12 etc
drwxr-xr-x   8 0      0           88 Oct 27 19:12 home
lrwxrwxrwx   1 0      0              7 Oct 27 14:46 lib -> usr/lib
lrwxrwxrwx   1 0      0             9 Oct 27 14:46 lib64 -> usr/lib64
drwxr-xr-x   2 0      0            6 Nov 05 2016 media
drwxr-xr-x   2 0      0            6 Nov 05 2016 mnt
drwxr-xr-x   2 0      0            6 Nov 05 2016 opt
dr-xr-xr-x 100 0      0            0 Oct 27 15:51 proc
dr-xr-xr-x   2 0      0           135 Oct 27 15:14 root
drwxr-xr-x  23 0      0            680 Oct 27 16:50 run
lrwxrwxrwx   1 0      0              8 Oct 27 14:46 sbin -> usr/sbin
drwxr-xr-x   2 0      0            6 Nov 05 2016 srv
dr-xr-xr-x  13 0      0            0 Oct 27 15:16 sys
drwxrwxrwt   8 0      0           211 Oct 27 17:03 tmp
drwxr-xr-x  13 0      0           155 Oct 27 14:46 usr
drwxr-xr-x  20 0      0           278 Oct 27 15:16 var
226 Directory send OK.
ftp>
```

La siguiente captura **muestra** como nos hemos cambiado al directorio raíz, y además hemos **listado todos los ficheros/directorios** ubicados, en él. Como bien, sabemos, esto **no debe ser así para el resto de usuarios**.

## Usuario 2

Para la prueba, hemos creado un par de ficheros en `/home/usu2` y un directorio con otro fichero:

```
[root@luigiftp ~]# touch /home/usu2/prueba
[root@luigiftp ~]# touch /home/usu2/prueba2
[root@luigiftp ~]# mkdir /home/usu2/carpeta
[root@luigiftp ~]# touch /home/usu2/carpeta/prueba3
[root@luigiftp ~]# _
```

Vamos a logearnos como **usu2**, no sin antes recordar, que este usuario puede **ver y descargar** ficheros/directorios, y además, se encuentra **enjaulado**:

```
luigi@luigi-VirtualBox:~$ ftp 172.16.0.1
Connected to 172.16.0.1.
220 Bienvenido al FTP de Luigi
Name (172.16.0.1:luigi): usu2
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
GNU nano 2.3.1      Fichero: /var/log/vsftpd.log
Sat Oct 28 00:16:28 2017 [pid 10009] CONNECT: Client "172.16.0.3"
Sat Oct 28 00:16:31 2017 [pid 10008] [usu1] OK LOGIN: Client "172.16.0.3"
Sat Oct 28 00:38:41 2017 [pid 10015] CONNECT: Client "172.16.0.3"
Sat Oct 28 00:38:44 2017 [pid 10014] [usu2] OK LOGIN: Client "172.16.0.3"
```

Vamos a comprobar donde se encuentra ubicado:

```
Using binary mode to transfer files.
ftp> pwd
257 "/"
ftp>
```

Como se puede observar en la captura, usu2 se encuentra ubicado en /. Es decir, para usu2 **su raíz de trabajo es el directorio dónde lo hemos enjaulado** (/home/usu2).

Vamos a intentar navegar por el servidor FTP:

```
ftp> cd /home
550 Failed to change directory.
ftp> cd /var
550 Failed to change directory.
ftp> █
```

Como vemos, no reconoce los directorios, ya que, para el servicio **no existen**. Una vez realizada la prueba, vamos a **listar el contenido** del directorio "raíz" y comprobar que podemos **acceder al directorio "carpeta"**:

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 0      0          21 Oct 27 21:49 carpeta
-rw-r--r--  1 0      0          0 Oct 27 21:49 prueba
-rw-r--r--  1 0      0          0 Oct 27 21:49 prueba2
226 Directory send OK.
ftp> cd /carpeta
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--  1 0      0          0 Oct 27 21:49 prueba3
226 Directory send OK.
ftp> █
```

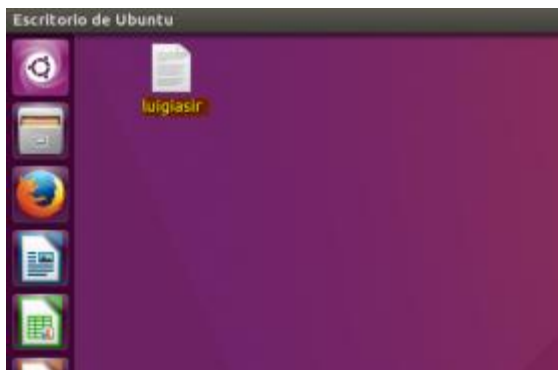
Una vez comprado que el usuario se encuentra **enjaulado en su directorio**, vamos a **descargar** el fichero "prueba2" en /home/luigi (DEL SISTEMA UBUNTU!!):

```
ftp> get prueba2 /home/luigi/prueba2
local: /home/luigi/prueba2 remote: prueba2
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for prueba2 (0 bytes).
226 Transfer complete.
ftp>
```

Comprobamos que se ha descargado:

```
luigi@luigi-VirtualBox:~$ ls /home/luigi
Descargas Documentos Escritorio examples.desktop Imágenes Música Plantillas prueba2 Público Videos
luigi@luigi-VirtualBox:~$
```

Ahora, vamos a intentar **subir un fichero**. Para eso, he creado un fichero (**luigiasir**) en el escritorio de Ubuntu:



```
ftp> put /home/luigi/Escritorio/luigiasir
local: /home/luigi/Escritorio/luigiasir remote: /home/luigi/Escritorio/luigiasir
200 PORT command successful. Consider using PASV.
550 Permission denied.
ftp>
```

Por último, solo nos queda **comprobar** que no podemos **eliminar, crear directorio y/o eliminar directorios**:

```
ftp> rm prueba
550 Permission denied.
ftp> mkdir carpeta2
550 Permission denied.
ftp> rmdir carpeta
550 Permission denied.
ftp>
```



### Usuario 3

Tal y como hemos configurado, realizarnos la comprobación de que usu3, puede subir ficheros, eliminar y crear directorios.

