

HES-SO Valais Wallis

File Transfer Protocol

Digital Team Academy

Dasek Joiakim

19/10/2022

Table of contents

Study the protocol.....	2
Usage of FTP	4
Upload and Download a file.....	6
Upload.....	6
Download	7
What are active/passive modes.....	8
Data port	9
1. During the Download, which server's port was used to transfer the file.....	10
2. During the Upload, which server's port was used to transfer the file.	10
Create a folder.....	11
Download a file	14
Upload a file	15

Study the protocol¹

FTP stands for file transfer protocol, and it refers to a set of rules that define how computers transfer files to each other. It is used, among other things, by developers to upload their source files to web hosts, which is a remote server. This protocol is designed on a client-server model, the server has the server-like tool installed on its system and the client has the client-like ftp on its system.

Here is the complete suite of the FTP tool, its arguments and their definitions:²

FTP Command	Description of Command
!	This command toggles back and forth between the operating system and ftp. Once back in the operating system, typing exit takes you back to the FTP command line.
?	Accesses the Help screen.
append	Append text to a local file.
ascii	Switch to ASCII transfer mode.
bell	Turns bell mode on or off.
binary	Switches to binary transfer mode.
bye	Exits from FTP.
cd	Changes directory.
close	Exits from FTP.
delete	Deletes a file.
debug	Sets debugging on or off.
dir	Lists files, if connected.
	dir -C = lists the files in wide format.
	dir -1 = Lists the files in bare format in alphabetic order.
	dir -r = Lists directory in reverse alphabetic order.
	dir -R = Lists all files in current directory and sub directories.
	dir -S = Lists files in bare format in alphabetic order.
disconnect	Exits from FTP.
get	Get file from the remote computer.
glob	Sets globbing on or off. When turned off, the file name in the put and get commands is taken literally, and wildcards will not be looked at.
hash	Sets hash mark printing on or off. When turned on, for each 1024 bytes of data received, a hash-mark (#) is displayed.
help	Accesses the Help screen and displays information about the command if the command is typed after help.
lcd	Displays local directory if typed alone or if path typed after lcd will change the local directory.
literal	Sends a literal command to the connected computer with an expected one-line response.
ls	Lists files of the remotely connected computer.

¹ <https://www.techtarget.com/searchnetworking/definition/File-Transfer-Protocol-FTP>

² <https://www.serv-u.com/ftp-server-windows/commands>

mdelete	Multiple delete.
mdir	Lists contents of multiple remote directories.
mget	Get multiple files.
mkdir	Make directory.
mls	Lists contents of multiple remote directories.
mput	Send multiple files.
open	Opens address.
prompt	Enables or disables the prompt.
put	Send one file.
pwd	Print working directory.
quit	Exits from FTP.
quote	Same as the literal command.
recv	Receive file.
remotehelp	Get help from remote server.
rename	Renames a file.
rmdir	Removes a directory on the remote computer.
send	Send single file.
status	Shows status of currently enabled and disabled options.
trace	Toggles packet tracing.
type	Set file transfer type.
user	Send new user information.
verbose	Sets verbose on or off.

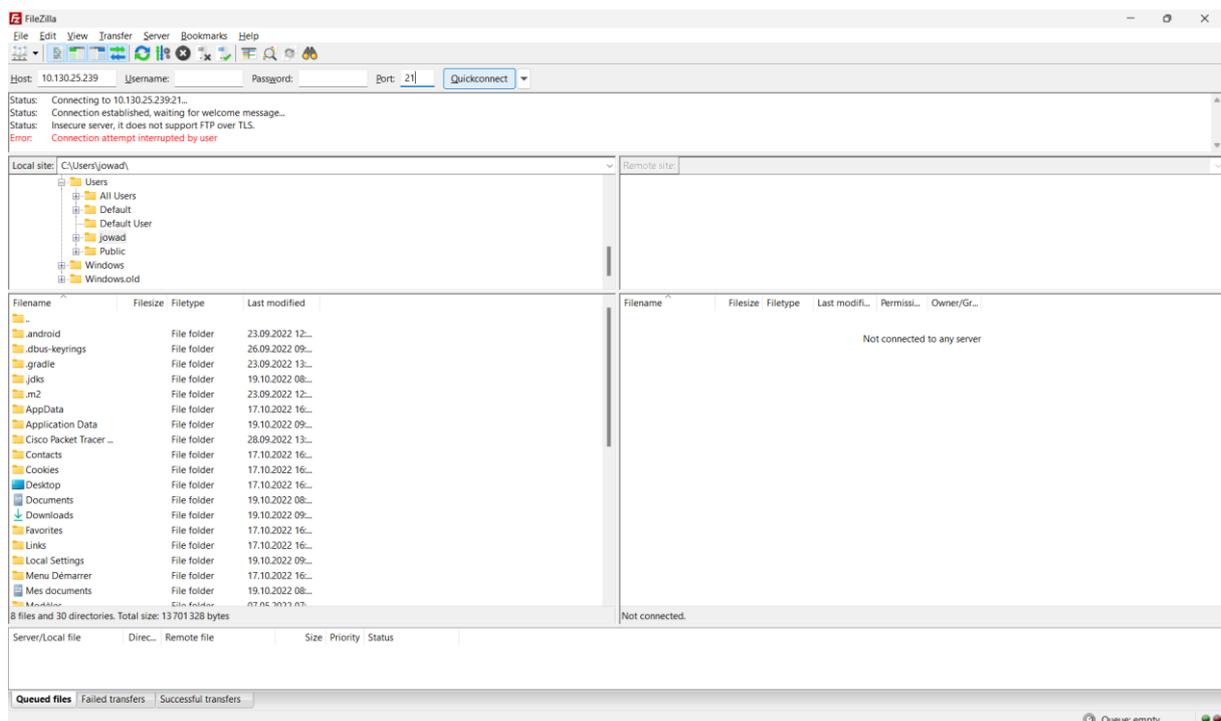
We can refer to this list for the rest of the exercises. all these arguments allow us to activate a function of the FTP tool on our terminal.

