

Capa de Aplicación, FTP

Andres Barbieri *

31 de marzo de 2017

1. Aclaración para el Lector

Este texto fue generado en el año 2008 para el postgrado de Redes II y la materia Redes y Comunicaciones de la Fac. de Informática - UNLP y con el tiempo fue sufriendo modificaciones. El protocolo FTP con el transcurso tecnológico a ido perdiendo terreno dejando su lugar a nuevos modelos. Parece ser que al no cuadrar en un protocolo optimizado por los balanceadores/aceleradores-web, genera mucha carga en los servidores finales [KORG-FTP]. También los desarrolladores de los navegadores web (los principales clientes hasta el momento) no han actualizado este comportamiento, por ejemplo no soportan FTP seguro. Por otro lado las herramientas usadas para mostrar los ejemplos prácticos ya han sido superadas por varias versiones posteriores. A pesar de su falta de actualidad el texto presentado parece útil para aprender el protocolo, del cual se adjuntan varias capturas de tráfico para ser analizadas.

2. Introducción

El protocolo FTP (File Transfer Protocol) conforma el grupo de los protocolos más viejos de la Internet aún utilizados [RFC-114], de forma conjunta con TELNET, el servicio de MAIL y otros. Este protocolo data de 1971 y funcionó antes de que exista TCP/IP, ejecutaba sobre NCP. Aunque, el protocolo original ha sufrido varias modificaciones para ser adaptado a IP y a nuevas necesidades, la esencia del mismo se conserva. FTP es un protocolo para copiar archivos completos, a diferencia de otros protocolos que brindan acceso a archivos, pero dando un servicio más transparente, como NFS (Network File System). El documento que estandariza a FTP es [RFC-765], luego convertido en obsoleto por [RFC-959]. Este último ha sido actualizado por otras RFC como [RFC-2228] o RFC-3659. De la misma forma que la mayoría de los protocolos de Internet, como HTTP, SMTP y TELNET, sus mensajes se codifican en ASCII estándar (de 7 bits codificados en 8). Este tipo de terminal se la conoce como ASCII NVT (Network Virtual Terminal).

Según [StevI], haciendo referencia a otras fuentes, FTP era, para 1992, el protocolo que más datos transportaba. En la actualidad, ha sido superado ampliamente por HTTP. Los clientes FTP utilizados, en general, no requieren interfaz gráfica (GUI). El más conocido es el BSD `ftp(1)` que apareció por primera vez en el Unix 4.2BSD. Actualmente, los clientes FTP gráficos son más comunes como el FileZilla [FZ], distribuido bajo licencia GNU, o WS_FTP de Ipswitch, bajo licencia comercial. También, los navegadores Web más populares incluyen soporte de FTP. En cuanto a los servidores se pueden mencionar el BSD

*barbieri@cespi.unlp.edu.ar, con colaboración de Matías Robles: mrobles@info.unlp.edu.ar.



ftpd(8) de Unix BSD, WU-FTP (Washington University FTP) el más utilizado hasta el 2000, Pure-FTPd y vsFTPd. Estos dos últimos distribuidos bajo licencia GNU. Para el 2008, los sitios de repositorios de RedHat, OpenBSD y SUSE estaban hosteados con vsFTPd.

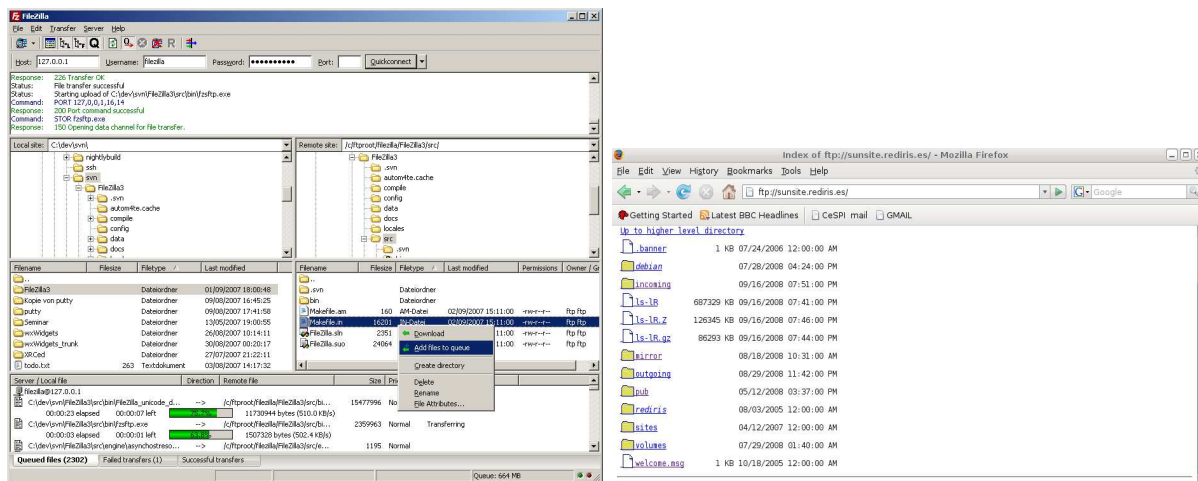


Figura 1: FileZilla y FTP vía web browser.

3. Protocolo FTP

El funcionamiento del protocolo FTP trabaja en un modelo cliente/servidor, pero se distingue de los demás servicios por el hecho de usar una conexión para el protocolo de control (Out-Of-Band Control) y otra conexión para la transferencia de datos. Cada conexión tendrá requerimientos de servicios diferentes. Por ejemplo, la de control, minimizar el delay, y la de datos, maximizar el throughput. El protocolo se ejecuta sobre TCP porque requiere un protocolo de transporte que le brinde detección de errores (confiabilidad). Como dicen [RFC-765] y [RFC-959].

...

There is no provision for detecting bits lost or scrambled in data transfer; this level of error control is handled by the TCP.

...

El protocolo FTP provee funcionalidad de la capa de Sesión al permitir hacer *Re-starts* de las transferencias de archivos que fueron canceladas por errores “graves”, como caída del servidor. Según [RFC-959].

...

error recovery

A procedure that allows a user to recover from certain errors such as failure of either host system or transfer process. In FTP, error recovery may involve restarting a file transfer at a given checkpoint.

...



Además, FTP se encarga de varios aspectos de la capa de Presentación, como es el formato de los datos. Puede manejar los formatos EBCDIC, ASCII o IMAGE (BINARY, RAW). Estos formatos deben ser escogidos por el usuario antes de la transferencia. En un principio, el formato default era ASCII, pero hoy es más común encontrar IMAGE.