Nos logeamos como usu3:

```
luigi@luigi-VirtualBox:~$ ftp 172.16.0.1
Connected to 172.16.0.1.
220 Bienvenido al FTP de Luigi
Name (172.16.0.1:luigi): usu3
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

GNU nano 2.3.1          Fichero: /var/log/vsftpd.log
Sat Oct 28 00:16:28 2017 [pid 10009] CONNECT: Client "172.16.0.3"
Sat Oct 28 00:16:31 2017 [pid 10008] [usu1] OK LOGIN: Client "172.16.0.3"
Sat Oct 28 00:38:41 2017 [pid 10015] CONNECT: Client "172.16.0.3"
Sat Oct 28 00:38:44 2017 [pid 10014] [usu2] OK LOGIN: Client "172.16.0.3"
Sat Oct 28 01:02:49 2017 [pid 10021] CONNECT: Client "172.16.0.3"
Sat Oct 28 01:02:54 2017 [pid 10020] [usu3] OK LOGIN: Client "172.16.0.3"
```

Esta vez tenemos un **directorio vacío**, por tanto vamos a **subir nosotros un fichero**, por ejemplo, uno creado anteriormente (luigiasir2):

```
ftp> ll
Descargas  Documentos  Escritorio  examples.desktop  Imágenes  luigiasir2  Música  Plantillas  Público  Videos
ftp> put luigiasir2
local: luigiasir2 remote: luigiasir2
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw----- 1 1003 50 0 Oct 27 22:44 luigiasir2
226 Directory send OK.
ftp>
```

Ahora, vamos a comprobar que podemos **cambiar el nombre** al fichero recién subido, luego **crearemos un directorio**, y **eliminaremos todo por completo**:

```
ftp> mkdir carpeta_temporal
257 "/carpeta_temporal" created
ftp> rename luigiasir2 fichero_temporal
350 Ready for RNT0.
250 Rename successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwx----- 2 1003 50 6 Oct 27 22:53 carpeta_temporal
-rw----- 1 1003 50 0 Oct 27 22:44 fichero_temporal
226 Directory send OK.
ftp> rmdir carpeta_temporal
250 Remove directory operation successful.
ftp> rm fichero_temporal
550 Remove directory operation failed.
ftp>
```

## Anónimo

Por último, nuestro servidor alberga la posibilidad de **logearnos como anónimo**, **opción muy común** en gran parte de los FTP que rondan la red, ya qué, es común que un usuario quiere **descargar un archivo**, como por ejemplo, unos drivers, **sin necesidad** de que este tenga un **cuenta** dentro del servidor.

Por tanto, nuestro usuario anónimo, **únicamente podrá descargar** ficheros que nosotros ubicamos en su único **directorio de trabajo** (/home/anonimo):

Vamos a empezar **logeandonos** como anónimo (anonymous), sin contraseña, esta claro...:

```
luigi@luigi-VirtualBox:~$ ftp 172.16.0.1
Connected to 172.16.0.1.
220 Bienvenido al FTP de Luigi
Name (172.16.0.1:luigi): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
GNU nano 2.3.1 Fichero: /var/log/vsftpd.log
Sat Oct 28 00:51:56 2017 [pid 100661] [usu3] FAIL UPLOAD: Client "172.16.0.3", "$
Sat Oct 28 00:52:32 2017 [pid 100661] [usu3] FAIL DELETE: Client "172.16.0.3", "$
Sat Oct 28 00:52:39 2017 [pid 100661] [usu3] OK RMDIR: Client "172.16.0.3", "/ca$
Sat Oct 28 00:52:47 2017 [pid 100691] CONNECT: Client "172.16.0.3"
Sat Oct 28 00:52:49 2017 [pid 100681] [usu3] OK LOGIN: Client "172.16.0.3"
Sat Oct 28 00:53:00 2017 [pid 100731] [usu3] OK MKDIR: Client "172.16.0.3", "/ca$
Sat Oct 28 00:53:15 2017 [pid 100731] [usu3] OK RENAME: Client "172.16.0.3", "/l$
Sat Oct 28 00:53:27 2017 [pid 100731] [usu3] OK RMDIR: Client "172.16.0.3", "/ca$
Sat Oct 28 00:53:36 2017 [pid 100731] [usu3] FAIL RMDIR: Client "172.16.0.3", "$
Sat Oct 28 01:03:10 2017 [pid 100921] CONNECT: Client "172.16.0.3"
Sat Oct 28 01:03:14 2017 [pid 100911] [ftp] OK LOGIN: Client "172.16.0.3", anon $
```

*Nota: En la anterior captura podemos comprobar como el fichero de log ha registrado las anteriores pruebas de contexto que hemos realizado.*

Comprobamos que podemos **descargar**:

