**Part A:**

**1.** Because logs can tell us who, when and how, want to compromise our system (server).

**2.** Yes there is a problem with concurrent logging because each client thread

can make request for write something to log file at the same time.

To solve this problem we should make method that allow us only one thread work with and other thread should wait until first thread done.

**Part C:**

**3.** Port mode is less secure. This command can produce "The Bounce Attack"

The attack involves sending an FTP "PORT" command to an FTP server containing the network address and the port number of the machine and service being attacked. At this point, the original client can instruct the FTP server to send a file to the service being attacked. Such a file would contain commands relevant to the service being attacked (SMTP, NNTP, etc.).

Instructing a third party to connect to the service, rather than connecting directly, makes tracking down the perpetrator difficult and can circumvent network-address-based access restrictions.

As an example, a client uploads a file containing SMTP commands to an FTP server. Then, using an appropriate PORT command, the client instructs the server to open a connection to a third machine's SMTP port. Finally, the client instructs the server to transfer the uploaded file containing SMTP commands to the third machine. This may allow the client to forge mail on the third machine without making a direct connection. This makes it difficult to track attackers.

Protecting Against the Bounce Attack

The original FTP specification [PR85] assumes that data connections will be made using the Transmission Control Protocol (TCP) [Pos81]. TCP port numbers in the range 0 - 1023 are reserved for well known services such as mail, network news and FTP control connections [RP94]. The FTP specification makes no restrictions on the TCP port number used for the data connection. Therefore, using proxy clients have the ability to tell the server to attack a well known service on any machine.

To avoid such bounce attacks, it is suggested that servers not open data connections to TCP ports less than 1024. If a server receives a PORT command containing a TCP port number less than 1024, the suggested response is 504 (defined as "Command not implemented for that parameter" by [PR85]). Note that this still leaves non-well known servers (those running on ports greater than 1023) vulnerable to bounce attacks. Several proposals provide a mechanism that would allow data connections to be made using a transport protocol other than TCP. Similar precautions should be taken to protect well known services when using these protocols. Also note that the bounce attack generally requires that a perpetrator be able to upload a file to an FTP server and later download it to the service being attacked. Using proper file protections will prevent this behavior. However, attackers can also attack services by sending random data from a remote FTP server which may cause problems for some services. Disabling the PORT command is also an option for protecting against the bounce attack. Most file transfers can be made using only the PASV command. The disadvantage of disabling the PORT command is that one loses the ability to use proxy FTP, but proxy FTP may not be necessary in a particular environment.

**Part D:**

**4.** FTP over TLS (Explicit)

Explicit security requires that the FTP client issues a specific command to the FTP server after establishing a connection to establish the SSL link. In explicit TLS the FTP client needs to send an explicit command (i.e. "AUTH TLS") to the FTP server to initiate a secure control connection. The default FTP server port is used. This formal method is documented in RFC 4217.

FTP over TLS (Implicit) - Deprecated

Implicit security is a mechanism by which security is automatically turned on as soon as the FTP client makes a connection to an FTP server. In this case, the FTP server defines a specific port for the client (990) to be used for secure connections. FTP over TLS Implicit has been deprecated and should no longer be used.

**5.**

Insecure because commands transferred as plain text and can be sniffed by attacker.

For PORT command we should check valid port range (at least port number should > 1024)

and remote host for data connection should be equal to control connection host.

**Compare FTP and SFTP**

FTP and SFTP are two different file transfer protocols and the major difference between the two is the security associated with the file transfer. FTP was the first file protocol and is less secure, while SFTP stands for secure file transfer protocol, and as the name suggests is more secure than FTP.

TCP/IP networks such as the internet use FTP or File Transfer Protocol to transfer files from one computer to another. FTP works on a server and client based architecture, meaning that the client can access any information on the server at any given time. Some servers are password protected so that one can only access the information after entering an ID and password.

FTP uses two separate channels for data and control both of which are unencrypted, this means that the information from either of the channel can be intercepted and accessed. Using unencrypted channel is a huge security loop hole and led to design of more secure mode of communication such as FTPS and SFTP.

SFTP is more secure way of communication and is based on SSH (secure shell). SSH is a secure way of providing access to all the shell accounts on remote server. The information communicated using SFTP is first divided into small packets and unlike FTP, SFTP uses only one channel for data and control. Before sharing the information between two computers SFTP verifies the identity of client and once a secured connection is established it sends the encrypted information (the encryption cipher is predefined).

**Compare FTP vs P2P**

FTP is what is called a peer-to-server connection. You connect directly to a server via the File Transfer Protocol (FTP) to download a file. Your download speed is limited by both the speed of your connection and the speed on the server's connection.

Bit Torrent is what is called peer-to-peer. You connect to a bunch of other computers that are also downloading or have downloaded the file. There is no one server that is hosting the file. You download the parts of the file you don't have from other users who already received that part, and while you are downloading, you are uploading the parts already downloaded to other computers. This spreads the load out so that a single server isn't needed, and if there are enough other downloader's, your download speed is only limited by your internet connection.

**6.**

Log PASS command attempts number this can help us to make anti-brute-force method.

As well we must log PORT command arguments. These logs can help us to make method vs "Bounce Attack".

To prevent this type of attack we at least must check port range (port should be > 1024)

AND IMPORTANT thing that we must compare control connection host and port arguments remote host, they should be equal otherwise bounce attack detected.