- ASCII (A).
- EBCDIC (E).
- IMAGE (I).
- LOCAL (L).

En el caso de ASCII se convierte el archivo origen a formato ASCII y, luego, el receptor lo debe pasar de ASCII al formato local. Cada fin de línea se traduce a CR,LF. En el caso de Unix, el fin de línea es solo LF. Si se utiliza este formato para transmitir un archivo binario, el archivo será modificado y no se lo podrá recuperar.

Las plataformas (OS) pueden almacenar los archivos en diferentes estructuras. FTP define estructuras para transportar datos. Formato default File (F). Se especifica el formato para transferencia con el comando STRU. Los formatos son:

- File F secuencia de bytes sin estructura.
- Record R una serie de registro.
- Page P una serie de bloques (pages).

Con respecto a la capa de Presentación, FTP define diferentes modos de transferencia:

- Un hilo de bytes: (S) Stream. EOF como Registro o cierre de stream de acuerdo al formato del archivo: STRU.
- En Bloques (B). 1 header + serie de bloques, permite interrupción.
- Comprimido (C). Utiliza RLE: Run Length Encoding. E.g. BBBBBBNN == 6B2N.

El primero, Stream, es el más utilizado.

3.1. Ejemplos de Respuestas FTP

Los siguientes son algunos de los mensajes de respuesta a los comandos ingresados durante una sesión de FTP.

```
** Positive preliminary reply:  
150 Here comes the directory listing.
```

```
** Positive completion reply:  
200 Switching to ASCII mode.  
200 Switching to Binary mode.
```



200 NOOP ok.
200 PORT command successful. Consider using PASV.
214 The following commands are recognized
226 Directory send OK.
230 Login successful.
250 Rename successful.

** Positive intermediate reply:
331 Please specify the password.
350 Ready for RNT0.

** Transient negative completion reply:
426 Failure writing network stream.

** Permanent negative completion reply:
500 Command not understood.
504 Bad MODE command.
530 Please login with USER and PASS.
550 RNFR command failed.
550 Failed to open file.

3.2. Conexiones del Protocolo FTP

Como se mencionó más arriba, FTP mantiene una conexión o sesión para enviar la información de control y otra para la transferencia de datos. En los siguientes puntos se explica, brevemente, el funcionamiento de FTP.

1. Primero, el cliente se conecta al servidor estableciendo la sesión de control. El servidor, para la sesión de control, escucha en el puerto 21.
2. El cliente tiene la posibilidad de autenticarse o de entrar como usuario anónimo. Esto lo hace utilizando la conexión y comandos de control.
3. Luego, el cliente enviará los comandos respectivo de acuerdo a las operaciones que quiera realizar. Entre éstas nos encontramos con comandos que requerirán la conexión de datos y otros que no:
 - Enviar un archivo al servidor (STOR).
 - Traer un archivo desde el servidor (RETR).
 - Listar los archivos en el servidor (LIST).

Algunos comandos que no requieren la sesión de datos:

- Ver en el directorio donde está parado (CWD, PWD).
- Ver información de un archivo en particular (STAT, SIZE).
- Cambiar el modo (MODE).
- Obtener ayuda (HELP).
- Borrar un archivo (DELE).



- Crear un directorio (MKD).
 - Borrar un directorio (RMD).
 - Moverse entre directorios (CD).
4. Antes de enviar un comando que requiere una conexión de datos, necesitará establecerla. Para esto, como se verá más adelante, existen dos modos.
 5. La sesión de datos se abre y cierra por cada transferencia individual.
 6. Por último, el cliente decidirá terminar la sesión de control. Los servidores pueden tener configurados timers que luego de estar la sesión de control y de datos relacionadas un tiempo sin usar pueden dar de baja la conexión.

3.3. Comandos sin sesión de datos

Se puede utilizar directamente el comando `ftp(1)` o usar el comando `telnet(1)` como programa para manejar la sesión TCP de control. Los puertos del servidor FTP en modo activo son los siguientes.

```
# grep ftp /etc/services
ftp-data      20/tcp
ftp           21/tcp
...
```

A continuación se muestra una sesión FTP usando un cliente FTP estándar:

```
? ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 2.0.5)
Name (127.0.0.1:andres): andres2
331 Please specify the password.
Password: ****
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221 Goodbye.
```

```
? ftp
ftp> help
Commands may be abbreviated.  Commands are:
```

!	debug	mdir	qc	send
\$	dir	mget	sendport	site
account	disconnect	mkdir	put	size
append	exit	mls	pwd	status
ascii	form	mode	quit	struct
bell	get	modtime	quote	system



binary	glob	mput	recv	sunique
bye	hash	newer	reget	tenex
case	help	nmap	rstatus	tick
cd	idle	nlist	rhelph	trace
cdup	image	ntrans	rename	type
chmod	lcd	open	reset	user
close	ls	prompt	restart	umask
cr	macdef	passive	rmdir	verbose
delete	mdelete	proxy	runique	?

```
ftp> open 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 2.0.5)
Name (127.0.0.1:andres): andres2
331 Please specify the password.
Password: ****
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 '/home/andres2'
ftp> mkdir OTR0
257 '/home/andres2/OTR0' created
ftp> rmdir OTR0
250 Remove directory operation successful.
ftp> delete tres
550 Delete operation failed.
ftp> delete dos
250 Delete operation successful.
ftp> cd Public
250 Directory successfully changed.
ftp> size uno
550 Could not get file size.
ftp> cd ..
250 Directory successfully changed.
ftp> size uno
213 13
ftp> rename uno dos
550 RNFR command failed.
ftp> rename uno
(to-name) dos
350 Ready for RNT0.
250 Rename successful.
ftp> rename
(from-name) dos
(to-name) uno
```



```
350 Ready for RNT0.
250 Rename successful.
ftp> mode
We only support stream mode, sorry.
ftp> type ascii
200 Switching to ASCII mode.
ftp> type image
200 Switching to Binary mode.
ftp> type EBCDIC
EBCDIC: unknown mode
ftp> bye
221 Goodbye.
```