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 0 Oct 27 23:07 README
226 Directory send OK.
ftp> get README
local: README remote: README
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for README (0 bytes).
226 Transfer complete.
ftp>
```

Comprobamos que **no podemos subir archivos/directorios**:

```
ftp> !ls
Descargas Documentos Escritorio examples.desktop Imágenes luigiasir2 Música Plantillas Público README Vídeos
ftp> put luigiasir2
local: luigiasir2 remote: luigiasir2
200 PORT command successful. Consider using PASV.
550 Permission denied.
ftp> mkdir carpeta
550 Permission denied.
```

Ni **navegar** a través del servidor:

```
ftp> cd /home/luigiftp
550 Failed to change directory.
ftp> cd /etc/vsftpd
550 Failed to change directory.
ftp> █
```

Ni **eliminar** contenido:

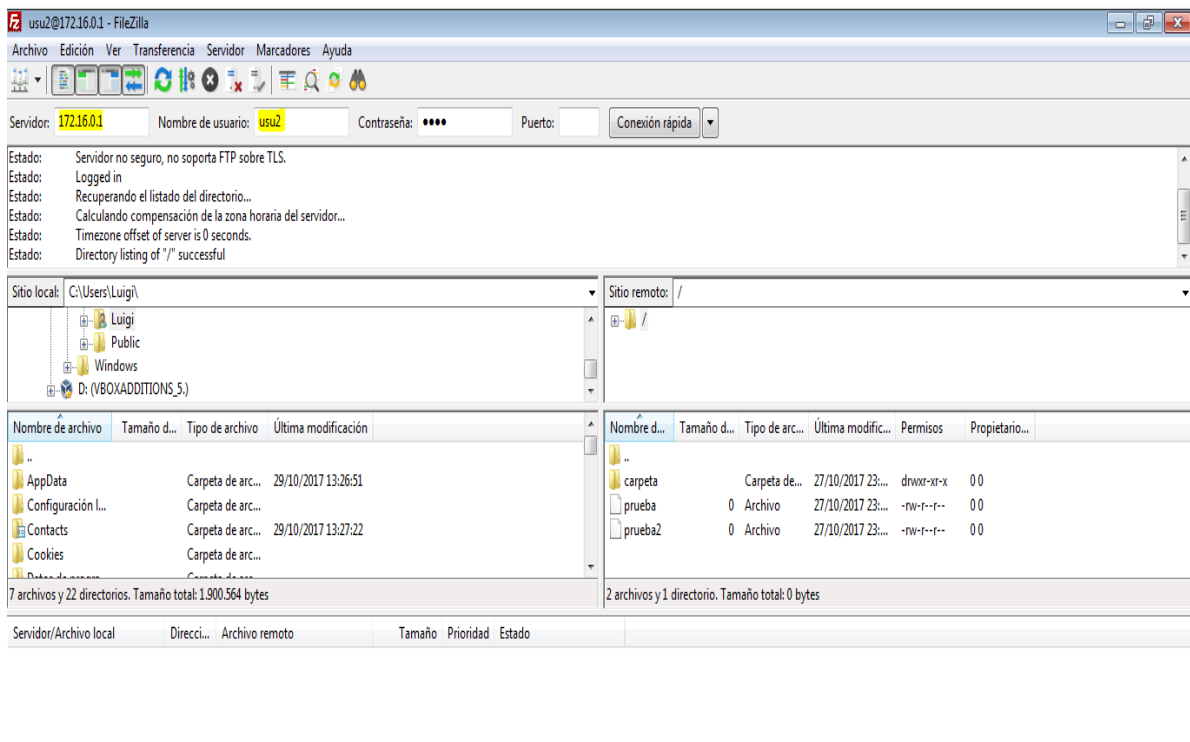
```
ftp> rm README
550 Permission denied.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--  1 0      0              0 Oct 27 23:07 README
226 Directory send OK.
ftp>
```

## 2. FTP mediante Filezilla y Chrome

Una vez comprobado que el servidor se encuentra **accesible** y que los usuarios están **limitados**, vamos a realizar la conexión al FTP vía **Filezilla Client** y mediante navegador web (**Chrome**)

### Filezilla Client

Para logearnos mediante este software, nos ubicamos en "**Servidor**" y escribimos la @ip del FTP. Además introducimos las **credenciales de inicio de sesión** de algunos de los usuarios:

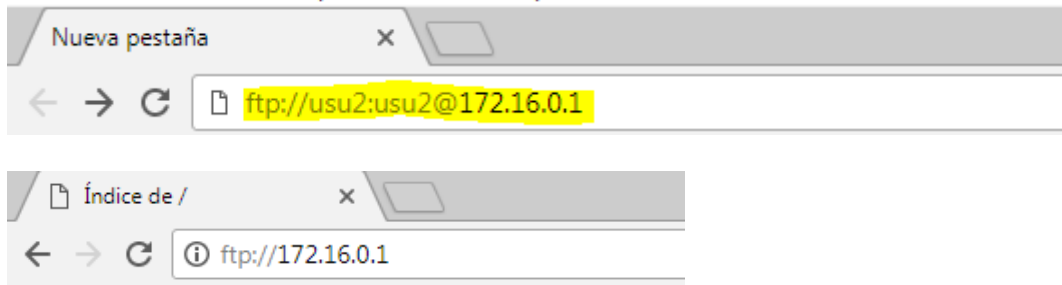


## Chrome




Para iniciar sesión en nuestro servidor, debemos seguir la siguiente sintaxis:

`ftp://usuario:contraseña@ip_ftp`

En nuestro caso:



### Índice de /

Nombre	Tamaño	Fecha de modificación
 carpeta/		27/10/17 23:49:00
 prueba	0 B	27/10/17 23:49:00
 prueba2	0 B	27/10/17 23:49:00

Si en cambio, escribimos en la barra de direcciones únicamente: `ftp://172.16.0.1`, accedemos como usuario anónimo.

Llegados a este punto, hemos obtenido un servidor FTP estable. A partir de este momento comenzaremos las configuraciones para añadir seguridad al mismo.

## 5. FTP Seguro (FTPS)

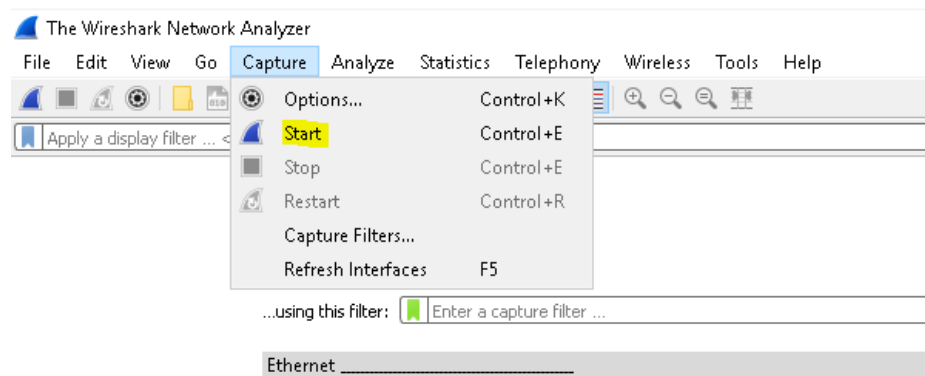
### 1. Auditar nuestro servidor antes de asegurarlo

Como comentamos anteriormente, **FTP** es un **protocolo no seguro**, ya que, realiza la transferencia de datos en **texto plano**. Por tanto, si un atacante **esnifara este trafico**, podría **capturar paquetes** dónde se encontraran, sin **ningún tipo de cifrado**, el usuario y la contraseña con la que se ha logeado un cliente.

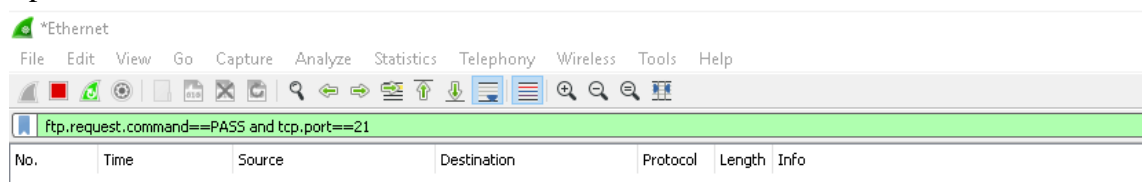