Usage of FTP

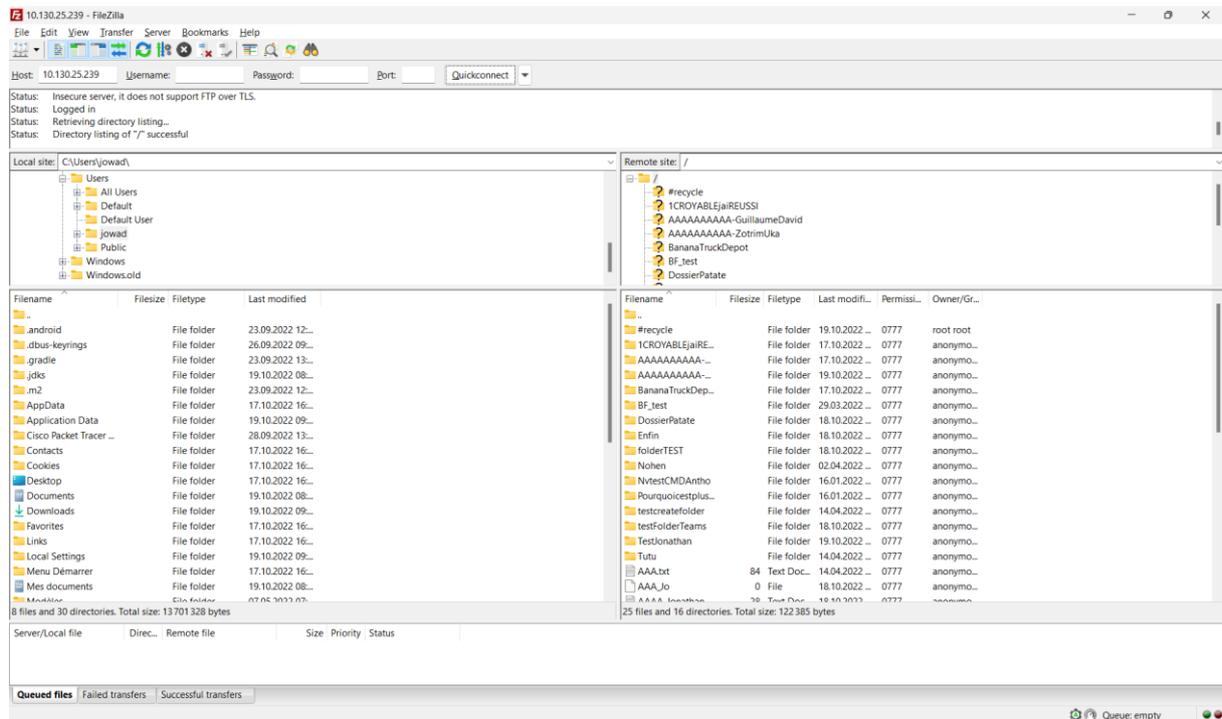
First thing to do, you must install FileZilla-client to be able to connect to the FTP server, here is the official link: [Download link](#). After installing it we can talk a little about the graphical interface of this software. We see a bar with the labels Host, username, password and port. These elements are essential to establish a connection between the client and the server.

The IP address is provided, we could also insert a domain name which is therefore linked with the IP address, without authentication, and the default port is 21. You can establish a connection by pressing "quickconnect".

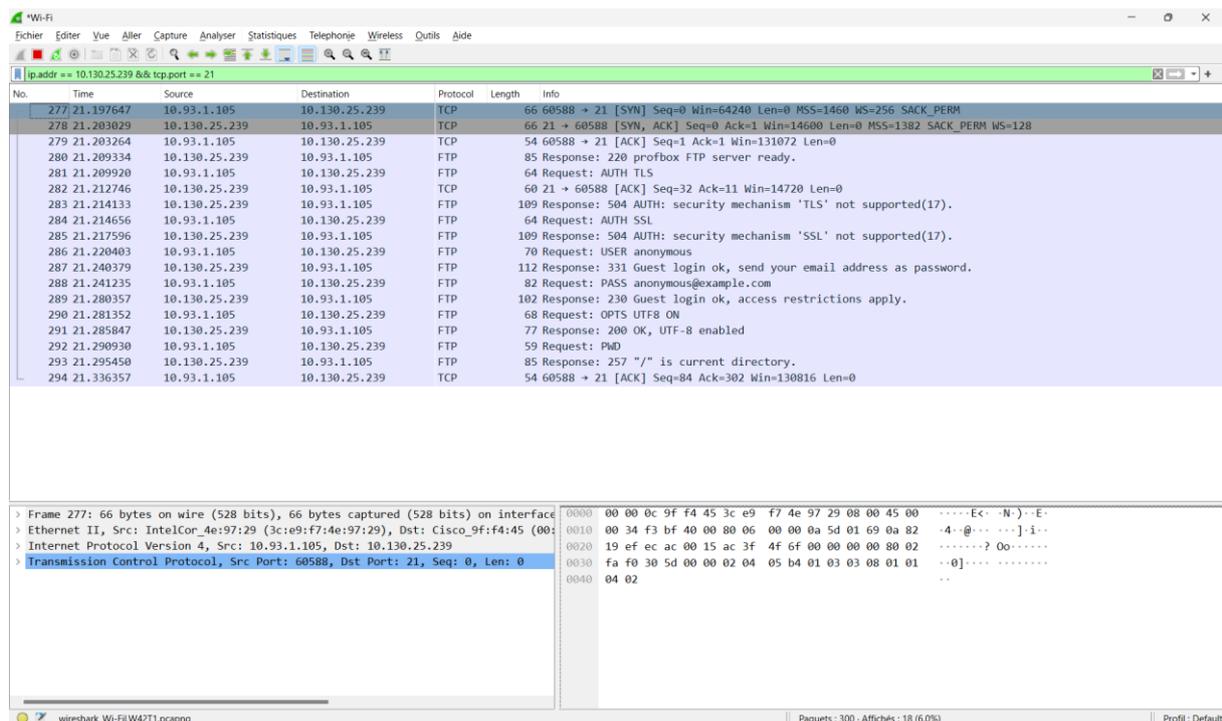
Under this same bar is the verbose console. The left side is the local file system, i.e., the client side, and the right side is the file explorer on the server side. Under this part we have the inside of the tree of the folder selected in the upper part. And for the last part of this software is the historical or pending queue of files transiting between client and server.



We can see the connection established in the following image:



Wireshark let us know the exchange of requests between the client and the server. In these unsecured exchanges, we find the anonymous connection and the default password, as well as the port, the commands of connection, file listing etc...



Upload and Download a file

Upload

Create an empty file on the server side as follows and name it 01test.txt. Wireshark will capture this request. In the blue line, selected in the Wireshark image we can see the command "STOR" which allows to create a file on the server side.

The screenshot displays two applications: FileZilla and Wireshark. FileZilla is in the foreground, showing a local site at C:\Users\jowad\ and a remote site at 10.130.25.239. A dialog box titled "Create empty file" is open, prompting for a filename, with "01test.txt" entered. The FileZilla interface shows a list of files and folders on the remote site, including folders like #recycle, #CROVABLEjaREUSS, and files like #AAAAA-A-GuillaumeDavid. The bottom status bar of FileZilla shows "Queued files: Failed transfers: Successful transfers".