En el siguiente ejemplo se muestran los comandos ejecutados sobre el emulador de terminal NVT ASCII telnet(1) sin la intervención de un cliente de FTP.

```
? telnet 127.0.0.1 21
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 (vsFTPd 2.0.5)
HELP
530 Please login with USER and PASS.
USER andres2
331 Please specify the password.
PASS 123
230 Login successful.
HELP
214-The following commands are recognized.
  ABOR ACCT ALLO APPE CDUP CWD  DELE EPRT EPSV FEAT HELP LIST MDTM MKD
  MODE NLST NOOP OPTS PASS PASV PORT PWD  QUIT REIN REST RETR RMD  RNFR
  RNT0 SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD
  XPWD XRMD
214 Help OK.
MKD OTRO
257 '/home/andres2/OTRO' created
RMD OTRO
250 Remove directory operation successful.
DELE tres
550 Delete operation failed.
DELE dos
250 Delete operation successful.
PWD
257 '/home/andres2'
CWD Public
250 Directory successfully changed.
```



```
PWD
257 '/home/andres2/Public'
SIZE uno
550 Could not get file size.
CWD ..
250 Directory successfully changed.
SIZE uno
213 13
STAT uno
213-Status follows:
-rw-r--r--    1 1001    1001          13 Sep 15 15:52 uno
213 End of status
RNFR uno dos
550 RNFR command failed.
RNFR uno
350 Ready for RNT0.
RNT0 dos
250 Rename successful.
RNFR dos
350 Ready for RNT0.
RNT0 uno
250 Rename successful.
MODE
504 Bad MODE command.
TYPE A
200 Switching to ASCII mode.
TYPE I
200 Switching to Binary mode.
NOOP
200 NOOP ok.
QUIT
221 Goodbye.
Connection closed by foreign host.
```

En los ejemplos de este documento se utilizó el servidor vsFTPD local. Más adelante, en la pág. 23, se muestra la configuración. En el archivo `/var/log/vsftpd.log` se almacenan los logs. Se muestra un resumen a continuación.

```
# tail -f /var/log/vsftpd.log
Tue Sep 16 16:20:46 2008 [pid 17907] CONNECT: Client '127.0.0.1'
Tue Sep 16 16:20:53 2008 [pid 17906] [andres2] OK LOGIN: Client '127.0.0.1'
...
```

3.3.1. Problemas con Formatos

Si se selecciona un formato no adecuado de transferencia, pueden producirse problemas. Por ejemplo, si se transfiere un archivo binario, como lo es un ejecutable, en formato ASCII, se “romperá”.



Antes de realizar la transferencia se calcula el hash del archivo `/bin/ls`, que es un ejecutable.

```
? md5sum /bin/ls
96fa54ba1486bdc462a7052848c43dee /bin/ls
? file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
for GNU/Linux 2.6.8, dynamically linked (uses shared libs), stripped

? ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 2.0.5)
Name (127.0.0.1:andres): andres2
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> type ascii
200 Switching to ASCII mode.
ftp> put /bin/ls ls.ascii
local: /bin/ls remote: ls.ascii
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 File receive OK.
78257 bytes sent in 0.01 secs (15067.6 kB/s)
ftp> quit
```

Una vez finalizada la transferencia, se calcula nuevamente el hash sobre el archivo enviado. Como se puede observar, el valor obtenido es totalmente diferente al valor obtenido del archivo original. Esto se debe a que el archivo fue modificado durante la transferencia.

```
# md5sum /home/andres2/ls.ascii
87470fdc1b51a6ecebe0161ebc789d72 /home/andres2/ls.ascii

# file /home/andres2/ls.ascii
/home/andres2/ls.ascii: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
for GNU/Linux 2.6.8, dynamically linked (uses shared libs), stripped
```

Al momento de ejecutarlo, recordar que transferimos un archivo ejecutable, terminamos con un dump del proceso.

```
# chmod +x /home/andres2/ls.ascii

? /home/andres2/ls.ascii
Segmentation fault (core dumped)
```

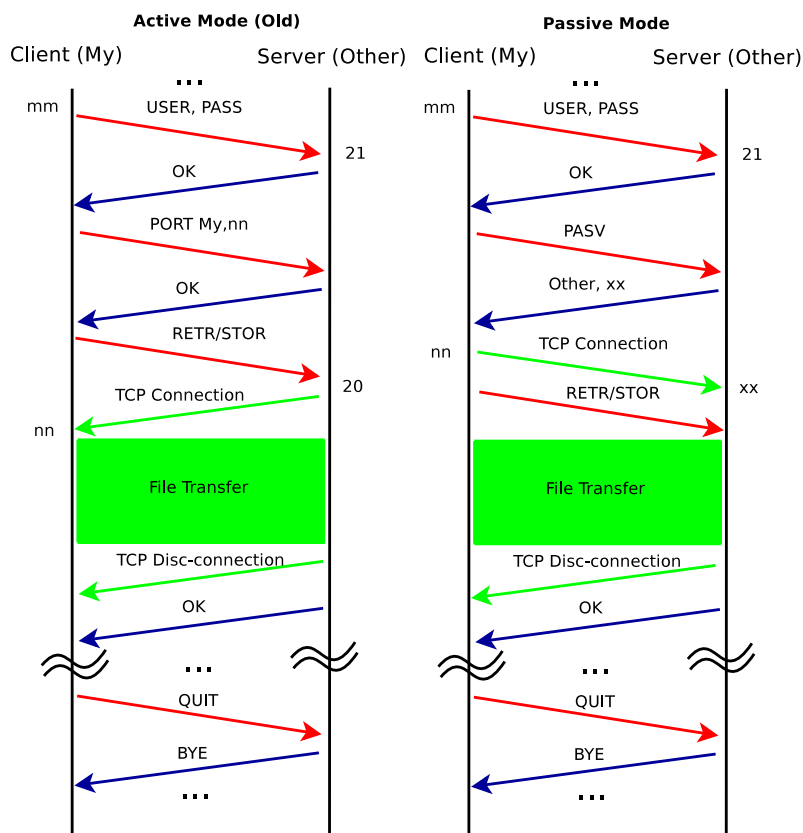


Figura 2: Ejemplo modos FTP

3.4. Modos del Protocolo FTP

FTP tiene la capacidad de trabajar de dos modos diferentes con respecto a la sesión de datos.

Activo: En el modo Activo, el cliente le debe indicar al servidor con el comando `PORT` en que dirección IP y puerto escuchará la conexión de datos. Es activo con respecto al servidor. Este puerto será un puerto no privilegiado o efímero (por encima del 1024). El servidor, desde el puerto número 20, se conectará al puerto indicado por el cliente para establecer la conexión de datos. Esto se realizará por cada operación que requiera una conexión de datos. Éste era el modo default. El formato del comando port es el siguiente:

```
PORT h1,h2,h3,h4,p1,p2
PORT 127,0,0,1,4,3 == 127.0.0.1:1027 (4*256)+3
```

El problema que presenta este modo es que se pueden encontrar complicaciones al realizar NAT (Network Address Translation) o al configurar un firewall que protege a los clientes. Actualmente, no se recomienda su uso, pero existen algunos servidores viejos que solo soportan este modo.