Para comprobarlo, vamos a establecer un sniffer en nuestra red. En este caso, utilizaremos uno de los más comunes **Wireshark en Windows 10 64 bits**.

Con nuestro Windows dentro de la red 172.16.0.0/16, lo primero que haremos sera iniciar el sniffer y filtrar los paquetes recibidos:

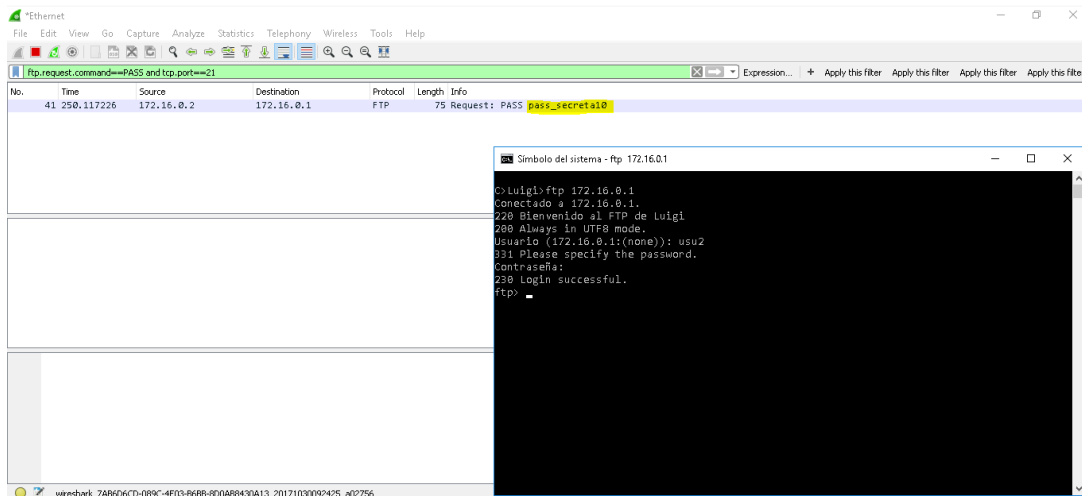
#### 1. Iniciamos Wireshark



#### 2. Aplicamos el filtro



### 3. Ahora, vamos a iniciar sesión el usuario (usu2) y comprobamos como se **captura la PASS**:



### 4. También podemos filtrar el usuario y así, tendremos las credenciales para poder logearnos dentro del servidor:

ftp.request.command==USER and tcp.port==21						
No.	Time	Source	Destination	Protocol	Length	Info
16	26.882818	172.16.0.2	172.16.0.1	FTP	65	Request: USER usu2
38	243.477837	172.16.0.2	172.16.0.1	FTP	65	Request: USER usu2

Como podemos observar, el filtro hace referencia al **puerto 21 TCP**, ya que sabemos que nuestro servidor FTP utiliza el **modo pasivo** y no el activo.

Sí además, retiramos el filtro de USER, vemos **toda la información transmitida al servidor**, incluido el mensaje de bienvenida... **Totalmente en texto plano, sin ningún cifrado**:

tcp.port==21						
No.	Time	Source	Destination	Protocol	Length	Info
9	25.272547	172.16.0.1	172.16.0.2	TCP	66	21 → 49685 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK...
10	25.272589	172.16.0.2	172.16.0.1	TCP	54	49685 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
11	25.273974	172.16.0.1	172.16.0.2	FTP	86	Response: 220 Bienvenido al FTP de Luigi
12	25.275346	172.16.0.2	172.16.0.1	FTP	68	Request: OPTS UTF8 ON
13	25.275559	172.16.0.1	172.16.0.2	TCP	60	21 → 49685 [ACK] Seq=33 Ack=15 Win=29312 Len=0
14	25.275619	172.16.0.1	172.16.0.2	FTP	80	Response: 200 Always in UTF8 mode.
15	25.318908	172.16.0.2	172.16.0.1	TCP	54	49685 → 21 [ACK] Seq=15 Ack=59 Win=8134 Len=0
16	26.882818	172.16.0.2	172.16.0.1	FTP	65	Request: USER usu2
17	26.883150	172.16.0.1	172.16.0.2	FTP	88	Response: 331 Please specify the password.

## 1. Asegurar FTP con SSL

Una vez visto que, el hecho de que alguien pudiera capturar un paquete en la ruta, seria una vulnerabilidad muy grave para nuestro servidor. La solución será implementar un protocolo seguro.

Para ello vamos a configurar FTP para que trabaje sobre SSL:

### 1. Generar certificado SSL:

Vamos a **generar un certificado** a partir de la herramienta [OpenSSL](#).

Lo primero instalamos el paquete: **yum install openssl**

Generemos el nuevo certificado (caduca en 1 año): **openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/vsftpd/vsftpd.pem -out /etc/vsftpd/vsftpd.pem**