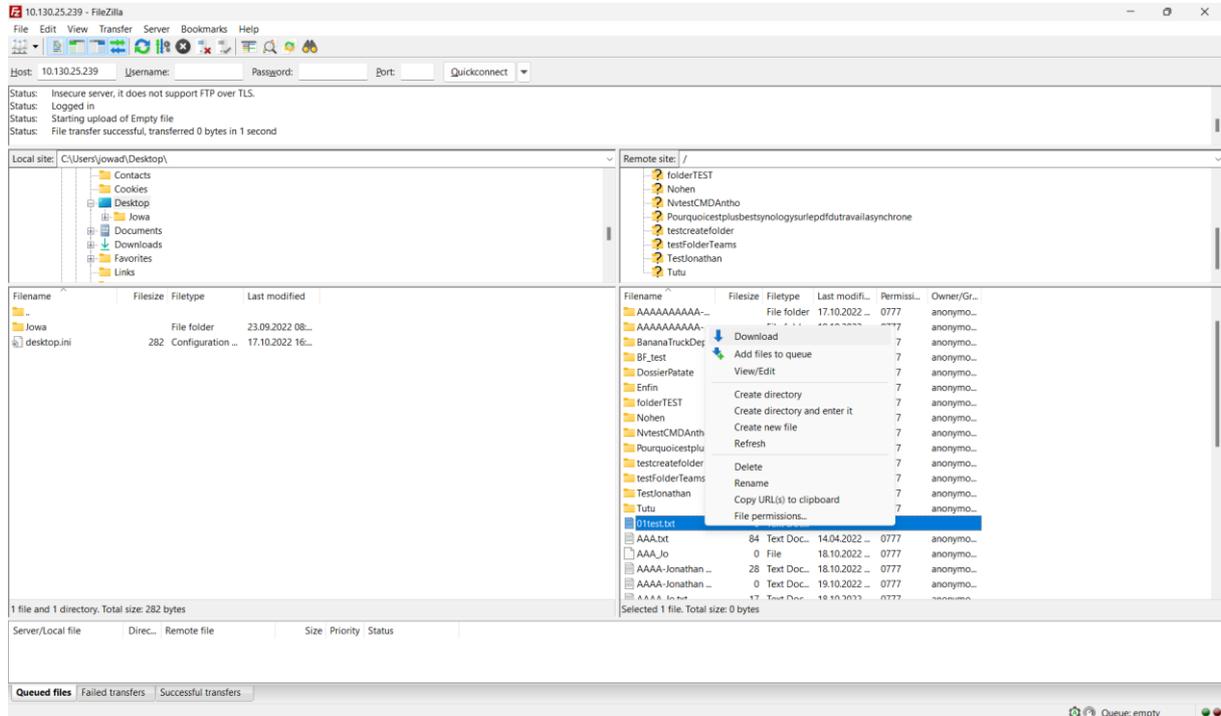
Wireshark is in the background, capturing network traffic. The filter is set to "ip.addr == 10.130.25.239 && tcp.port == 21". The packet list shows a sequence of FTP commands and responses. The selected packet (No. 71) is a "Request: STOR 01test.txt" from 10.93.1.105 to 10.130.25.239. The packet details pane shows the File Transfer Protocol (FTP) structure, including the "Request command: STOR" and "Request arg: 01test.txt". The packet bytes pane shows the raw data of the request.

No.	Time	Source	Destination	Protocol	Length	Info
389	70.425907	10.93.1.105	10.130.25.239	FTP	64	Request: AUTH TLS
390	70.428841	10.130.25.239	10.93.1.105	TCP	60	21 → 60606 [ACK] Seq=32 Ack=11 Win=14720 Len=0
391	70.428841	10.130.25.239	10.93.1.105	FTP	109	Response: 504 AUTH: security mechanism 'TLS' not supported(17).
392	70.429096	10.93.1.105	10.130.25.239	FTP	64	Request: AUTH SSL
393	70.432321	10.130.25.239	10.93.1.105	FTP	109	Response: 504 AUTH: security mechanism 'SSL' not supported(17).
394	70.434456	10.93.1.105	10.130.25.239	FTP	70	Request: USER anonymous
395	70.451511	10.130.25.239	10.93.1.105	FTP	112	Response: 331 Guest login ok, send your email address as password.
396	70.452204	10.93.1.105	10.130.25.239	FTP	82	Request: PASS anonymous@example.com
397	70.495737	10.130.25.239	10.93.1.105	TCP	60	21 → 60606 [ACK] Seq=200 Ack=65 Win=14720 Len=0
398	70.495737	10.130.25.239	10.93.1.105	FTP	102	Response: 230 Guest login ok, access restrictions apply.
399	70.496183	10.93.1.105	10.130.25.239	FTP	68	Request: OPTS UTF8 ON
400	70.499912	10.130.25.239	10.93.1.105	TCP	60	21 → 60606 [ACK] Seq=248 Ack=79 Win=14720 Len=0
401	70.499912	10.130.25.239	10.93.1.105	FTP	77	Response: 200 OK, UTF-8 enabled
402	70.500546	10.93.1.105	10.130.25.239	FTP	61	Request: CWD /
403	70.508472	10.130.25.239	10.93.1.105	FTP	83	Response: 250 CWD command successful.
404	70.509117	10.93.1.105	10.130.25.239	FTP	59	Request: PWD
405	70.512307	10.130.25.239	10.93.1.105	FTP	85	Response: 257 "/" is current directory.
406	70.513151	10.93.1.105	10.130.25.239	FTP	62	Request: TYPE I
407	70.516465	10.130.25.239	10.93.1.105	FTP	74	Response: 200 Type set to I.
408	70.516941	10.93.1.105	10.130.25.239	FTP	60	Request: PASV
409	70.520679	10.130.25.239	10.93.1.105	FTP	104	Response: 227 Entering Passive Mode (10,130,25,239,217,56)
410	70.521772	10.93.1.105	10.130.25.239	FTP	71	Request: STOR 01test.txt
415	70.529659	10.130.25.239	10.93.1.105	FTP	113	Response: 150 Opening BINARY mode data connection for '01test.txt'.
416	70.529659	10.130.25.239	10.93.1.105	FTP	78	Response: 226 Transfer complete.
418	70.529824	10.93.1.105	10.130.25.239	TCP	54	60606 → 21 [ACK] Seq=122 Ack=484 Win=130560 Len=0

