Homework 4

CS472

Michael Giorgianni

**PART A**

1. In PORT or active mode one of the major security flaws is the fact that the server initializes the connection which is generally uncommon. At first it didn’t register to me then I realized, if some server (which is just a beefy computer, more or less) can just form a connection to your pc to send data, then what is stopping other computers from opening a connection under the guise of a FTP server and make your computer part of a bot net without you even knowing something connected to your pc. With Passive mode circumvents this issue because the client initialized the connection, and the server responds with the information to the data port so then it would be up to the client to know if the Server is trustworthy or not. Assuming the server is trustworthy this is the safer option. With Active mode the server must check that the client is trustworthy, is a valid IP, an accepted IP (not previously banned or something). Using NAT would be interesting because the server would have to open the data connection to the public IP of the NAT and then the router or whatever that was handling NAT would have to relay the conversation to the intended participant in the internal network. If the application had access to the computers actual IP address it could leave the user open to attacks. Generally, VIA the server becoming compromised and the hacker uses the server’s knowledge of user IP address and then attacks the users. Having a kind of shroud over the internal network using NAT on the users end would generally protect that internal network from various cyber-attacks.

**PART A**

2. Having a fixed pathname for deploying as a system service for example is nice for updating things that go out on a network where if you need to make a change to the file you can find it, or if you want to verify config file is correct you can find it quickly. Having the fixed pathname can be more secure because the server will not be affected if somehow it was starting up in the wrong directory. If you moved the server to a new directory and the config file was in the same place the server could still find the config file. Whereas if you have a relative path location moving the server to a new directory would make it lose track of the config file because it would normally be found relative to the server directory which is now in a new location. Having it relative like to the current directory, it will keep the configuration stored with the other files which can make zipping up a personal project easier and allow for more flexible code and directory changes in the future. The downside to a relative path is hackers could somehow add a file of their own that was not authorized and have the server pick up on it, say the server is checking in current directory for a text file and it sees the hacker’s version first, so it uses that one. Implementing a check or maybe have the first line in the text file be some kind of key and if nobody knows about the key when the hacker injects their file the server would detect the lack of a key and do something like shut down the server or kick the user off. Often vulnerabilities to path traversal can be dangerous and allow hackers to get to things like the user file containing usernames and passwords which would be devastating. If you have a fixed pathname for traversal, then clients connecting could only go to specific locations that are pre coded into the server. Things like the user files would be sectioned off somewhere and you could even include a check that makes sure any users other than a system admin could look at user files.

**PART B**

3. Logging is an important part of security similar to the camera at the entrance of a Walmart. If someone stole something, if you can narrow the time down with logs then you can see who logged into the system and is a potential suspect of the theft. Even more the logging of file transfers and other manipulations would let the server admin know who exactly did what and allow for more straightforward solutions to such attacks. In FTP if someone were able to access another user’s files and transfer them without logs the victim wouldn’t know that their files were copied and if they files were completely stolen (transferred out) then they victim is now without potentially important files. What if someone tried to get onto the system over and over, using the wrong password, well with logging you would be able tell that someone is trying to break in, you could have a system set up to punish repeated attempts so brute force attacks would be better defended against.

**PART B**

4. I have not logged anyone trying to break into my server, if they were to try, I could see when people are joining the server that are not supposed to. If I were to add logs to test for security flaws, I would have the username printed with the commands so that you can see if that user should be using those commands with those files. For example, if Jim has file x only Jim can modify file x, if you see Eric Transferred file x, then you know something is wrong. If I were to log someone repeated attempting to brute force onto the server or maybe just forming connections at rates that are not humanly possible, maybe attempting to use up server resources, then I would be able to properly implement checks to punish people for doing such things. Multiple failed log-on attempts? “Too many failed attempts try again in x time” or “access denied unauthorized user contact system admin if this was a mistake”. Possible DDoS attack? Deny connections coming from that IP address. I know it is not necessarily simple to avoid DDoS attacks but its just a potential attack I could think of.

**PART C**

5. For implicit assuming other applications won't be using port 990 you wouldn’t need to worry about interference from unsecure communications, which could cause problems. You would have the benefit of always secure and encrypted data communication because as soon as a client connects, they are switched to TLS. With explicit you would need to have more flexibility in the communication at the default port for FTP. Explicit communication passes through port 21 to tell the server to switch to TLS. With either explicit and implicit the TLS/SSL protocol is identical however with implicit you would need a dedicated port at 990 listening for TLS connections and for explicit it would be an optional request that the client would have to send to the server. For either the USER and PASS commands are sent encrypted and everything after, so all communication is secure. If the client chooses not to toggle encryption on via explicit TLS then they would face the normal issues that come with non-encrypted communication online, which if important files are involved could be disastrous. Along side this, you would also have to be able to handle connections encrypted or not which would require some more work. If the user implicitly connects via TLS/SSL then their information is always as secure as SSL can make it which is generally good and what you want.

**Extra Credit**

FTP is great and lays a good protocol out to make communication "simple" and efficient. SFTP is great because it allows for secure and private connections for file transfer. It is generally used in cases where confidentiality is so important it could be criminal, for example, health care information is supposed to be confidential. If a hacker could get that information because a hospital was using normal FTP then they could be in serious trouble with the law, therefore, using SFTP is the preferred method. Using bitTorrent can be good because it’s a peer to peer connection where each client downloading is also uploading the data they are downloading so that if a lot of people are downloading the same torrent it will download faster because every client is sharing the burden instead of a central server handling all the work. The downside to FTP is the fact its unsecure, especially today we should strive to be as private online as possible. With SFTP the downsides could be attributed to the management of SSH keys. As for bitTorrent the dangers could come with the uncertainty of who is on the other side. The fact that there isn’t one central location for the file to be downloaded from makes it hard to verify where it is you are getting that file from. What if the source is not trustworthy and now hundreds possibly thousands of people are sharing this file downloading and uploading without realizing what they are “hosting” it could lead to some serious problems.