```
[root@luigiftftp ~]# openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/vsftpd/vsftpd.pem -out /etc/vsftpd/vsftpd.pem
Generating a 1024 bit RSA private key
.....+++++
writing new private key to '/etc/vsftpd/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) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
[root@luigiftftp ~]#
```

*Nota: los datos solicitados son opciones (en mi caso los he dejado en blanco)*



## 2. Editar el fichero de configuración:

**ssl\_enable=YES** : Directiva que habilita el uso de SSL

**allow\_anon\_ssl=YES** : Habilitar SSL con usuarios anónimos

**force\_local\_data\_ssl=YES** : Obliga a usar certificado SSL para traferir datos.

**force\_local\_logins\_ssl=YES**: Obliga a usar certificado SSL para autenticar usuarios locales.

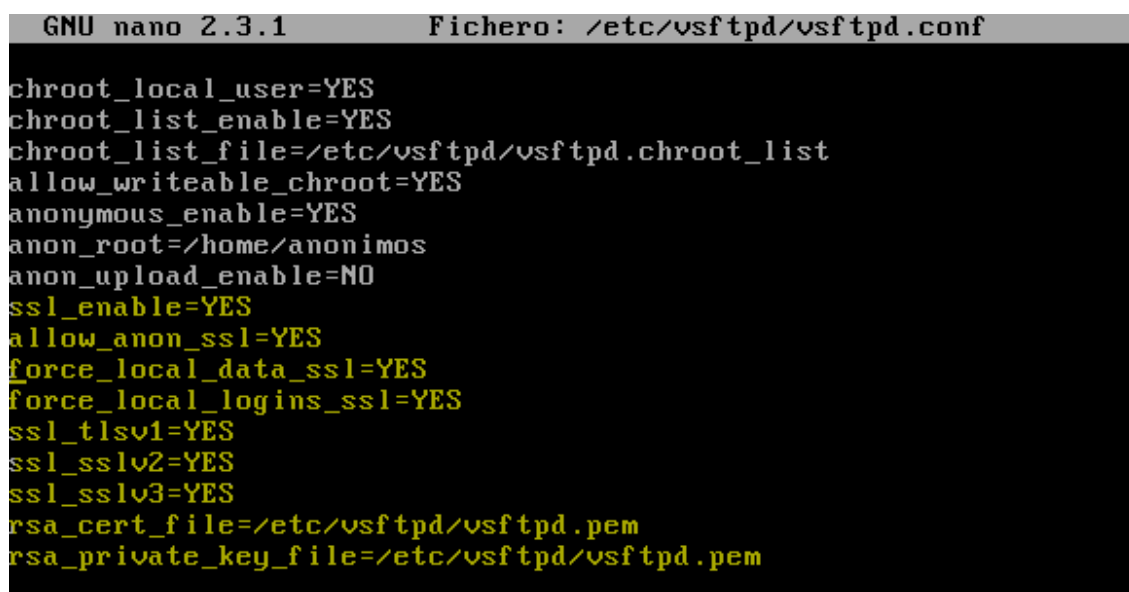
**ssl\_tlsv1=YES** :Habilita el uso de SSLv1

**ssl\_sslv2=YES**: Habilita el uso de SSLv1

**ssl\_sslv3=YES**: Habilita el uso de SSLv1

**rsa\_cert\_file=/etc/vsftpd/vsftpd.pem**: Ubicación del certificado SSL

**rsa\_private\_key\_file=/etc/vsftpd/vsftpd.pem**: Ubicación de la llave privada SSL



