File Transfer Protocol

File Transfer Protocol

- FTP (File Transfer Protocol) is the simplest and most secure way to exchange files over the Internet.
- Transferring files from a client computer to a server computer is called **"uploading"** and transferring from a server to a client is **"downloading".**
- To access an FTP server, users must be able to connect to the Internet or an intranet (via a modem or local area network) with an FTP client program.

File Transfer Protocol

- FTP doesn’t not really move, it copies files from one computer to another
- FTP is the file transfer protocol in the Internet's TCP/IP protocol suite’s Application Layer.

File Transfer Protocol

- An FTP Client is software that is designed to move files back-and-forth between two computers over the Internet.
- It needs to be installed on your computer and can only be used with a live connection to the Internet.

FTP Clients

- Some commonly used FTP clients include the following:
- **FileZilla**- a popular FTP client that is freely available for Windows, Macintosh, and Linux users
  Available as a free download from the Internet.
- **Fire FTP**- a plug-in for the popular Firefox web browser that can be used just like a standalone FTP program
  Installed through the FireFox browser.
- **Dreamweaver**- page layout/design program, which include FTP access as one of its many features
  Available for purchase from Adobe

FTP Clients

- There are many FTP client programs, some of which are run from a command-line (such as the command ***ftp***, a standard installed in many operating systems), but a large majority allow the user to manipulate files via a graphical interface (such as CuteFTP), which makes file transfers more user-friendly.

History

- The completion of FTP dates from 1971 when a file transfer system (described in RFC141) between MIT machines (*Massachusetts Institute of Technology*) was developed.
- Many RFC have since made improvements/changes to the basic protocol, but the greatest innovation date from July 1973.
- The FTP protocol is currently defined by RFC 959 (*File Transfer Protocol (FTP) - Specifications*).

Features

- FTP operates in a client/server environment, meaning that the remote machine is configured as a server, and consequently waits for the other machine(client) to request a service from it.
- In UNIX, the service is provided by what is called a daemon, a small task that runs in the background. The FTP daemon is called *ftpd*.


My Site
images
index
FTP Download
Internet
Yahoo!
Web Hosting
My Site
images
index
Internet
FTP Upload
Your Computer

Features

- **The FTP protocol is used for transferring one file at a time,** in either direction, between the client machine (the one which initiated the connection, i.e. the calling machine) and the server machine (which provided the FTP service, i.e. the called machine).
- The FTP protocol can also perform other actions, such as creating and deleting directories (only if they are empty), listing files, deleting and renaming files, etc.

Features

- FTP allows files to have ownership and access restrictions
- FTP hides the details of individual computer systems

Example

- downloading MP3.
- Online games rely heavily on FTP.
- Other events like auctions and online trading use FTP to transact business
- **FTP** is commonly used to transfer Web page files from their creator to the computer that acts as their server for everyone on the Internet.

Differences between FTP and HTTP

- The major difference between FTP and HTTP is that FTP is a two-way system - FTP can be used to copy or move files from a server to a client computer as well as upload or transfer files from a client to a server.
- HTTP, on the other hand, is strictly one-way: "transferring" text, pictures and other data(Multimedia files) from the "server" to a client computer which uses a web browser to view the data.

Differences between FTP and HTTP

- FTP systems generally encode and transmit their data in binary sets which allow for faster data transfer; HTTP systems encode their data in MIME format which is larger and more complex.
- Files are automatically copied or moved from a file server to a client computer's hard drive, and vice versa. On the other hand, files in an HTTP transfer are viewed and can 'disappear' when the browser is turned off unless the user executes commands to move the data to the computer's memory.