Pasivo: En el modo Pasivo, en lugar de que el servidor se conecte al cliente, es el cliente el que se conecta al servidor. Es pasivo con respecto al servidor. El cliente debe indicarle al servidor que quiere



trabajar con este modo usando el comando PASV. El servidor, como respuesta, indicará el puerto, con el mismo formato que el comando PORT, donde esperará la conexión. Es el modo recomendado actualmente [RFC-1579].

La figura 2 muestra las dos formas de trabajar que tiene FTP. A continuación se muestran dos ejemplos usando los dos modos antes descriptos.

3.4.1. Ejemplos de FTP Activo

Primero, usando el programa tcpdump, se pone a capturar el tráfico.

```
# tcpdump -n -i lo -s 1500 -w 01-ftp-active.pcap
```

La sesión se muestra a continuación. Se establece la conexión de control entre el cliente y el servidor, no se requieren conexiones de datos aún. Como se puede observar, el cliente y el servidor se encuentran en la misma máquina.

```
? ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 2.0.5)
Name (127.0.0.1:andres3): ftptest
530 Please login with USER and PASS.
...
331 Please specify the password.
Password: *****
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Se ejecuta el comando ls/dir (hacen lo mismo) para listar el contenido del directorio remoto. Se utiliza una conexión de datos.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
lrwxrwxrwx    1 1006    1007      26 May 07 18:47 Examples -> /usr/...
-rw-r--r--    1 0      0        9 May 07 18:48 uno
226 Directory send OK.
```

El siguiente paso consiste en traer un archivo desde el servidor con el comando get. Requiere otra conexión de datos.

```
ftp> get uno
local: uno remote: uno
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for uno (9 bytes).
226 File send OK.
9 bytes received in 0.04 secs (0.2 kB/s)
```



Al hacer un `dir/ls` local (eso lo indica con el signo `!` que permite invocar un shell interactivo en la máquina local) no se genera ninguna transferencia, ni de control ni de datos.

```
ftp> !ls
Confiteria Desktop http http.tar.gz internet Linksys Trash UDP_v4.odt uno
```

Usando el comando `put` se envía un archivo al servidor. Necesita una nueva conexión de datos.

```
ftp> put http.tar.gz
local: http.tar.gz remote: http.tar.gz
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 File receive OK.
1624334 bytes sent in 0.11 secs (13994.1 kB/s)
```

Si se ejecuta el comando `rm`, que permite remover un directorio, para eliminar un archivo, se genera un error.

```
ftp> rm uno
550 Remove directory operation failed.
```

Para ver todos los comandos disponibles en el cliente se ejecuta `help`.

```
ftp> help
Commands may be abbreviated.  Commands are:
!          debug          mdir          qc            send
$          dir            mget          sendport      site
account    disconnect          mkdir          put           size
append     exit                  mls           pwd           status
ascii      form                  mode          quit          struct
bell       get                   modtime       quote         system
binary     glob                  mput          recv          sunique
bye        hash                  newer          reget         tenex
case       help                  nmap          rstatus       tick
cd         idle                  nlist         rhelp         trace
cdup       image                 ntrans        rename        type
chmod      lcd                   open          reset         user
close      ls                    prompt         restart       umask
cr         macdef                passive        rmdir         verbose
delete     mdelete               proxy          runique        ?
```

Tampoco se requiere una conexión de datos al borrar un archivo.

```
ftp> del uno
250 Delete operation successful.
```

Por último, se cierra la conexión.



```
ftp> quit
221 Goodbye.
```

A continuación se muestra una sesión en modo activo usando las herramientas `telnet(1)` y `nc/netcat(1)`. La herramienta `nc/netcat(1)` se utiliza para generar la sesión de datos de manera pasiva en el cliente.

Se establece la sesión entre el cliente y el servidor, lo que genera una conexión de control.

```
? telnet 127.0.0.1 21
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 (vsFTPd 2.0.5)
LIST
530 Please login with USER and PASS.
USER andres2
331 Please specify the password.
PASS 123
230 Login successful.
HELP
214-The following commands are recognized.
  ABOR ACCT ALLO APPE CDUP CWD  DELE EPRT EPSV FEAT HELP LIST MDTM MKD
  MODE NLST NOOP OPTS PASS PASV PORT PWD  QUIT REIN REST RETR RMD  RNFR
  RNT0 SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD
  XPWD XRMD
214 Help OK.
```

Luego, se intenta ejecutar un `ls/dir`, pero sin establecer la sesión de datos.

```
LIST
425 Use PORT or PASV first.
PORT 127,0,0,1,4,6
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
426 Failure writing network stream.
```

En otra terminal virtual del mismo host se establece el “servidor” del lado del cliente para la conexión de datos.

```
? nc -l -p 1026
```

Se vuelve a ejecutar el comando, pero esta vez con éxito.

```
PORT 127,0,0,1,4,2
200 PORT command successful. Consider using PASV.
```