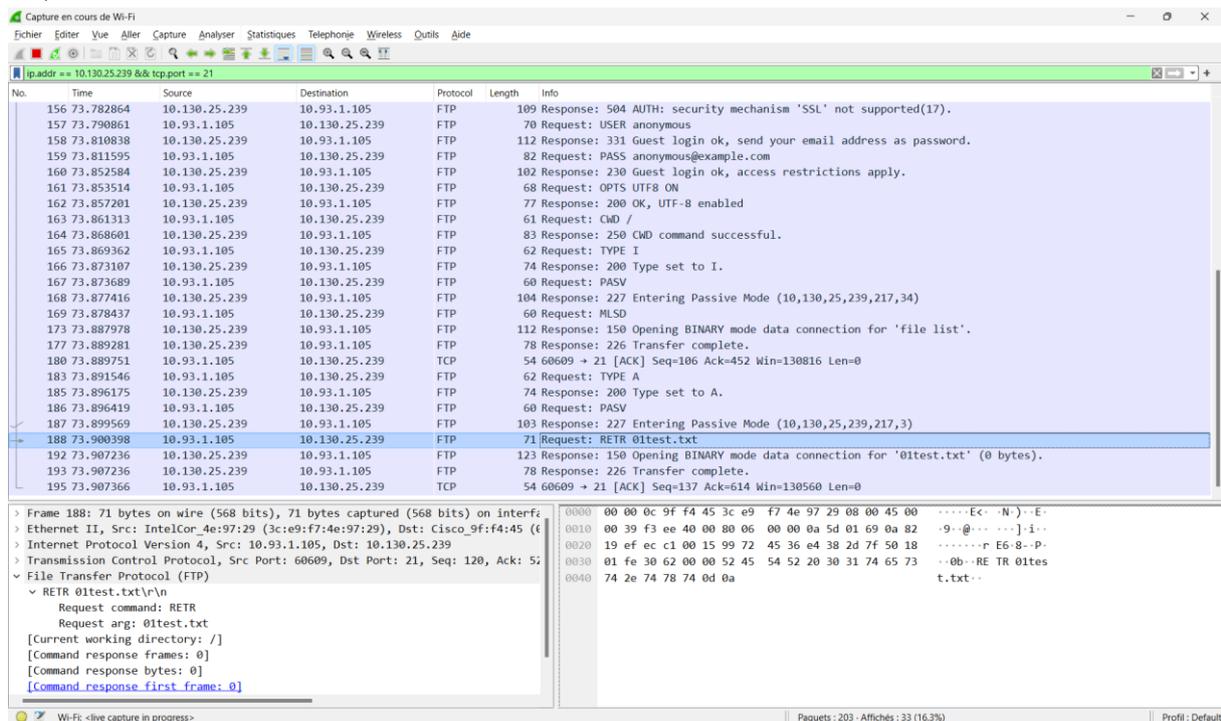
```
> Frame 410: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0
> Ethernet II, Src: IntelCor_4e:97:29 (3c:e9:f7:4e:97:29), Dst: Cisco_9f:f4:45 (08:00:0c:9f:f4:45)
> Internet Protocol Version 4, Src: 10.93.1.105, Dst: 10.130.25.239
> Transmission Control Protocol, Src Port: 60606, Dst Port: 21, Seq: 105, Ack: 484, Len: 71
File Transfer Protocol (FTP)
  STOR 01test.txt\r\n
    Request command: STOR
    Request arg: 01test.txt
    [Current working directory: /]
    [Command response frames: 0]
    [Command response bytes: 0]
    [Command response first frame: 0]
```

Download

It's pretty much the same for downloading files, we must right click on the server side on the file we have previously created. Then press download. This same file will be uploaded to the client in the current folder on the left side.

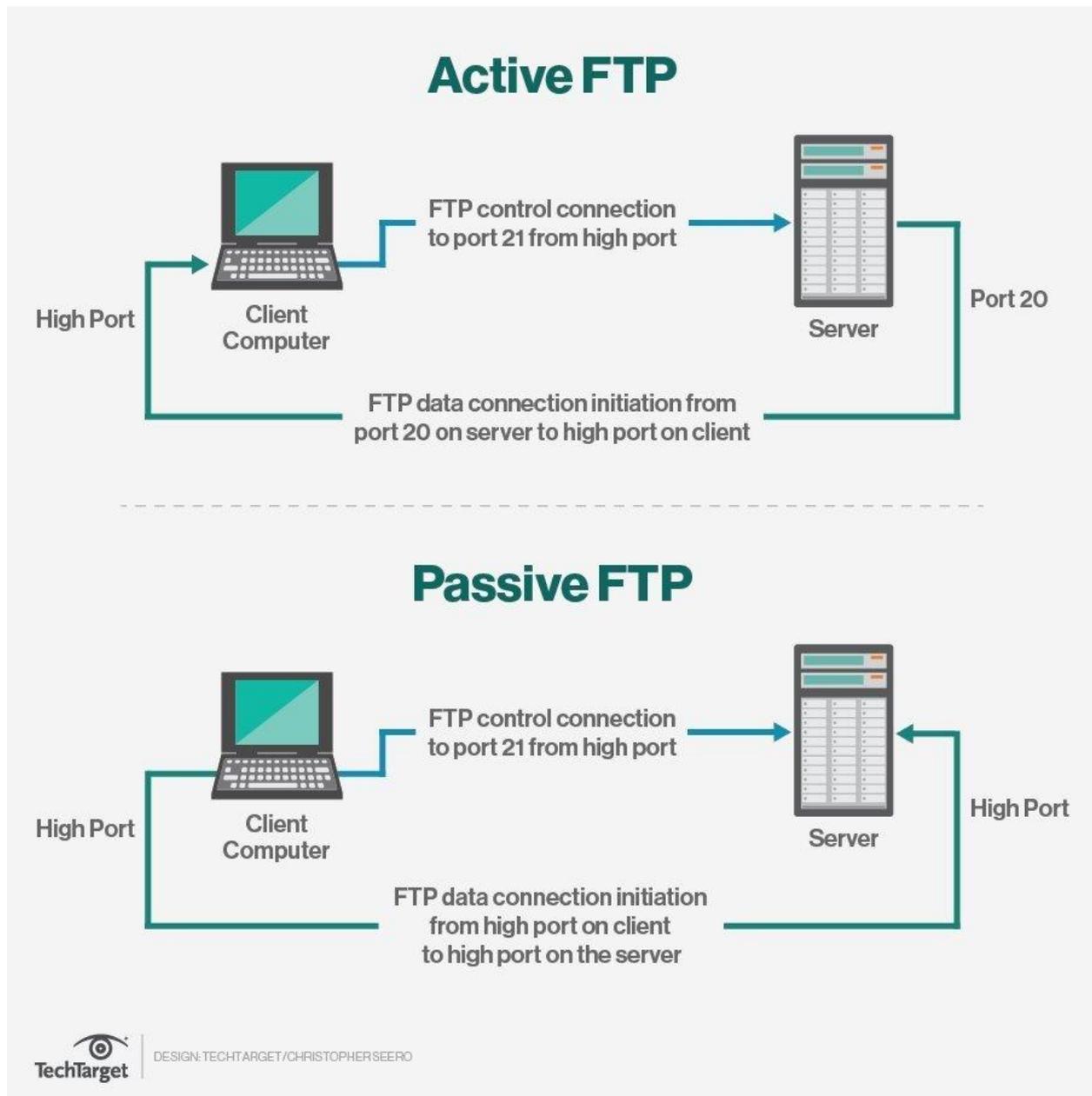


With Wireshark we see in blue the request line with the command "RETR" followed by the name of the file, in our case "01test.txt".



What are active/passive modes³

- For the active mode, this was originally the only mode available. The client will first establish a connection to the server with the "ACTV" command on the terminal and send its ports, the FTP server will create a data tunnel to the client and start the transfer.
- In passive mode, the client will have to issue the "PASSV" command and this time the server will send information such as the port to establish a data connection with the server. This method is useful to counteract the firewall which could block incoming connections to the client.