```
GNU nano 2.3.1      Fichero: /etc/vsftpd/vsftpd.conf
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
allow_writeable_chroot=YES
anonymous_enable=YES
anon_root=/home/anonimos
anon_upload_enable=NO
ssl_enable=YES
allow_anon_ssl=YES
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=YES
ssl_sslv3=YES
rsa_cert_file=/etc/vsftpd/vsftpd.pem
rsa_private_key_file=/etc/vsftpd/vsftpd.pem
```

Por ultimo, **reiniciamos el servicio** para aplicar los cambios: **systemctl restart vsftpd**.

## 3. Configurando firewall

En el anterior post, tuvimos que desactivar el servicio firewall para poder hacer uso del FTP. Ahora, ya que estamos añadiendo seguridad a nuestro servidor, vamos a configurar nuestro firewall para que permita el uso de FTP. Así, conseguiremos que nuestro servidor no sea tan vulnerable, escuchando únicamente a los puertos que son necesarios.

Para este caso, vamos a utilizar los puertos 10090-10100. Vamos a añadir la excepción al firewall:

1. Primero, descubrimos como se llama nuestra zona de actuación, y después, añadimos la excepción:

```
firewall-cmd --get-active-zones
```

```
firewall-cmd --permanent --zone=public --add-port=10090-10100/tcp
```

2. Y reiniciamos el firewall:

```
firewall-cmd --reload
```

```
[root@luigiftip /]# firewall-cmd --get-active-zones
public
  interfaces: enp0s3
[root@luigiftip /]# firewall-cmd --permanent --zone=public --add-port=10090-10100
/tcp
success
[root@luigiftip /]# firewall-cmd --reload
success
[root@luigiftip /]# _
```

3. Ahora, vamos a establecer las directivas en el archivo de configuración:

```
pasv_min_port=10090
```

```
pasv_max_port=10100
```

```
GNU nano 2.3.1      Fichero: /etc/vsftpd/vsftpd.conf

listen=YES
write_enable=YES
userlist_enable=YES
dirmessage_enable=YES
ftpd_banner=Bienvenido al FTP de Luigi
pam_service_name=vsftpd
xferlog_enable=YES
pasv_enable=YES
connect_from_port_20=NO
pasv_min_port=10090
pasv_max_port=10100
local_enable=YES
user_config_dir=/etc/vsftpd/usuarios
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
allow_writeable_chroot=YES
anonymous_enable=YES
anon_root=/home/anonimos
anon_upload_enable=NO
```

4. Reiniciamos el servicio

## 4. Uso del FTPS

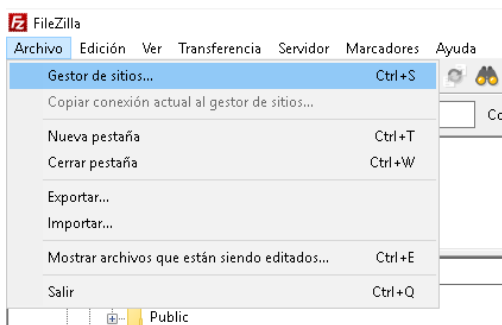
Ya tenemos listo nuestro **servidor seguro**. Ahora solo nos queda hacer uso de él. Ya que, hemos implementado el uso de certificados para cifrar la conexión, ya no podremos acceder al servidor mediante la consola:

```
C>Luigi>ftp 172.16.0.1
Conectado a 172.16.0.1.
220 Bienvenido al FTP de Luigi
200 Always in UTF8 mode.
Usuario (172.16.0.1:(none)): usu2
530 Non-anonymous sessions must use encryption.
Conexión cerrada por el host remoto.