0.000	(36127)	36127 > ftp [SYN] S	TCP: 36127 > ftp [SYN] Seq=1456344417 Win=5840 Len=0 MSS=1460 TSV=267025 TSER=0 WS=5
0.003	(36127)	ftp > 36127 [SYN, A]	TCP: ftp > 36127 [SYN, ACK] Seq=1459768666 Ack=1456344418 Win=5792 Len=0 MSS=1460 TSV=269650 TSER=2
0.003	(36127)	36127 > ftp [ACK] S	TCP: 36127 > ftp [ACK] Seq=1456344418 Ack=1459768667 Win=183 Len=0 TSV=267025 TSER=269650
0.017	(36127)	Response: 220 (vsFTP	FTP: Response: 220 (vsFTPd 2.0.3)
0.018	(36127)	36127 > ftp [ACK] S	TCP: 36127 > ftp [ACK] Seq=1456344418 Ack=1459768687 Win=183 Len=0 TSV=267029 TSER=269653
1.962	(36127)	Request: USER andres	FTP: Request: USER andres
1.962	(36127)	ftp > 36127 [ACK] S	TCP: ftp > 36127 [ACK] Seq=1459768687 Ack=1456344431 Win=181 Len=0 TSV=270140 TSER=267515
1.963	(36127)	Response: 331 Please	FTP: Response: 331 Please specify the password
1.963	(36127)	36127 > ftp [ACK] S	TCP: 36127 > ftp [ACK] Seq=1456344431 Ack=1459768721 Win=183 Len=0 TSV=267515 TSER=270140
4.619	(36127)	Request: PASS 12345	FTP: Request: PASS 123456
4.663	(36127)	ftp > 36127 [ACK] S	TCP: ftp > 36127 [ACK] Seq=1459768721 Ack=1456344444 Win=181 Len=0 TSV=270815 TSER=268180
4.749	(36127)	Response: 230 Login	FTP: Response: 230 Login successful.
4.752	(36127)	36127 > ftp [ACK] S	TCP: 36127 > ftp [ACK] Seq=1456344444 Ack=1459768744 Win=183 Len=0 TSV=268212 TSER=270836
4.752	(36127)	Request: SYST	FTP: Request: SYST
4.752	(36127)	ftp > 36127 [ACK] S	TCP: ftp > 36127 [ACK] Seq=1459768744 Ack=1456344450 Win=181 Len=0 TSV=270837 TSER=268212
4.753	(36127)	Response: 215 UNIX	FTP: Response: 215 UNIX Type: L8
4.796	(36127)	36127 > ftp [ACK] S	TCP: 36127 > ftp [ACK] Seq=1456344458 Ack=1459768763 Win=183 Len=0 TSV=268223 TSER=270837
37.013	(36127)	Request: TYPE I	FTP: Request: TYPE I
37.014	(36127)	Response: 200 Switc	FTP: Response: 200 Switching to Binary mode.
37.015	(36127)	36127 > ftp [ACK] S	TCP: 36127 > ftp [ACK] Seq=1456344458 Ack=1459768794 Win=183 Len=0 TSV=276274 TSER=278898
37.015	(36127)	Request: PORT 172,2	FTP: Request: PORT 172,20,1,101,197,149
37.015	(36127)	Response: 200 PORT	FTP: Response: 200 PORT command successful. C
37.015	(36127)	Request: RETR uno	FTP: Request: RETR uno
37.017	(50581)	ftp-data > 50581 [S]	TCP: ftp-data > 50581 [SYN] Seq=2038782054 Win=5840 Len=0 MSS=1460 TSV=278898 TSER=0 WS=5
37.019	(50581)	50581 > ftp-data [S]	TCP: 50581 > ftp-data [SYN, ACK] Seq=2039831331 Ack=2038782055 Win=5792 Len=0 MSS=1460 TSV=276275 T
37.019	(50581)	ftp-data > 50581 [A]	TCP: ftp-data > 50581 [ACK] Seq=2038782055 Ack=2039831332 Win=183 Len=0 TSV=278899 TSER=276275
37.020	(36127)	Response: 150 Openi	FTP: Response: 150 Opening BINARY mode data c
37.022	(50581)	FTP Data: 1448 byte	FTP-DATA: FTP Data: 1448 bytes
37.022	(50581)	FTP Data: 1448 byte	FTP-DATA: FTP Data: 1448 bytes
37.025	(50581)	50581 > ftp-data [A]	TCP: 50581 > ftp-data [ACK] Seq=2039831332 Ack=2038783503 Win=272 Len=0 TSV=276276 TSER=278900

Figura 3: FTP Activo.

LIST

150 Here comes the directory listing.
226 Directory send OK.

En el proceso escuchando en el puerto 1026 se obtiene la salida del dir/ls.

```
? nc -l -p 1026
```

```
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Desktop
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Documents
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Music
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Pictures
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Public
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Templates
drwxr-xr-x  2 1001  1001      4096 Aug 12  2007 Videos
drwxr-xr-x 12 1001  1001      4096 Apr 18  2006 mozilla
```

```
?
```

Se crea un archivo en el servidor con el siguiente contenido.

```
? vi uno
```



```
uno
dos
tres
```

Se establece otro “servidor” para la sesión de datos.

```
? nc -l -p 1027
```

Se realiza la transferencia.

```
STAT uno
213-Status follows:
-rw-r--r--    1 1001    1001          13 Sep 15 15:52 uno
213 End of status
```

```
PORT 127,0,0,1,4,3
200 PORT command successful. Consider using PASV.
```

```
RETR uno
150 Opening BINARY mode data connection for uno (13 bytes).
226 File send OK.
```

En el “servidor” establecido para la sesión de datos del lado del cliente se puede observar la recepción del archivo.

```
? nc -l -p 1027
uno
dos
tres
?
```

Finalmente, se cierra la sesión de control.

```
QUIT
221 Goodbye.
Connection closed by foreign host.
```

3.4.2. Ejemplos de FTP Pasivo

Al igual que en el modo activo, primero se pone a capturar el tráfico.

```
# tcpdump -n -i lo -s 1500 -w 02-ftp-passive.pcap
```

El primer paso es establecer una conexión entre el cliente y el servidor en modo pasivo.

```
? ftp -p 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 2.0.5)
```



```
Name (127.0.0.1:andres3): ftptest
530 Please login with USER and PASS.
...
331 Please specify the password.
Password: ****
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Al hacer un `dir/ls` se genera una conexión de datos en modo pasivo.

```
ftp> ls
227 Entering Passive Mode (127,0,0,1,148,211)
150 Here comes the directory listing.
lrwxrwxrwx    1 1006    1007          26 May 07 18:47 Examples -> /usr/...
-rw-----    1 1006    1007    1624334 May 08 01:52 http.tar.gz
226 Directory send OK.
```

También se establece una nueva conexión de datos cuando se baja un archivo usando el comando `get`.

```
ftp> get http.tar.gz
local: http.tar.gz remote: http.tar.gz
227 Entering Passive Mode (127,0,0,1,120,149)
150 Opening BINARY mode data connection for http.tar.gz (1624334 bytes).
226 File send OK.
1624334 bytes received in 0.01 secs (174621.7 kB/s)
```

Para borrar un archivo no se requiere una nueva conexión de datos.

```
ftp> del http.tar.gz
250 Delete operation successful.
ftp> quit
221 Goodbye.
```

A continuación se muestra una sesión en modo pasivo usando las herramientas `telnet(1)` y `nc/netcat(1)`.

```
? telnet 127.0.0.1 21
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 (vsFTPd 2.0.5)
USER andres2
331 Please specify the password.
PASS 123
230 Login successful.
PASV
227 Entering Passive Mode (127,0,0,1,133,205)
```




Se pone a escuchar un proceso en el puerto indicado del lado del servidor. Se realiza la conexión que queda en espera de recibir información.