4

³ <https://www.techtarget.com/searchnetworking/definition/File-Transfer-Protocol-FTP>

⁴ https://cdn.ttgtmedia.com/rms/onlinelImages/FTP_active_passive.jpg

Data port

I use active mode on FileZilla, so the control port is 21 and the data transfer port is 20. But when in passive mode. When the client sends the command to be in passive then the server responds with this: "227 Entering Passive Mode: (192,168,150,90,195,149).

The representation of the parenthesis is (B1, B2, B3, B4, P1, P2), B means Byte so Byte number 1 of the IP address of the server then we have the P's, these are the ports. Here is the calculation so that the client knows which port to connect to and establish a data connection. $P1 * 256 + P2$, in this example it would be $195 * 256 + 149 = 50'069$. So, we have found our port.

```
testbox1: {/home/p-t/slacker/public_html} % ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (192,168,150,90,195,149) .
---> LIST
150 Opening ASCII mode data connection for file list
drwx-----  3 slacker  users          104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

5

⁵ https://www.cosmos.esa.int/documents/772136/977578/psa_activeVsPassiveFtp.pdf/5e36a7b8-8732-4e65-ab6b-6cf94a742ea6

1. During the Download, which server's port was used to transfer the file.

273	118.465122	10.93.1.105	10.130.25.239	FTP	60 Request: PASV
274	118.470112	10.130.25.239	10.93.1.105	FTP	104 Response: 227 Entering Passive Mode (10,130,25,239,217,10)
275	118.471295	10.93.1.105	10.130.25.239	FTP	71 Request: RETR 01test.txt
279	118.480090	10.130.25.239	10.93.1.105	FTP	123 Response: 150 Opening BINARY mode data connection for '01test.txt' (0 bytes).
280	118.480090	10.130.25.239	10.93.1.105	FTP	78 Response: 226 Transfer complete.
282	118.480378	10.93.1.105	10.130.25.239	TCP	54 60617 → 21 [ACK] Seq=117 Ack=463 Win=130816 Len=0
304	139.297125	10.130.25.239	10.93.1.105	FTP	110 Response: 421 Timeout (300 seconds): closing control connection.
305	139.297125	10.130.25.239	10.93.1.105	TCP	60 21 → 60606 [FIN, ACK] Seq=57 Ack=1 Win=115 Len=0
306	139.297125	10.130.25.239	10.93.1.105	TCP	60 [TCP Retransmission] 21 → 60606 [FIN, ACK] Seq=57 Ack=1 Win=115 Len=0
307	139.297381	10.93.1.105	10.130.25.239	TCP	54 60606 → 21 [ACK] Seq=1 Ack=58 Win=510 Len=0
308	139.297473	10.93.1.105	10.130.25.239	TCP	54 [TCP Dup ACK 307#1] 60606 → 21 [ACK] Seq=1 Ack=58 Win=510 Len=0
309	139.300778	10.93.1.105	10.130.25.239	TCP	54 60606 → 21 [FIN, ACK] Seq=1 Ack=58 Win=510 Len=0
310	139.304947	10.130.25.239	10.93.1.105	TCP	60 21 → 60606 [ACK] Seq=58 Ack=2 Win=115 Len=0


```

> Internet Protocol Version 4, Src: 10.93.1.105, Dst: 10.130.25.239
  Transmission Control Protocol, Src Port: 60617, Dst Port: 21, Seq: 100, Ack: 37
    Source Port: 60617
    Destination Port: 21
    [Stream index: 15]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 17]
    Sequence Number: 100 (relative sequence number)
    Sequence Number (raw): 1702614970
    [Next Sequence Number: 117 (relative sequence number)]
    Acknowledgment Number: 370 (relative ack number)
    Acknowledgment number (raw): 2258684366
  
```

```

0000 00 00 0c 9f f4 45 3c e9 f7 4e 97 29 08 00 45 00  ....E< .N) .E<
0010 00 39 f4 00 40 00 80 06 00 00 0a 5d 01 69 0a 82  .G. @ . . . .] i<
0020 19 ef ec c9 00 15 65 7b d7 ba 86 a0 c9 ce 50 18  . . . .oI L? .p .bP<
0030 01 ff 30 62 00 00 52 45 54 52 20 30 31 74 65 73  . . . .oP . . . . .P<
0040 74 2e 74 78 74 0d 0a                               . . . .RE TR 01tes
                                                    t.txt<
  
```

Destination Port (tcp.dstport), 2 byte(s) | Paquets : 319 - Affichés : 11.9% | Profil : Def

2. During the Upload, which server's port was used to transfer the file.

26	1.267600	10.130.25.239	10.93.1.105	FTP	105 Response: 227 Entering Passive Mode (10,130,25,239,217,109)
27	1.268346	10.93.1.105	10.130.25.239	FTP	85 Request: STOR Wireshark - Shortcut.lnk
34	1.278525	10.130.25.239	10.93.1.105	FTP	127 Response: 150 Opening BINARY mode data connection for 'Wireshark - Shortcut.lnk'.
36	1.278525	10.130.25.239	10.93.1.105	FTP	78 Response: 226 Transfer complete.
38	1.278856	10.93.1.105	10.130.25.239	TCP	54 60620 → 21 [ACK] Seq=131 Ack=468 Win=130816 Len=0


```

> Ethernet II, Src: IntelCor_4e:97:29 (3c:e9:f7:4e:97:29), Dst: Cisco_9f:f4:45 (6
  Internet Protocol Version 4, Src: 10.93.1.105, Dst: 10.130.25.239
  Transmission Control Protocol, Src Port: 60620, Dst Port: 21, Seq: 100, Ack: 37
    Source Port: 60620
    Destination Port: 21
    [Stream index: 1]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 31]
    Sequence Number: 100 (relative sequence number)
    Sequence Number (raw): 1867074623
    [Next Sequence Number: 131 (relative sequence number)]
    Acknowledgment Number: 371 (relative ack number)
  
```