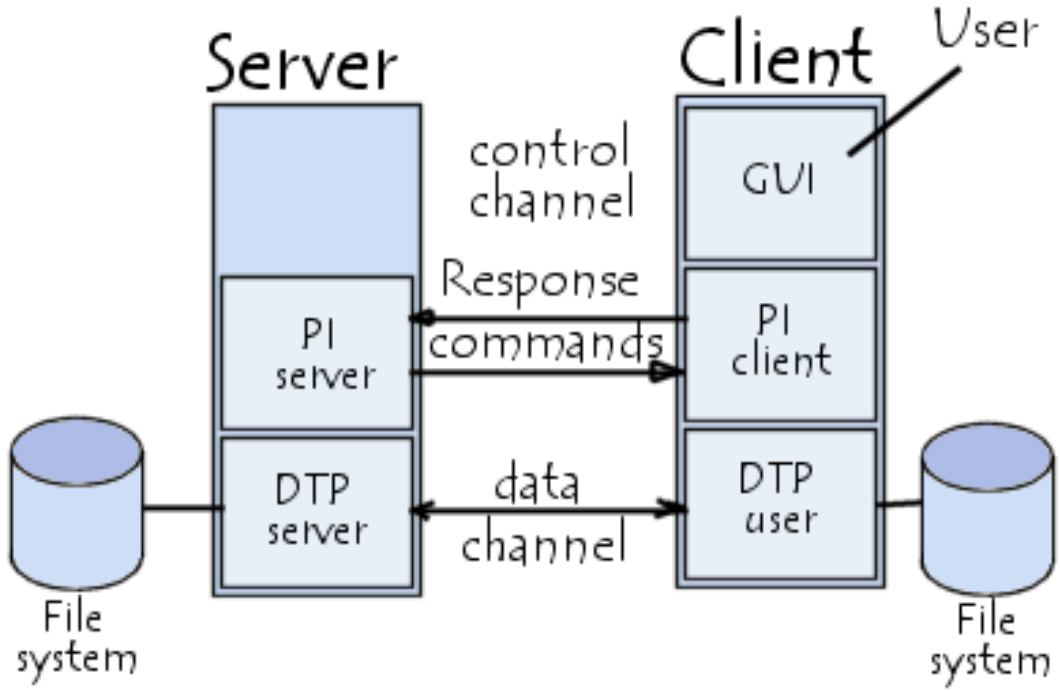
The FTP model

FTP protocol falls within a client-server model, i.e. one machine sends orders (the client) and the other awaits requests to carry out actions (the server).

During an FTP connection, two transmission channels are open:

- A channel for commands (control channel) , a control channel that stays open for the entire session
- A channel for data , data channel that opens and closes to transfer data such as folder listings and files to or from the server as requested by the client.

The FTP model



Control Channel occurs on port 21

The FTP model

So, both the client and server have two processes allowing these two types of information to be managed:

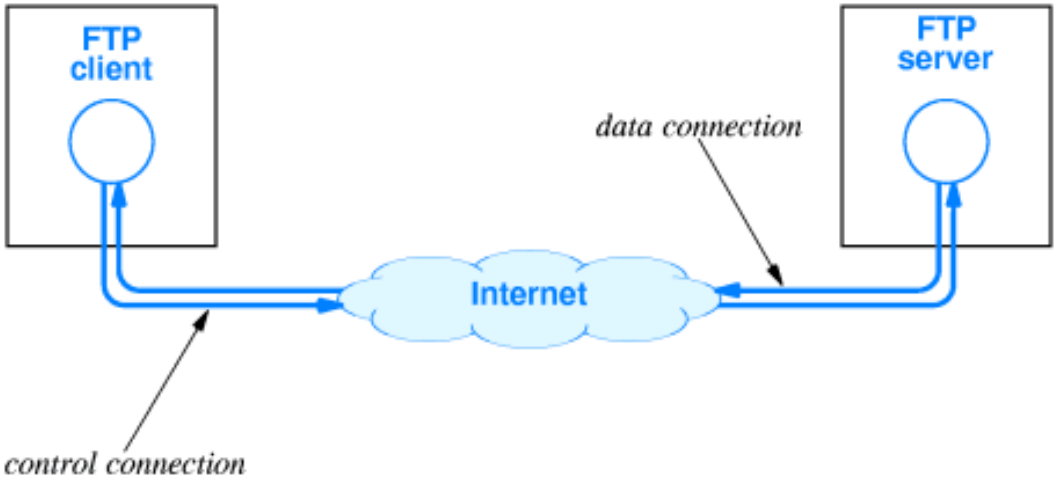
- **DTP** (*Data Transfer Process*) It establishes the connection and managing the data channel. The server side DTP is called *SERVER-DTP*, the client side DTP is called *USER-DTP*
- **PI** (*Protocol Interpreter*) Controls DTP using commands received over the control channel. It is different on the client and the server:

The FTP model

- The **SERVER-PI** is responsible for listening to the commands coming from a USER-PI over the control, establishing the connection for the control channel, receiving FTP commands from the USER-PI over this, responding to them and running the SERVER-DTP.
- The **USER-PI** is responsible for establishing the connection with the FTP server, sending FTP commands, receiving responses from the SERVER-PI and controlling the USER-DTP if needed.

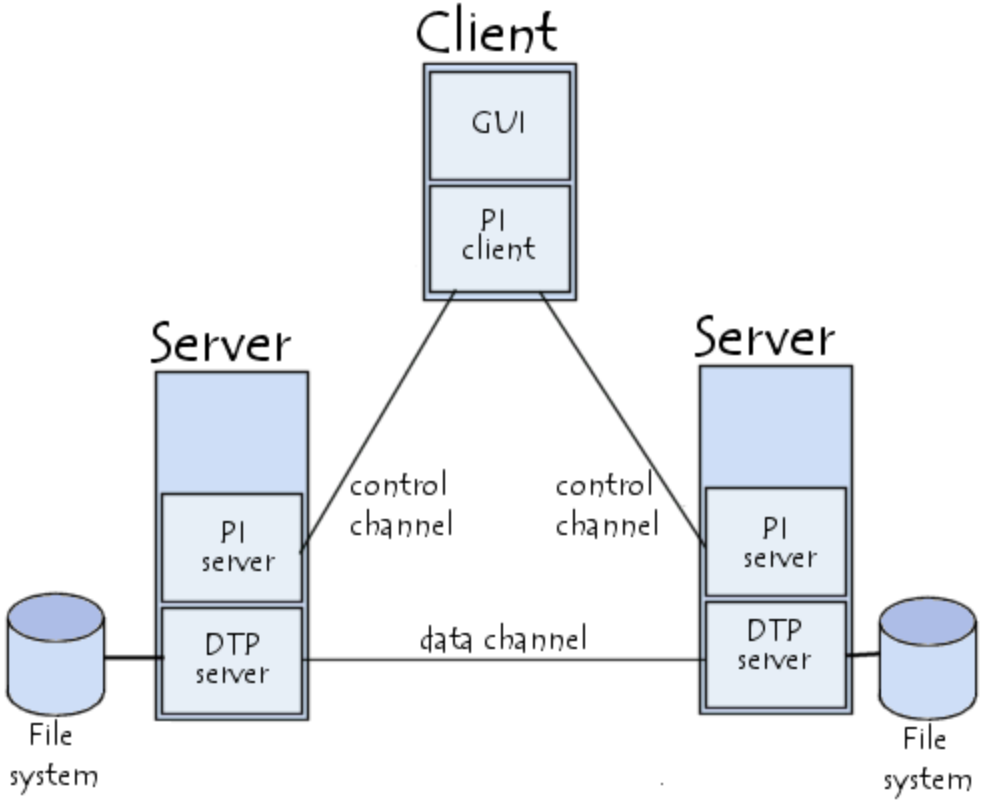
The FTP model

- When an FTP client is connected to a FTP server, the USER-PI initiates the connection to the server according to the Telnet protocol.
- The client sends FTP commands to the server, the server interprets them, runs its DTP, then sends a standard response.
- Once the connection is established, the server-PI gives the port on which data will be sent to the Client DTP.
- The client DTP then listens on the specified port for data coming from the server.