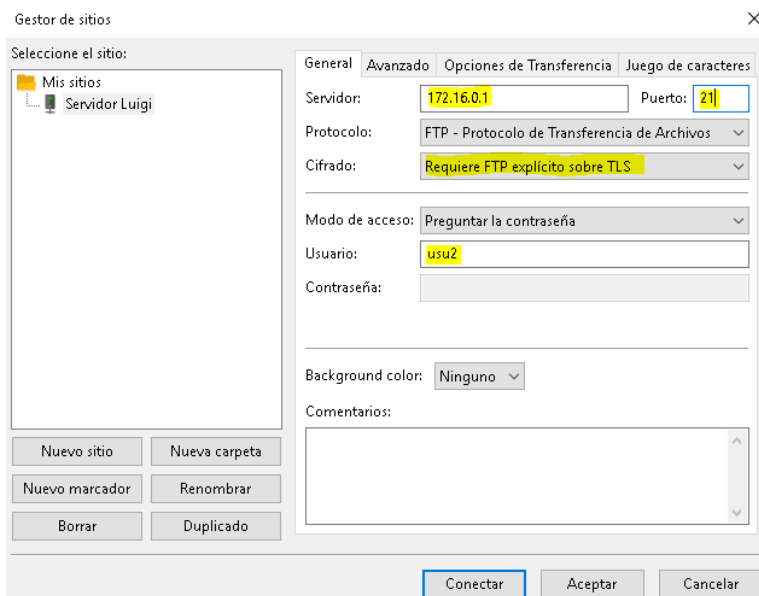
C>Luigi>_
```

Por ello, vamos a configurar nuestro **Filezilla Client** para poder acceder a FTPS:

### 1. Creamos la nueva conexión: "Gestor de sitios...":

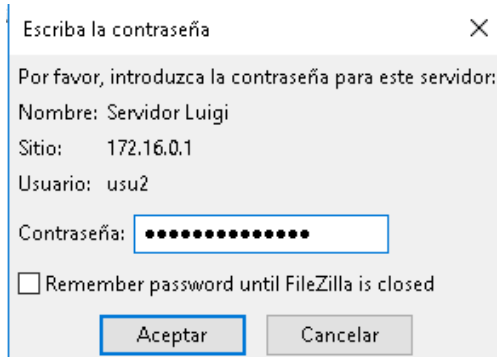


### 2. Clic sobre "Nuevo sitio" → Damos parámetros a la conexión:



Tal y como se muestra en la captura, hacemos referencia a la **@ip del servidor FTP**, en el **puerto 21 (modo pasivo)**. Vamos a acceder con usu2.

3. Hacemos clic sobre "**Conectar**" y nos solicitara la contraseña del usuario:



Escriba la contraseña

Por favor, introduzca la contraseña para este servidor:

Nombre: Servidor Luigi

Sitio: 172.16.0.1

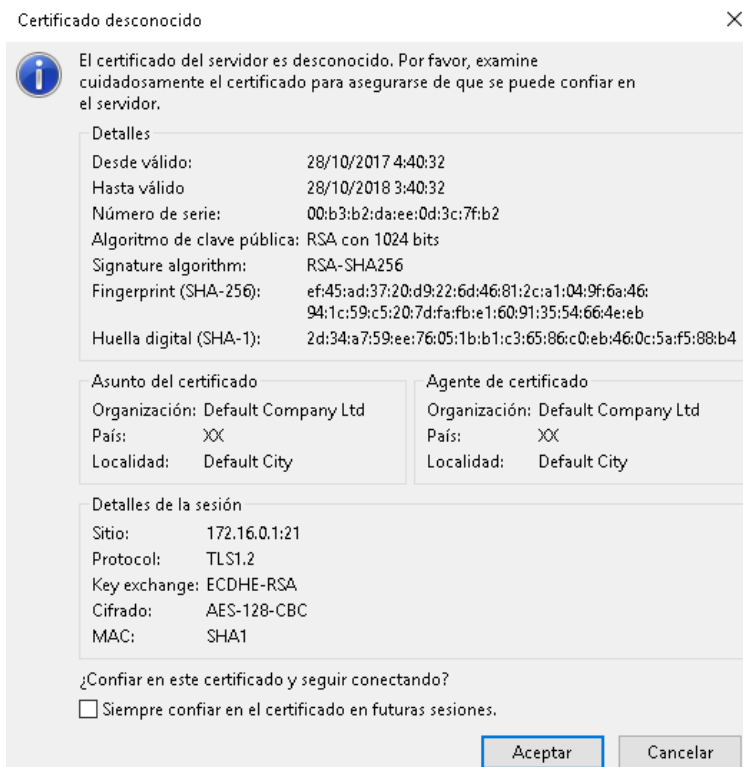
Usuario: usu2

Contraseña: [ocultada]

☐ Remember password until FileZilla is closed

Aceptar Cancelar

4. Advertencia sobre el certificado → **Aceptar**:



Certificado desconocido

El certificado del servidor es desconocido. Por favor, examine cuidadosamente el certificado para asegurarse de que se puede confiar en el servidor.

**Detalles**

Desde válido:	28/10/2017 4:40:32
Hasta válido:	28/10/2018 3:40:32
Número de serie:	00:b3:b2:da:ee:0d:3c:7f:b2
Algoritmo de clave pública:	RSA con 1024 bits
Signature algorithm:	RSA-SHA256
Fingerprint (SHA-256):	ef:45:ad:37:20:d9:22:6d:46:81:2c:a1:04:9f:6a:46:94:1c:59:c5:20:7d:fa:fb:e1:60:91:35:54:66:4e:eb