```

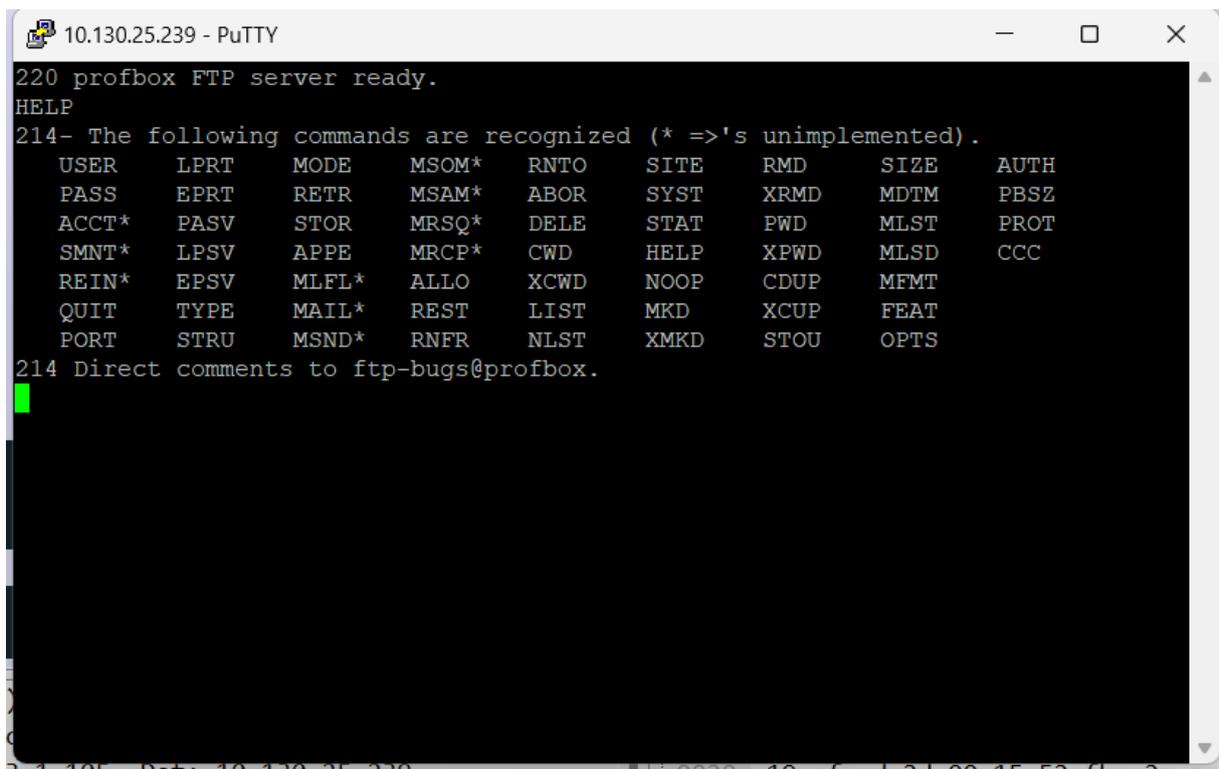
0000 00 00 0c 9f f4 45 3c e9 f7 4e 97 29 08 00 45 00  ....E< .N) .E<
0010 00 47 f4 15 40 00 80 06 00 00 0a 5d 01 69 0a 82  .G. @ . . . .] i<
0020 19 ef ec cc 00 15 6f 49 4c 3f 99 70 19 62 50 18  . . . .oI L? .p .bP<
0030 01 ff 30 70 00 00 53 54 4f 52 20 57 69 72 65 73  . . . .oP . . . . .P<
0040 68 61 72 6b 20 2d 20 53 68 6f 72 74 63 75 74 2e  hark - S horkcut<
0050 6c 6e 6b 0d 0a                               lnk<
  
```

Destination Port (tcp.dstport), 2 byte(s) | Paquets : 98 - Affichés : 27.6% | Profil : Def

Create a folder

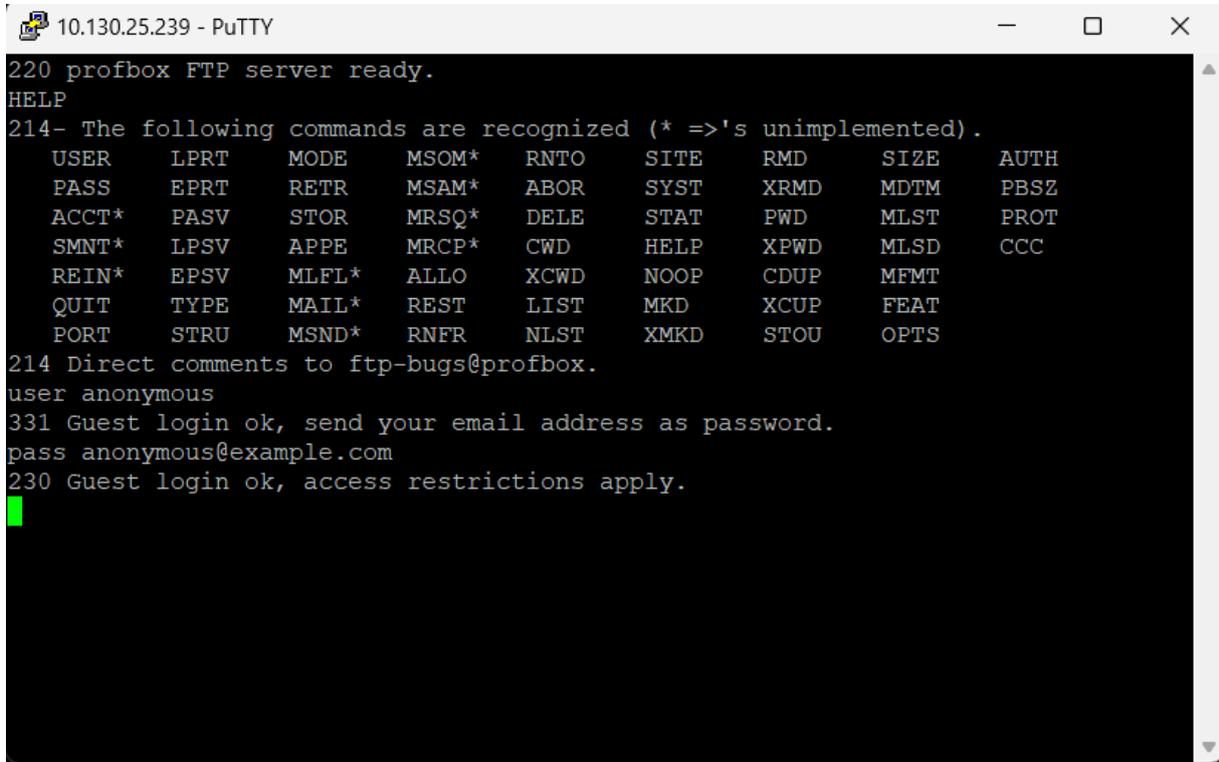
We will create a file on the remote server in the command line. First open PuTTY, and to establish the connection we need to enter the IP address of the server and port 21, then choose the RAW method and finally press connect.

This is what is displayed afterwards. In order to have the rights to create a folder, you will have to do a few things on the command line, first you must authenticate yourself. We will consult the list of commands implemented by the FTP server and enter "help".



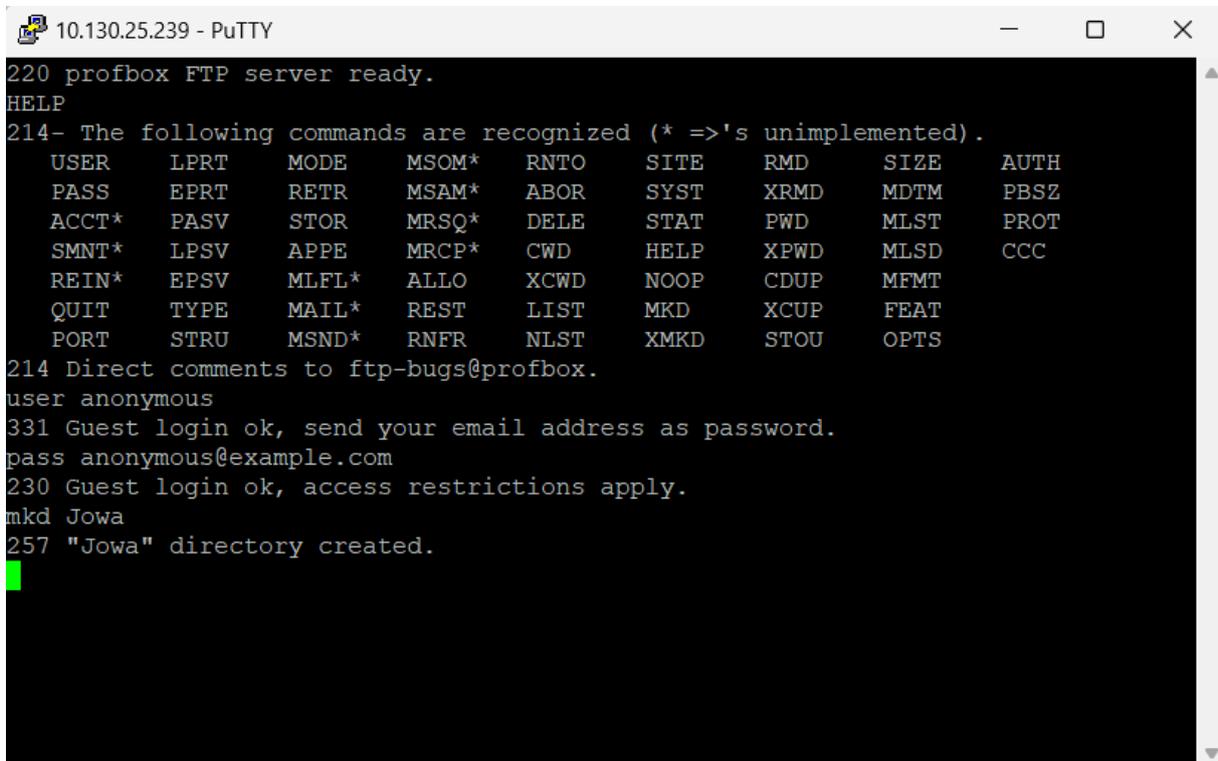
```
10.130.25.239 - PuTTY
220 profbox FTP server ready.
HELP
214- The following commands are recognized (* =>'s unimplemented).
USER      LPRT      MODE      MSOM*     RNT0      SITE      RMD       SIZE      AUTH
PASS      EPRT      RETR      MSAM*     ABOR      SYST      XRMD      MDTM      PBSZ
ACCT*     PASV      STOR      MRSQ*     DELE      STAT      PWD       MLST      PROT
SMNT*     LPSV      APPE      MRCP*     CWD       HELP      XPWD      MLSD      CCC
REIN*     EPSV      MLFL*     ALLO      XCWD      NOOP      CDUP      MFMT
QUIT      TYPE      MAIL*     REST      LIST      MKD       XCUP      FEAT
PORT      STRU      MSND*     RNFR      NLST      XMKD      STOU      OPTS
214 Direct comments to ftp-bugs@profbox.
```

Here is the one we are interested in: So, enter the command "USER anonymous" and then "PASS anonymous@example.com". The user value "anonymous" is mandatory and the password is the one you want.

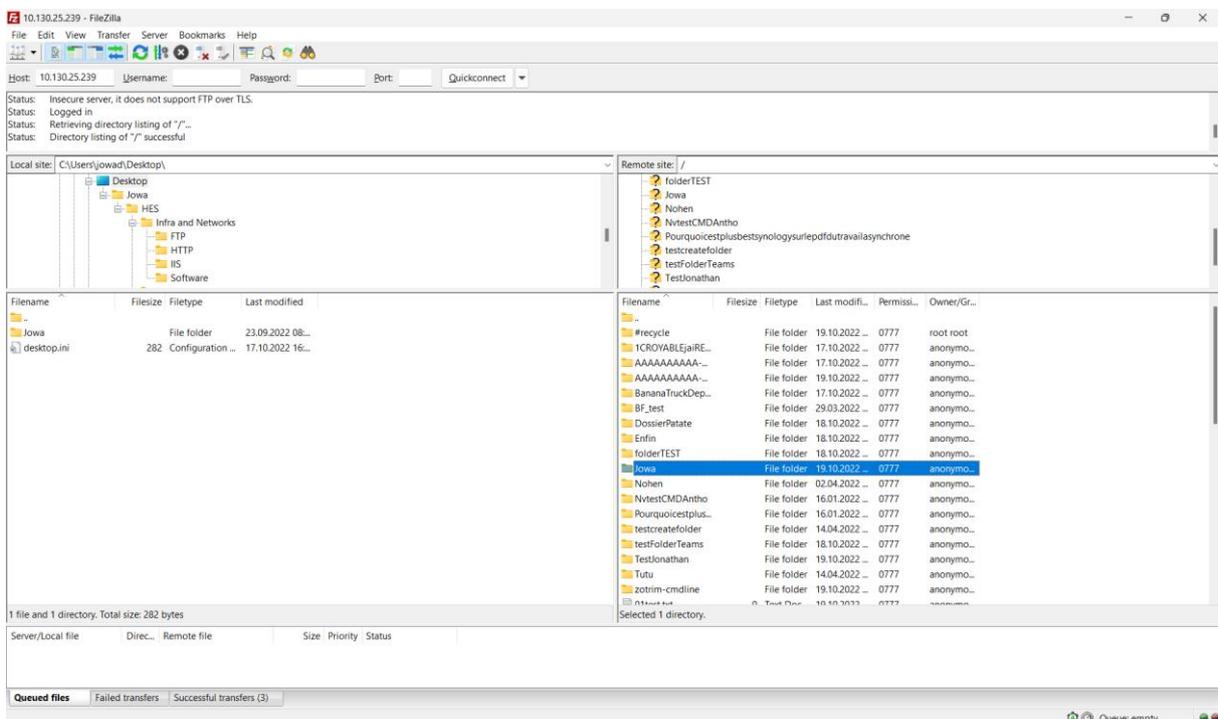