FTP
client
FTP
server
data connection
Internet
control connection

The FTP model

- It is important to note that since the control and data ports are separate channels, it is possible to send commands from one machine and receive data on another. So, for example it is possible to transfer data between FTP servers by passing through a client to send control instructions and by transferring information between two server processes connected on the right port.


Client
GUI
PI client
Server
PI server
DTP server
control channel
control channel
data channel
Server
PI server
DTP server
File system
File system

Types of connections

- When an FTP client connects to an FTP server it opens a connection to the FTP control port 21. Then the *client* tells the FTP *server* whether to establish an ***active* or *passive*** connection.
- The type of connection chosen by the client determines how the server responds and on what ports transactions will occur.

Types of connections

- **Active Connections**:

  When an active connection is established, the *server* opens a data connection to the client from port 20 to a high range port on the client machine.

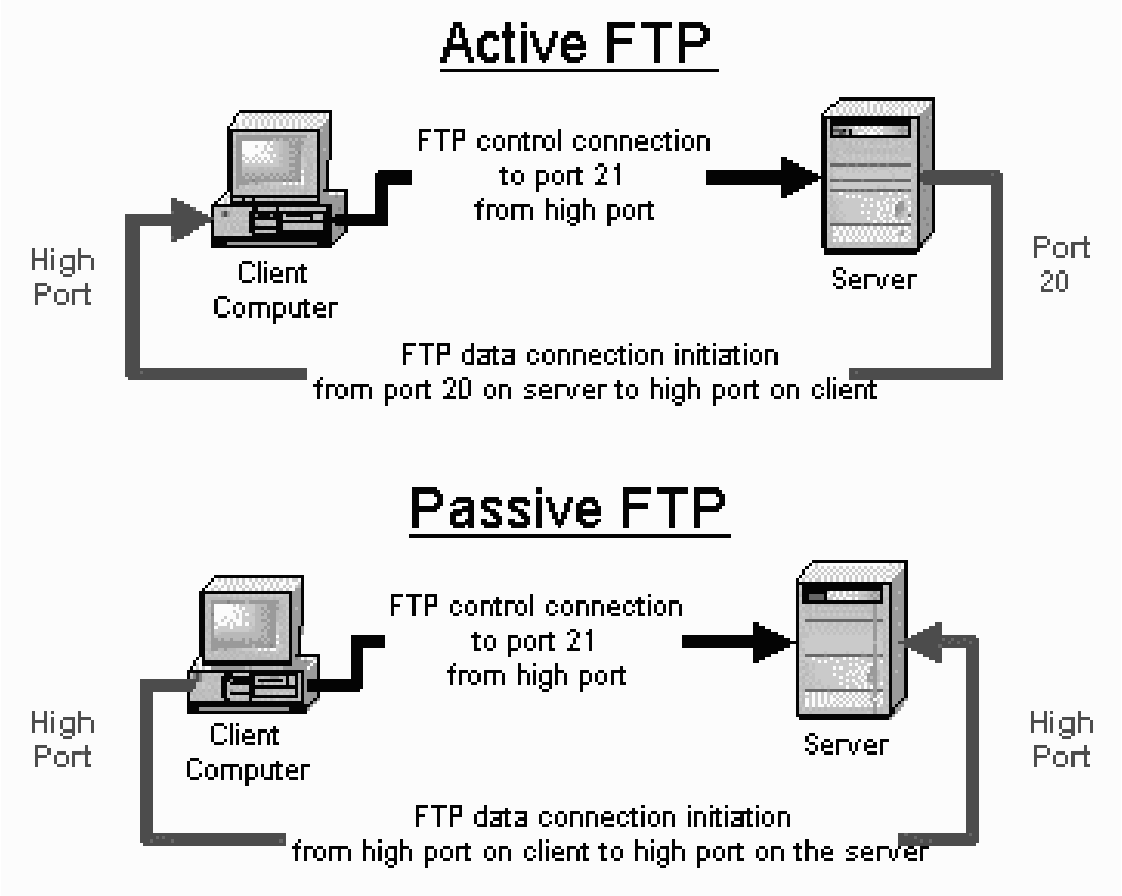
  All data from the server is then passed over this connection.