```
# netstat -atnp | grep 34253
tcp        0      0 127.0.0.1:34253      0.0.0.0:*    LISTEN    18235/vsftpd

? telnet 127.0.0.1 34253
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

Se solicita la información.

```
RETR uno
150 Opening BINARY mode data connection for uno (13 bytes).
226 File send OK.
```

La conexión de datos recibe la información y el servidor cierra el puerto.

```
# telnet 127.0.0.1 34253
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
uno
dos
tres
Connection closed by foreign host.
```

Un ejemplo con nc/netcat(1). Se usa otro puerto, otra sesión de datos sobre la misma de control.

```
PASV
227 Entering Passive Mode (127,0,0,1,127,109)
```

Se conecta.

```
? nc 127.0.0.1 32621
```

Solicita el contenido.

```
RETR uno
150 Opening BINARY mode data connection for uno (13 bytes).
226 File send OK.
```

Se recibe y el servidor cierra la conexión.

```
? nc 127.0.0.1 32621
uno
dos
tres
?
```



Un ejemplo enviando un archivo. Se solicita el puerto primero.

PASV

227 Entering Passive Mode (127,0,0,1,223,161)

```
# netstat -atnp | grep ftp
tcp        0      0 127.0.0.1:57249      0.0.0.0:*             LISTEN      18273/vsftpd
tcp        0      0 0.0.0.0:21           0.0.0.0:*             LISTEN      17824/vsftpd
tcp        0      0 127.0.0.1:21         127.0.0.1:55798       ESTABLISHED 18271/vsftpd
```

Se crea el archivo local y se manda por la conexión. Del otro lado no se leen datos hasta que no se indica con el comando STOR.

```
? cat otro
III
III
? cat otro | nc 127.0.0.1 57249
```

```
# netstat -atnp | grep ftp
tcp        0      0 127.0.0.1:57249      0.0.0.0:*             LISTEN      18273/vsftpd
tcp        0      0 0.0.0.0:21           0.0.0.0:*             LISTEN      17824/vsftpd
tcp        0      0 127.0.0.1:21         127.0.0.1:55798       ESTABLISHED 18271/vsftpd
```

Se indica que se lea el archivo, lo que comienza y termina la transferencia. La sesión queda abierta, el encargado de cerrarla será el cliente.

STOR otro.bak

150 Ok to send data.

226 File receive OK.

```
# netstat -atnp | grep ftp
tcp        0      0 127.0.0.1:57249      0.0.0.0:*             LISTEN      18273/vsftpd
tcp        0      0 0.0.0.0:21           0.0.0.0:*             LISTEN      17824/vsftpd
tcp        0      0 127.0.0.1:21         127.0.0.1:55798       ESTABLISHED 18271/vsftpd
tcp        0      0 127.0.0.1:53045      127.0.0.1:55798       ESTABLISHED 18273/vsftpd
```

226 File receive OK.

CTRL+C

?

```
# netstat -atnp | grep TIME
tcp        0      0 127.0.0.1:55798      127.0.0.1:53045       TIME_WAIT
```

En el servidor vsFTPD se puede configurar qué rango de puertos se pueden utilizar para las conexiones pasivas. Para este propósito se utilizan las siguientes directivas en el archivo de configuración.

pasv_max_port:



The maximum port to allocate for PASV style data connections.
Can be used to specify a narrow port range to assist firewalling.
Default: 0 (use any port)

pasv_min_port:

The minimum port to allocate for PASV style data connections.
Can be used to specify a narrow port range to assist firewalling.

```

0.000 (57002) 57002 > ftp [SYN] S (21)
0.003 (57002) ftp > 57002 [SYN, A] (21)
0.003 (57002) 57002 > ftp [ACK] S (21)
0.013 (57002) Response: 220 (vsFT (21)
0.015 (57002) 57002 > ftp [ACK] S (21)
1.975 (57002) Request: USER andre (21)
1.976 (57002) ftp > 57002 [ACK] S (21)
1.978 (57002) Response: 331 Please (21)
1.980 (57002) 57002 > ftp [ACK] S (21)
3.990 (57002) Request: PASS 12345 (21)
4.005 (57002) Response: 230 Login (21)
4.006 (57002) 57002 > ftp [ACK] S (21)
4.006 (57002) Request: SYST (21)
4.008 (57002) Response: 215 UNIX (21)
4.050 (57002) 57002 > ftp [ACK] S (21)
11.730 (57002) Request: TYPE I (21)
11.734 (57002) Response: 200 Switc (21)
11.734 (57002) 57002 > ftp [ACK] S (21)
11.734 (57002) Request: PASV (21)
11.734 (57002) Response: 227 Enter (21)
11.734 (52743) 52743 > htulsrv [S (5023)
11.734 (52743) htulsrv > 52743 [S (5023)
11.734 (52743) 52743 > htulsrv [A (5023)
11.734 (57002) Request: RETR ugo (21)
11.735 (57002) Response: 150 Openi (21)
11.735 (52743) htulsrv > 52743 [A (5023)
11.736 (52743) htulsrv > 52743 [A (5023)
11.736 (52743) 52743 > htulsrv [A (5023)