Huella digital (SHA-1):	2d:34:a7:59:ee:76:05:1b:b1:c3:65:86:c0:eb:46:0c:5a:f5:88:b4

Asunto del certificado	Agente de certificado
Organización: Default Company Ltd	Organización: Default Company Ltd
País: XX	País: XX
Localidad: Default City	Localidad: Default City

**Detalles de la sesión**

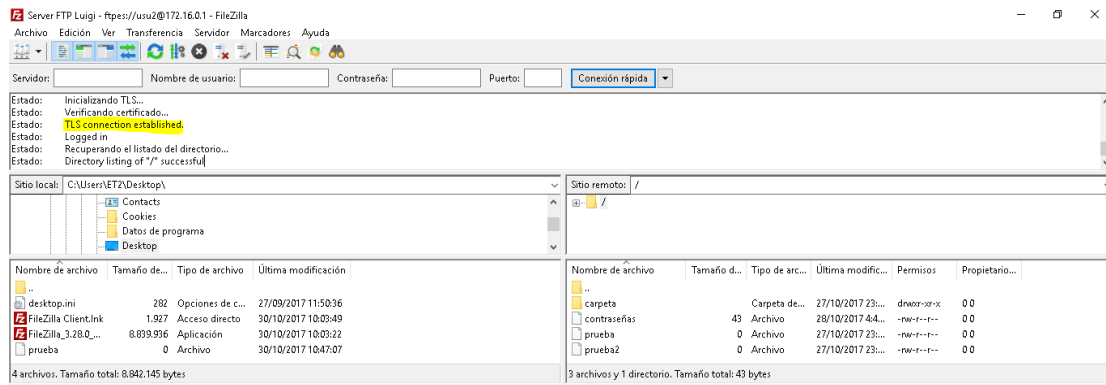
Sitio:	172.16.0.1:21
Protocol:	TLS1.2
Key exchange:	ECDHE-RSA
Cifrado:	AES-128-CBC
MAC:	SHA1

¿Confiar en este certificado y seguir conectando?

☐ Siempre confiar en el certificado en futuras sesiones.

Aceptar Cancelar

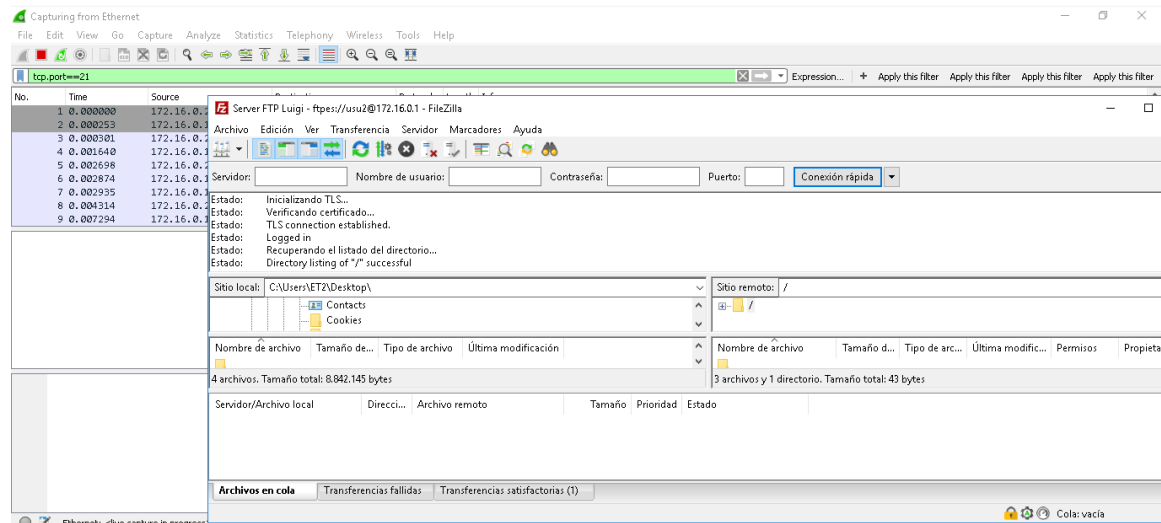
## 5. Conexión segura establecida



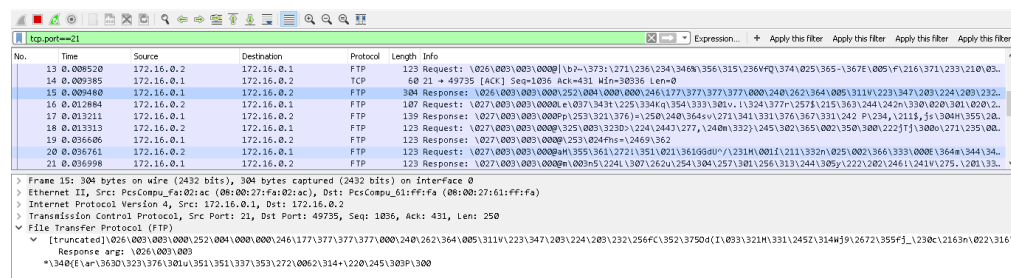
## 5. Segunda auditoración

Vamos a volver a **capturar paquetes** con Wireshark, aplicando unicamente el filtro **tcp.port==21**.

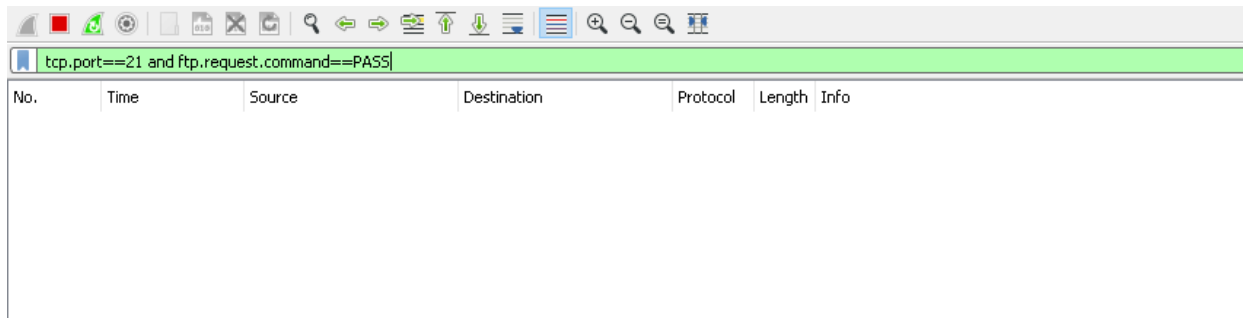
Iniciamos sesión en el servidor FTPS:



Vemos como se han **capturado los paquetes**. Ahora vamos a analizarlos:



Comprobamos que no se muestra nada de información, ya que se encuentran **cifrados los datos**. Si aplicamos el filtro **ftp.request.command==PASS**, no obtendremos ningún resultado:



## 6. Conclusión

Una vez realizado, tenemos nuestro servidor FTP seguro, con cifrado para nuestra conexión al servidor usando SSL y además, hemos configurado el firewall de CentOS 7 para que permita el uso de FTP sin desactivar complementos el cortafuegos.