Types of connections

Passive Connections :When a passive connection is established, the *client* asks the FTP server to establish a passive connection port, which can be on any port higher than 10,000.

- The server then binds to this high-numbered port for this particular session and relays that port number back to the client. The client then opens the newly bound port for the data connection.
- Each data request the client makes results in a separate data connection. Most modern FTP clients attempt to establish a passive connection when requesting data from servers.

Types of connections



The FTP commands

There are three different types of FTP commands:

- Access control commands
- Transfer parameter commands
- FTP service commands

Access Control Commands

- USER: *Character string allowing the user to be identified*. User identification is necessary to establish communication over the data channel.
- PASS :*Character string specifying the user's password*. This command must immediately precede the *USER* command. It falls to the client to hide the display of this command for security reasons.
- ACCT: *Character string representing the user's account*. The command is generally not necessary. During the response accepting the password, if the response is 230 this stage is not necessary, if the response is 332, it is.

Access Control Commands

- CWD :*Change Working Directory*: This command enables the current directory to be changed. This command requires the directory's access path to be fulfilled as an argument.
- CDUP *Change to Parent Directory*: This command allows you to go back to the parent directory. It was introduced to solve problems of naming the parent directory according to the system (generally "..").

Access Control Commands

- SMNT: *Structure Mount*
- REIN: *Reinitialize*
- QUIT: Command enabling the current session to be terminated. The server waits to finish the transfer in progress if the need arises, then supplies a response before closing the connection.

Transfer parameter commands

- PORT: Character string allowing the port number used to be specified.
- PASV: Command making it possible to indicate to the DTP server to stand by for a connection on a specific port chosen randomly from among the available ports. The response to this command is the IP address of the machine and port.
- TYPE:This command enables the type of format in which the data will be sent to be specified.

Transfer parameter commands

- STRU :Telnet character specifying the file structure (F for *File*, R for *Record*, P for *Page*).
- MODE :Telnet character specifying data transfer method (S for *Stream*, B for *Block*, C for *Compressed*)

FTP service commands

- RETR: This command (*RETRIEVE*) asks the server DTP for a copy of the file whose access path is given in the parameters.
- STOR: This command (*store*) asks the server DTP to accept the data sent over the data channel and store them in a file bearing the name given in the parameters. If the file does not exist, the server creates it, if not it overwrites it.
- STOU: This command is identical to the previous one, only it asks the sever to create a file where the name is unique. The name of the file is returned in the response.

FTP service commands

- APPE :(*append*) the data sent is concatenated into the file bearing the name given in the parameter if it already exists, if not, it is created.
- ALLO: This command (*allocate*) asks the server to plan a storage space big enough to hold the file whose name is given in the argument.
- REST: This command (*restart*) enables a transfer to be restarted from where it stopped. To do so, the command sends the marker representing the position in the file where the transfer had been interrupted in the parameter. This command must immediately follow a transfer command

FTP service commands

- RNFR: This command (*rename from*) enables a file to be renamed. In the parameters it indicates the name of the file to be renamed and must be immediately followed by the *RNTO* command.
- RNTO: This command (*rename to*) enables a file to be renamed. In the parameters it indicates the name of the file to be renamed and must be immediately followed by the *RNFR* command.

FTP service commands

- ABOR :This command (*abort*) tells the server DTP to abandon all transfers associated with the previous command. If no data connection is open, the DTP sever does nothing, if not it closes it. The control channel however remains open.
- DELE :This command (*delete*) allows a file to be deleted, the name of which is given in the parameters. This command is irreversible, confirmation can only be given at client level.