```

```

TCP: 57002 > ftp [SYN] Seq=831336350 Win=5840 Len=0 MSS=1460 TSV=393836 TSER=0 WS=5
TCP: ftp > 57002 [SYN, ACK] Seq=830010892 Ack=831336351 Win=5792 Len=0 MSS=1460 TSV=396463 TSER=393
TCP: 57002 > ftp [ACK] Seq=831336351 Ack=830010893 Win=183 Len=0 TSV=393836 TSER=396463
FTP: Response: 220 (vsFTPd 2.0.3)
TCP: 57002 > ftp [ACK] Seq=831336351 Ack=830010913 Win=183 Len=0 TSV=393839 TSER=396465
FTP: Request: USER andres
TCP: ftp > 57002 [ACK] Seq=830010913 Ack=831336364 Win=181 Len=0 TSV=396956 TSER=394329
FTP: Response: 331 Please specify the passwor
TCP: 57002 > ftp [ACK] Seq=831336364 Ack=830010947 Win=183 Len=0 TSV=394330 TSER=396956
FTP: Request: PASS 123456
FTP: Response: 230 Login successful.
TCP: 57002 > ftp [ACK] Seq=831336377 Ack=830010970 Win=183 Len=0 TSV=394836 TSER=397463
FTP: Request: SYST
FTP: Response: 215 UNIX Type: L8
TCP: 57002 > ftp [ACK] Seq=831336383 Ack=830010989 Win=183 Len=0 TSV=394847 TSER=397464
FTP: Request: TYPE I
FTP: Response: 200 Switching to Binary mode.
TCP: 57002 > ftp [ACK] Seq=831336391 Ack=830011020 Win=183 Len=0 TSV=396768 TSER=399395
FTP: Request: PASV
FTP: Response: 227 Entering Passive Mode (172
TCP: 52743 > htulsrv [SYN] Seq=1011613631 Win=5840 Len=0 MSS=1460 TSV=396768 TSER=0 WS=5
TCP: htulsrv > 52743 [SYN, ACK] Seq=1013529088 Ack=1011613632 Win=5792 Len=0 MSS=1460 TSV=399395 T
TCP: 52743 > htulsrv [ACK] Seq=1011613632 Ack=1013529089 Win=183 Len=0 TSV=396768 TSER=399395
FTP: Request: RETR ugo
FTP: Response: 150 Opening BINARY mode data c
TCP: htulsrv > 52743 [ACK] Seq=1013529089 Ack=1011613632 Win=181 Len=1448 TSV=399396 TSER=396768
TCP: htulsrv > 52743 [ACK] Seq=1013530537 Ack=1011613632 Win=181 Len=1448 TSV=399396 TSER=396768
TCP: 52743 > htulsrv [ACK] Seq=1011613632 Ack=1013530537 Win=273 Len=0 TSV=396769 TSER=399396

```

Figura 4: FTP Pasivo.

3.5. FTP Anónimo

El protocolo FTP permite conectarse a un servidor sin necesidad de poseer una cuenta. Una de las formas más populares de usar FTP es la modalidad anónima: **Anonymous FTP**, la cual admite “logonearse” sin necesidad de una cuenta. Esta modalidad es, generalmente, en modo de solo lectura, es decir, sólo se pueden obtener archivos pero no eliminarlos. Usualmente, se suele encontrar un directorio pub/incoming en el cual se pueden dejar archivos hasta cierto tamaño. Cuando se accede a un servidor con esta modalidad se usa, como nombre de usuario, el texto **anonymous** y como contraseña, la dirección de e-mail. Para habilitar una sesión de este tipo es necesario reconfigurar el servidor.

```
# vi /etc/vsftpd.conf
...
```



```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
...
#### EOF ####
```

Es importante, al utilizar FTP anónimo, restringir el acceso a un directorio en particular, no permitiendo que el usuario anónimo se pueda cambiar a otro directorio o listar contenidos del sistema. Es por esto que se dedica un directorio en particular para este propósito. Este directorio es, en el siguiente ejemplo, /home/pub, pero puede ser cualquier otro que decida el usuario.

```
# grep ftp /etc/passwd
ftp:x:116:65534::/home/ftp:/bin/false

# ls -l -d /home/ftp
drwxr-xr-x 2 root nogroup 4096 2008-09-19 11:29 /home/ftp

# /etc/init.d/vsftpd start
```

Se empieza a capturar tráfico y se establece la conexión.

```
# tcpdump -n -i lo -s 1500 -w 03-ftp-anon.pcap

? ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPD 2.0.5)
Name (127.0.0.1:andres3): anonymous
...
331 Please specify the password.
Password: barbieri@cespi.unlp.edu.ar
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.
226 Directory send OK.
ftp> cd /
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> quit
221 Goodbye.
```



3.6. FTP sobre SSL, FTPS

Como se puede observar en los ejemplos, la información confidencial como las contraseñas o, incluso, el contenido de los archivos viaja en la red sin cifrar. SSL puede montarse sobre FTP de forma similar como sucede con HTTP. Más adelante, en la pág. 23, se muestra la configuración de FTP con soporte de SSL. A continuación se muestra la sesión.

Si nos conectamos con un cliente que no tiene soporte de SSL, obtendremos un error.

```
# tcpdump -i lo -s 1500 -w 04-ftp-ssl.pcap
? ftp 127.0.0.1 # /// o puede ser ftp -z nssl 127.0.0.1
Connected to 127.0.0.1.

220-----
220- Server FTP ...
220-
220-          -----
220-          ATENCION
220-          -----
220-
220- Este equipo pertenece a la red de la ....
220- El acceso a este equipo por usuarios no autorizados
220- ...
220-
220- Todos los accesos que se realicen a este sistema
220- podrán ser monitoreados.
220-
220- Gracias y que tenga un buen día.
220-
220- Soporte Técnico
220-
220
Name (127.0.0.1:andres): andres2
530 Non-anonymous sessions must use encryption.
Login failed.
ftp> quit
221 Goodbye.
```

En cambio, si usamos un cliente con soporte de SSL todo funciona OK.

```
? ftp-ssl 127.0.0.1 # /// 0 directamente ftp 127.0.0.1
Connected to 127.0.0.1.

220-----
220- Server FTP ...
220-
220-          -----
```



```
220- ATENCION
220- -----
220-
220- Este equipo pertenece a la red de la ....
220- El acceso a este equipo por usuarios no autorizados
    ...
220-
220- Todos los accesos que se realicen a este sistema
220- podrán ser monitoreados.
220-
220- Gracias y que tenga un buen día.
220-
220- Soporte Técnico
220-
220
Name (127.0.0.1:andres): andres2
234 Proceed with negotiation.
[SSL Cipher DES-CBC3-SHA]
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.
...
226 Directory send OK.
ftp> quit
221 Goodbye.
```

Para FTP sobre SSL existen puertos asignados especialmente.

```
? grep ftps /etc/services
ftps-data      989/tcp      # FTP over SSL (data)
ftps           990/tcp
```

Como en este caso se utiliza TLS, directamente se ejecuta sobre el port 21.