```
10.130.25.239 - PuTTY
220 profbox FTP server ready.
HELP
214- The following commands are recognized (* =>'s unimplemented).
  USER    LPRT    MODE    MSOM*   RNT0    SITE    RMD     SIZE    AUTH
  PASS    EPRT    RETR    MSAM*   ABOR    SYST    XRMD    MDTM    PBSZ
  ACCT*   PASV    STOR    MRSQ*   DELE    STAT    PWD     MLST    PROT
  SMNT*   LPSV    APPE    MRCP*   CWD     HELP    XPWD    MLSD    CCC
  REIN*   EPSV    MLFL*   ALLO    XCWD    NOOP    CDUP    MFMT
  QUIT    TYPE    MAIL*   REST    LIST    MKD     XCUP    FEAT
  PORT    STRU    MSND*   RNFR    NLST    XMKD    STOU    OPTS
214 Direct comments to ftp-bugs@profbox.
user anonymous
331 Guest login ok, send your email address as password.
pass anonymous@example.com
230 Guest login ok, access restrictions apply.
█
```

We are authenticated now we want to create a new folder, the command is "make directory" so `mkd` followed by the folder name "`mkd myFolder`". The server replies that the folder has been created successfully! This way we can check if everything went well on FileZilla and if so, the file is present.