FTP service commands

- RMD: This command (*remove directory*) enables a directory to be deleted. The name of the directory to be deleted is indicated in the parameters.
- MKD :This command (*make directory*) causes a directory to be created. The name of the directory to be created is indicated in the parameters.

FTP service commands

- PWD: This command (*print working directory*) makes it possible to resend the complete current directory path.
- LIST :This command allows the list of files and directories present in the current directory to be resent. This is sent over the passive DTP. It is possible to place a directory name in the parameter of this command, the server DTP will send the list of files in the directory placed in the parameter.
- NLST: This command (*name list*) enables the list of files and directories present in the current directory to be sent.

FTP service commands

- SITE :This command (*site parameters*) causes the server to offer specific services not defined in the FTP protocol.
- SYST :This command (*system*) allows information on the remote server to be sent.
- STAT :This command (*status*) makes it possible to transmit the status of the server, for example to know the progress of a current transfer. This command accepts an access path in the argument, it then returns the same information as LIST but over the control channel.

FTP service commands

- HELP: This command gives all the commands understood by the server. The information is returned on the control channel.
- NOOP: This command (*no operations*) is only used to obtain an OK command from the server. It can only be used in order not to be disconnected after an excessive period of inactivity.

The FTP responses

- The FTP responses make it possible to ensure synchronization between the client and FTP server. So, at each command sent by the client, the server will potentially carry out an action and systematically send back a response.
- The responses are made up of a 3 digit code indicating the way in which the command sent by the client has been processed. However, since this 3 digit code is hard to read for humans, it is accompanied by a text (Telnet character string separated from the numeric code by a space).

The FTP responses

The response codes are made up of 3 numbers the meanings of which are as follows:

- The first number indicates the status of the response (success or fail)
- The second number indicates what the response refers to.
- The third number gives a more specific meaning (relative to each second digit)

First number Digit Meaning

- **1**yz means Preliminary positive response :The action requested is in progress, a second response must be obtained before sending a second command
- **2**yz Positive fulfilment response The action requested has been fulfilled, a new command can be sent
- **3**yz Intermediary positive response The action request is temporarily suspended. Additional information is awaited from the client

First number Digit Meaning

- **4**yz Negative fulfilment response The action requested has not taken place because the command has temporarily not been accepted. The client is requested to try again later
- **5**yz Permanent negative response The action requested has not taken place because the command has not been accepted. The client is requested to formulate a different request

Second number Digit Meaning

- x**0**z Syntax The action has a syntax error, or is a command not understood by the server
- x**1**z Information This is a response sending back information (for example a response to a STAT command)
- x**2**z Connections The response relates to the data channel
- x**3**z Authentication and accounts The response relates to the (USER/PASS) login or the request to change the account (CPT)

Second number Digit Meaning

- x**4**z Not used by the FTP protocol
- x**5**z File system The response relates to the remote file system

Logging in to an FTP server

- The command "**ftp**" is available across various platforms, including UNIX, Windows and Linux. The command initiates an FTP session, and is usually run as follows:

  ftp server_name

Logging in to an FTP server

- *server_name* represents the name or IP address of the remote machine that the user wants to connect to. The target machine must, of course, have an FTP service.
- Once the connection has been initialised, a few lines of text appear on the screen. The first line lets you know that you have connected to an FTP server, the next lines welcome you to it, and may indicate which kind of FTP site it is (i.e, what sort of files it hosts or which organisation owns it), or instructions for users.

Logging in to an FTP server

- In FTP, each line begins with a number that represents either success or failure. For a welcome message, the line might be preceded by the number 220, which means "the service is ready for the new user."
- The server asks you to enter your user name (also called a *login* or *identification*), in order to set access rights (such as read/write privileges). After the user name has been accepted, a line beginning with the number 331 invites you to input your password, which is masked, meaning that it doesn't appear on the screen

Logging in to an FTP server

- In some cases the server may be public, in which case you can log in anonymously, and you will therefore have to log in as "anonymous" (or "guest"). For public servers, custom dictates that the user enters his/her email address as the password, but you can enter whatever you choose.
- Once the password has been accepted, a message will show if connection has been established or not, in which case a reason will be given (for example, the site may have reached its maximum number of users allowed at a time, in which case the message "*No more users allowed*" appears).