4. Ejemplo, Configuración vsFTPD

4.1. Configuración Básica

Se muestra la configuración inicial del servidor FTP. Se utilizó para los ejemplos la versión `vstftpd(8)` sobre GNU/Linux.



```
# uname
Linux

# vsftpd -v
vsftpd: version 2.0.5

# cat /etc/vsftpd.conf
### Created By Andres 2008-10-16 (C) ###

# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES

# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO

# Uncomment this to allow local users to log in.
local_enable=YES

# Uncomment this to enable any form of FTP write command.
write_enable=YES

# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES

#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES

#### EOF ####

# /etc/init.d/vsftpd start

# netstat -atnp | grep 21
tcp          0      0 0.0.0.0:21      0.0.0.0:*    LISTEN      6943/vsftpd
```

4.2. Configuración con SSL

```
# cat /etc/vsftpd.conf
### Created By Andres 2008-10-16 (C) ###

# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
```



```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO

# Uncomment this to allow local users to log in.
local_enable=YES

# Uncomment this to enable any form of FTP write command.
write_enable=YES

# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES

#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES

# Banner de Acceso
banner_file=/etc/vsftpd.banner.txt

##### SSL #####

ssl_enable=YES
#dsa_cert_file=
rsa_cert_file=/etc/ssl/private/ftps-srv.pem

#Disable support for TLSv1
###ssl_tlsv1=YES

#Do you want to enable support SSLv2? The default is NO. There's no
# reason to uncomment this unless your clients can't talk TLS.
# Support SSLv2
###ssl_sslv2=YES

#Do you want to enable support SSLv3? The default is NO.
#There's no reason to uncomment this unless your clients can't talk TLS.
# Support SSLv3
###ssl_sslv3=YES

###ssl_ciphers=des3,des,rc4...

force_local_logins_ssl=YES
force_local_data_ssl=YES

#### EOF ####
```




```
# /etc/init.d/vsftpd start

# netstat -atnp | grep 21
tcp        0      0 0.0.0.0:21          0.0.0.0:*          LISTEN     7923/vsftpd
```

La alternativa más corta para crear un certificado de prueba es crear un “Self-Signed Certificate” de la siguiente forma, sin usar una CA.

```
# cd /etc/ssl/private/

# openssl req -new -key ftps-srv.key -out ../certs/ftps-srv.csr
...
# openssl x509 -req -in ../certs/ftps-srv.csr -out ../certs/ftp-self.crt \
    -signkey ftps-srv.key
...

# cat ../certs/ftp-self.crt ftps-srv.key > ftps-srv.pem
```

Tener la precaución de asignar los permisos adecuados a las claves privadas. Es importante que en el archivo PEM esté la clave pública y la clave privada. Si sólo está la clave pública, al intentar arrancar el servicio, dará el siguiente error.

```
# vsftpd
500 00PS: SSL: cannot load RSA private key
```

4.3. Usos de FTP

En la actualidad, el protocolo FTP ha caído en desuso a causa del avance de HTTP. En 1992 era el protocolo que más datos transportaba en la Internet. La desventaja de HTTP, con respecto a FTP, es que fue pensado para trabajar más como protocolo de descarga (download) que de carga (upload), en cambio, FTP tiene los dos roles. Si bien HTTP, a partir de otros comandos y programación extra, permite ser usado como protocolo de carga FTP parece ser más eficiente para grandes archivos. La gran ventaja de HTTP es la facilidad que ofrece ante la existencia de un firewall o servicios de NAT en redes que utilizan este servicio, para FTP esta labor es más difícil. Otra ventaja de HTTP parece ser que sus conexiones consumen menos recursos.

Con respecto a su uso como de protocolo de descarga, FTP sigue estando disponible en la mayoría de los sitios donde los archivos para descargar suelen ser grandes, de gran cantidad o donde hay muchas descargas. La ventaja que tenía FTP ante HTTP era que permitía resumir una descarga. El protocolo FTP en 2017 se lo encuentra en sites de descargas de distribuciones de sistemas operativos, como los mirrors de OpenBSD, de la distribución Scientific Linux, de software como SunSITE (Sun Software, Information & Technology Exchange). Pero de alguna forma podemos ver que va perdiendo fuerza, por ejemplo en este mismo año, 2017, se quitó el soporte de los mirrors de los Kernel de GNU/Linux, dejando solo RSYNC, GIT y HTTPS.



Referencias

- [StevI] TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994. W. Richard Stevens.
- [StevI2] TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, (2nd. Ed). 2011. Kevin R. Fall, W. Richard Stevens.
- [KR] Computer Networking: A Top-Down Approach, Addison-Wesley, (6th Edition). 2012. Kurose/Ross.
- [LX] The Linux Home Page: <http://www.linux.org/>.
- [Siever] Linux in a Nutshell, Fourth Edition June, 2003. O'Reilly. Ellen Siever, Stephen Figgins, Aaron Weber.
- [FZ] FileZilla Project. <http://filezilla-project.org/>.
- [VSFTP] vsftpd <http://vsftpd.beasts.org/>.
- [RFC-793] <http://www.rfc-editor.org/rfc/rfc793.txt>. TCP Transmission Control Protocol (Jon Postel 1981 USC-ISI IANA).
- [RFC-114] A FILE TRANSFER PROTOCOL. A. Bhushan (MIT), 1971.
<http://tools.ietf.org/html/rfc114>.
- [RFC-765] FILE TRANSFER PROTOCOL. J. Postel (ISI), 1980. <http://tools.ietf.org/html/rfc765>.
- [RFC-959] FILE TRANSFER PROTOCOL. J. Postel, J. Reynolds (ISI), 1985.
<http://tools.ietf.org/html/rfc959>.
- [RFC-1579] Firewall-Friendly FTP, S. Bellovin (AT&T), 1995. <http://tools.ietf.org/html/rfc1579>.
- [RFC-2228] FTP Security Extensions. M. Horowitz (Cygnus Solutions), S. Lunt (Bellcore), 1997.
<http://tools.ietf.org/html/rfc2228>.
- [KORG-FTP] <https://www.kernel.org/shutting-down-ftp-services.html>.
- [OpenSSL] OpenSSL project: <http://www.openssl.org/>.
- [COM05] Ethereal, Wireshark. Autor original Gerald Combs, 2005.
<http://www.ethereal.com/>.
<http://www.wireshark.org/>.



Índice

1. Aclaración para el Lector	1
2. Introducción	1
3. Protocolo FTP	2
3.1. Ejemplos de Respuestas FTP	3
3.2. Conexiones del Protocolo FTP	4
3.3. Comandos sin sesión de datos	5
3.3.1. Problemas con Formatos	8
3.4. Modos del Protocolo FTP	10
3.4.1. Ejemplos de FTP Activo	11
3.4.2. Ejemplos de FTP Pasivo	15
3.5. FTP Anónimo	19
3.6. FTP sobre SSL, FTPS	21
4. Ejemplo, Configuración vsFTPD	22
4.1. Configuración Básica	22
4.2. Configuración con SSL	23
4.3. Usos de FTP	25

Índice de figuras

1. FileZilla y FTP vía web browser.	2
2. Ejemplo modos FTP	10
3. FTP Activo.	14
4. FTP Pasivo.	19