```
10.130.25.239 - PuTTY
220 profbox FTP server ready.
HELP
214- The following commands are recognized (* =>'s unimplemented).
USER      LPRT      MODE      MSOM*     RNTO      SITE      RMD       SIZE      AUTH
PASS      EPRT      RETR      MSAM*     ABOR      SYST      XRMD      MDTM      PBSZ
ACCT*     PASV     STOR      MRSQ*     DELE      STAT      PWD       MLST      PROT
SMNT*     LPSV     APPE      MRCP*     CWD       HELP      XPWD      MLSD      CCC
REIN*     EPSV     MLFL*     ALLO      XCWD      NOOP      CDUP      MFMF
QUIT      TYPE     MAIL*     REST      LIST      MKD       XCUP      FEAT
PORT      STRU     MSND*     RNFR      NLST      XMKD      STOU     OPTS
214 Direct comments to ftp-bugs@profbox.
user anonymous
331 Guest login ok, send your email address as password.
pass anonymous@example.com
230 Guest login ok, access restrictions apply.
mkd Jowa
257 "Jowa" directory created.
```



Download a file

Downloading is more complicated. We have to go through the same steps as before but with a few more. The connection, then the authentication commands are done as before but we are not going to create a folder this time but to download so for that we are going to put ourselves in passive mode here is the command to send to the server: "pasv". The server answers us, and we have to calculate the port: $P1 * 256 + P2 = 217 * 256 + 21 = 55573$.

The PORT verb

A PORT request asks the server to use a different mechanism of creating a data connection: the server makes a TCP connection to the client.

The PORT request has a parameter in the form

$h1, h2, h3, h4, p1, p2$

meaning that the client is listening for connections on TCP port $p1*256+p2$ at IP address $h1, h2, h3, h4$. (The RFC 959 formal syntax does not allow any of these numbers to be 0. The formal syntax is wrong.)

The server normally accepts PORT with code 200. If the server was listening for a connection, it stops, and drops any connections already made.

The server does not connect to the client's port immediately. After the client sends RETR and after the server sends its initial mark, the server attempts to connect. It rejects the RETR request with code 425 if the connection attempt fails; otherwise it proceeds normally.

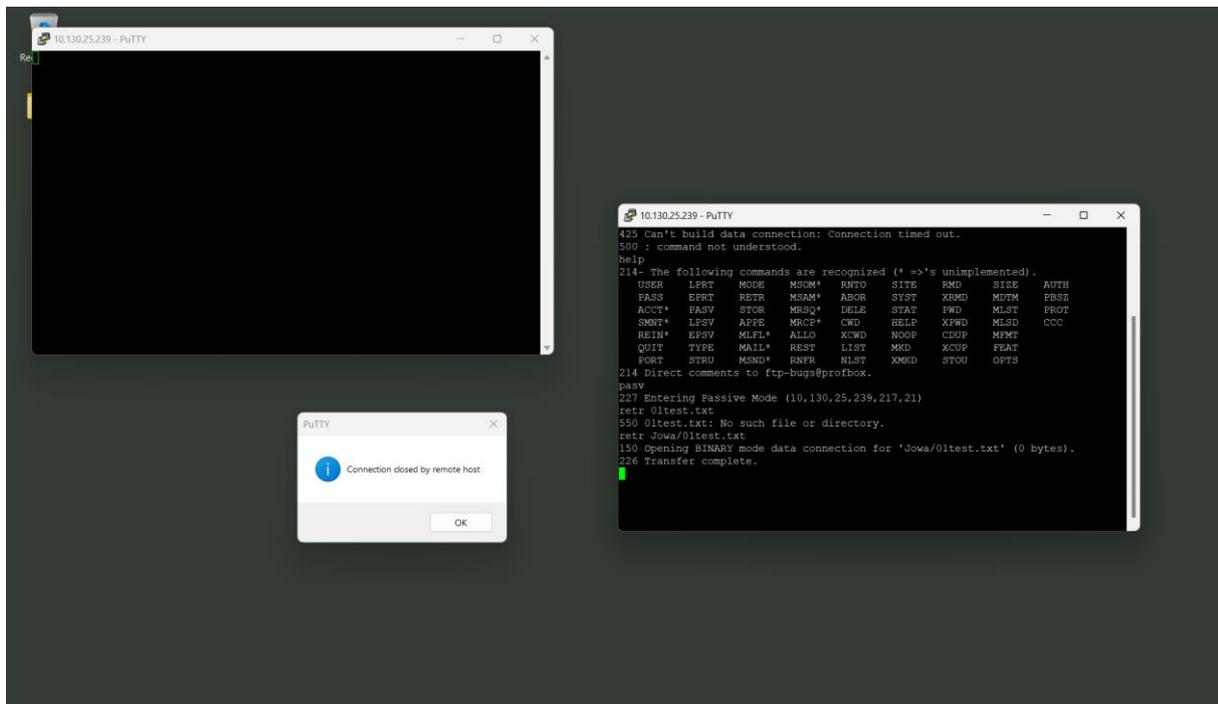
In theory, the client can send RETR without a preceding PORT or PASV. The server is then supposed to connect to port 20 at the client's IP address. In practice, however, servers refuse to do this.

For [security reasons](#), clients should never use PORT. However, some clients still rely on PORT, and will give up on a file transfer if PORT is rejected. My current recommendation is that servers continue to support PORT.

6

We must use the "retrieve" command followed by the name of the file to download so that the server understands and listens, waiting for a connection from the client, therefore "retr 01test.txt" and press the "Enter" key. At this time, we must create a second tunnel to receive the file and for this we will create a second PuTTY session.

So, we launch a second PuTTY process, and we enter the IP address and the port calculated previously, 55573. With the RAW connection method and finally press connect. A new window opens and initializes the connection. If the window disappears, it means that the passive mode is not activated or that the port calculation is incorrect or that you went too slow between the "retr 01test.txt" command step of the first session and the opening of the second session.



⁶ <https://cr.yip.to/ftp/retr.html>

Upload a file

The process for uploading is the same as the download step and moreover we can directly write in the second window the content of the file to be uploaded. The command to send our file is "stor 02test.txt".

When the second window is open you can write whatever you want, and the data is transferred when this same window is intentionally closed.

You can then for all exercises check your terminal actions on the FileZilla interface. So, you can see that the file has been sent!

