Name: Dinh Nguyen

CS472 Homework 4

**Part A – Add a server configuration file and restrict PORT/PASV**

**Question 1**

The difference between PASV and PORT commands is the direction of how the server and the client is connected. For PASV command, the server will remain listening for the clients to connect to the chosen port. Meanwhile, for PORT command, the client requests the server to be connected and the server actually has to act to initiate and connect to the client. Thus, this implies a middle-man threat if the hackers know the IP addresses of the server and the client, he can connect to the server before the client when the client start the PASV or he can connect to the client before the server when using the PORT command. With that, he can get information and data on the server and client and then mimic a successful file/data transfer with an empty file. Both commands PASV and PORT create threats to the server and the client but PORT command attack is harder to be tracked down and protected against. The attacker may make his own FTP connection to the same server and have the server make the connection to the targeted port number. It is impossible for the legitimate client to distinguish the resulting connection from a legitimate server connection.

Network Address Translating or NAT will act as a firewall in between the client and server that will block any connection outgoing from the server (thus, blocking PORT command). NAT firewalls will only allow one way connection from client to the server thus only works with PASV mode. This will provide more security to our FTP. It is not a good thing for an application to know about IP addresses because attackers can easily use it to manipulate or potentially hack onto the system. NAT will translate our networks into a signal IP address, hence, it will hide our actual IP and protect our systems from outside attacker.

**Question 2**

Depends on the type of service we want to provide, the server can use a fixed pathname or a relative pathname. Using a fixed pathname will be better in term of managing the system as we will not want to have different version of ftp config files on the server. The variety of ftp config files will create conflicts, some unexpected behaviors and also potentials for attackers to break into the system. Some config may actually allow other users or outsiders to get into the system. Thus, it is better to have the config file in a fixed directory as we can have more control on the server and we can modify the permission on the directory so that no one can access the file or change the file as well. However, if the server goal is to provide customized services, then using relative pathname would be the best option as it will allow users to customized their options for their service.

**Part B – Logging**

**Question 3**

Logging is an important part of security as it plays a vital role in debugging and making sure that everything is configured correctly. Logging usually will reflect all configurations and how things are working on the system and thus, help admins to identify malicious vulnerability or attacks on our systems.

**Question 4**

There will be a problems with concurrent servers and log file as servers have to allow multiple clients to connect at the same time. Thus, if we do not manage it well, some clients may try to write into the same log file at the same time and that will cause confusion or misunderstanding while reading the server log and it will be harder to debug or detect an attack as well. To solve the problem, for each thread created for each client, they will be assigned a thread ID and the ID will be included in the information that specific client is writing to the log so we can know where/which connection/which client it comes from.

**Part C – Securing the connection with SSL/TLS**

**Question 5**

The issue with IMPLICIT mode is it only allow encrypted connection via port 990. If the client does not support encryption or TLS, the server will reject the connection. Meanwhile, because EXPLICIT mode allows for flexibility for the clients which allow them to use unencrypted data connection, the issue would be potential security breach. The tradeoff of IMPLICIT mode is more security for the systems but it requires the clients to support encryption for both control and data connection. The tradeoff of EXPLICIT mode is the trade of strictness/security for flexibility and performance as it allows clients to use regular FTP when the client does not support encryption. Thus, depends on the system's policy, the company policy, the type of service we want to provide and the data type that we want to transfer (is it a shared data, confidential data and so on), we will have to choose between IMPLICT and EXPLICIT mode for it.

**Part D – Analyzing the conversation**

**FTP, SFTP and BitTorrent**

FTP is file transfer protocol that uses port 21 for control and another port for data transfer. SFTP is file transfer protocol using SSH encryption which offers service on port 22 for both data and control commands. Thus, FTP is considered to have better organizing structure as the data is operated on a separate port. With that, FTP would have better performance when transferring data faster and may be more reliable. However, FTP is doing raw and unencrypted service which will propose a huge security threats for both client and server. Thus, SFTP will be better in term of security. SFTP uses SSH to encrypt the data transfer but using the same port 22 for data and control operations. This will cause SFTP to do extra work to manage the info and communication between the server and the client and may cause extra delay/ longer operation which affect the file transfer performance. BitTorrent, on the other hands, allows the high level of flexibility as everyone can be both client and server. The security risks of BitTorrent is very high and that is the tradeoff for performance as files and data can be transferred easily and quickly among the network.

**Part E – Analyzing the operation of the server**

**Question 6**

There are a couple ways to attack an FTP server such as DDoS attack (Distributed Denial of Service) where the attackers try to disrupt the normal traffic of the target server by overwhelming the server with excessive traffic. Another way to hack onto FTP server is when the logs is leaked outside and attacker can read the data in the logs which may contain users accounts details and then use the username and password info to get on the server. I am using a large number for the port when implementing the server

so I didn’t have any logging that looks like someone is trying to get onto the server. However, if someone happens to use the same port, I would expect to see some strange logs where it shows that someone is entering a wrong password or a wrong username. He/she will have to enter the correct pair of user/password to get on the system. To solve that, we can add a log for the client IP to identify where the client comes from and we can encrypt the password using BCRYPT python library and uses the hash of password for the server service and log file instead of raw user/password and then check the two hashes before allowing the client to connect. We can look at the server log to see the hash for the password and how it is implemented.