Logging in to an FTP server

- Once logged in, the FTP site waits for the user to enter commands describing actions to perform.

Some more commands

- help Displays all commands supported by the FTP server.
- status Used for showing some of the client machine's settings
- binary This command switches you from ASCII mode (sending text documents) to binary mode (sending binary files, i.e. non-text files like images or programs)
- ascii Switches from binary mode to ASCII mode. This is the default mode.

Some more commands

- type Displays the current transfer mode (binary or ASCII)
- user Allows you to log in on the current FTP server again using a different user name. You will then be requested to enter a new password
- ls Lists all files found in the current directory. The command "ls -l" gives additional information on the files.
- pwd Displays the full name of the current directory

Some more commands

- CD The command means *change directory*, and is used for changing to a different directory. The command "cd .." is used to access the parent directory
- mkdir The command *mkdir* (in UNIX, or *md* in Microsoft) is used for creating a directory within the current directory. The use of this command is reserved for users with access allowing it.
- rmdir The command *rmdir* (in UNIX, or r*md* in Microsoft) is used for deleting a directory within the current directory. The use of this command is reserved for users with access allowing it.

Some more commands

- **get** This command is used to retrieve a file found on the server. If the command is followed by a file name, the remote file will be transferred to the local machine, into the current local directory
- If the command is followed by two file names, the remote file (the first name) is transferred to the local machine in the current local directory, with the specified file name (the second name)
- If the file name contains spaces, be sure to enter it within quote marks.

Some more commands

put This command is used to send a local file to the server If the command is followed by a file name, the local file will be transferred to the remote machine, into the current remote directory

- If the command is followed by two file names, the local file (the first name) is transferred to the remote machine in the current remote directory, with the specified file name (the second name) If the file name contains spaces, be sure to enter it within quote marks.

Some more commands

- open Logs out and opens a new session on another FTP server
- close Logs out, leaving the FTP client active
- bye Disconnects the FTP client from the server and puts it into inactive mode
- quit Disconnects the FTP client from the server and puts it into inactive mode

Problems with FTP

- FTP does not interpret the contents of a file transferred in binary mode -- this can cause problems, e.g. a file of 32-bit floating point numbers where the representation is different on the two computers
- it doesn't use strong authentication.
- It is based on password logins which can be guessed, or discovered by cybercriminals using a sniffer.

Problems with FTP

- FTP sends files in clear plain-text
- The main reason that web sites get hacked is because they are being updated with insecure FTP transfers.

Secure FTP

- SFTP, or secure FTP, is a program that uses SSH to transfer files. Unlike standard FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted in the clear over the network.
- It is functionally similar to FTP, but because it uses a different protocol, you can't use a standard FTP client to talk to an SFTP server, nor can you connect to an FTP server with a client that supports only SFTP.
- You can use SFTP with a graphical SFTP client or at the command line.

Secure FTP

- Most secure FTP products use encryption and X.509 certificates.
- There are numerous encryption algorithms used in secure FTP products including: DES, 3DES, CAST-128, Blowfish, AES-128, and others.

Bibliography

- Douglas Comer, *Computer Networks and Internets with Internet Applications*
- RFC 959, *File Transfer Protocol (FTP)*
- Kermit: A File Transfer Protocol by Frank Da Cruz
- Big Book of Internet File Transfer Rfcs (Big Book (Morgan Kaufmann)) by Peter Loshin

Bibliography

- Using Ftp (User Friendly Reference) by Mary Ann Pike, Noel Estabrook
- E-Mail and Ftp (File Transfer Protocol) by L. Joyce Arnston, Kathy Berkemeyer, Ken Halliwell, Thomas Neuburger
- Kermit: A File Transfer Protocol by Frank Da Cruz