CS472 Computer Networks
Fall 2016 - 2017
Homework Assignment # 4
Due Date: Monday, November 14th, 2016 at 5:59pm (CLASS TIME)

NOTE: Assignments must be submitted in electronic format via Drexel Learn (http://learning.drexel.edu). All the work must be original, NO TEAM WORK. Late assignments will not be accepted. Please submit your assignment as your first initial and last name as a zip file with all files needed (e.g. mine would be mkain_hw4.zip).

Objective

Homeworks #2 and #3 asked you to implement a popular network protocol. The goal of this assignment is to adjust that implementation with several security features to help it be more secure.

Part A – Add a server configuration file and implement log rolling

This part of the assignment is to modify your server to implement reading in a configuration file when you initialize the server. The configuration file should be in a relative location (it should get it out of the current directory at a fixed name (for example ftpserverd.conf). This configuration file should be of a few ("attribute" = "value" pairs). It should ignore any lines which begin with a pound sign ("#"). The first configuration attribute that should be added is "logdirectory" which should have some value and a default if it is not specified in the configuration file. For example,

```
# This defines where the logfile will reside
# defaults to /var/spool/log if not set (you may replace the default)
logdirectory=/home/mkain/logfiles
```

Also, you need to implement log rolling. You may specify the name of the log file (which for this example is logfile). When the new logfile is created, the old logfile becomes logfile.000 and a new logfile is created. If the attribute numlogfiles is specified in the configuration file, you must maintain that number of older logfiles. For example, if numlogfiles=5, then you have to keep the current logfile and then logfile.000 through logfile.004.
You do not have to worry the cases if you configure numlogfiles lower than a previous version and to clean up the current directory or if you change logdirectory.

```
# number of logfiles to keep (defaults to 5)
numlogfiles=5
```

You have to handle all of the error cases about directories that don't exist, you don't have the right privileges, etc. If the logfile cannot be created, it should be a fatal error to the server (the server should print an error message and exit).

Question to be answered:

1. Why is logging an important part of security?
2. Do you see any problems with concurrent servers and log files? (dealing with multiple processes or threads writing to the log file at the same time)? Brainstorm how to solve this problem.

Part B – Username file

This part of the assignment is to modify your server to implement an external username file.

```
# name of username file
usernamefile = /home/mkain/ftp.users
```

You have to handle all of the error cases about files that don't exist, you don't have the right privileges, etc. If the username file cannot be found, it should be a fatal error to the server (the server should print an error message, log it as well, and exit).

Part C – Restrict PORT/PASV via configuration file

This part of the assignment is to modify your server to restrict PORT (and EPRT) and PASV (and EPSV) by the configuration file.

```
# port_mode supported (default = no)
port_mode = NO
# pasv_mode supported (default = yes)
pasv_mode = YES
```

It is possible to set both of these attributes to YES. If both are set to NO, then it is a fatal error and the server should print an error message and exit. You should test via a client using both modes and ensure that the correct error messages are returned.

Question to be answered:

3. What are the security considerations with port_mode? With pasv_mode? Why would I want one or the other (think about some of the problems that you had with the client and the server – and who calls who)? Think of the conversation between client and server.

Part D – Securing the connection with SSL/TLS

Questions to be answered:

4. What are the different issues with securing the connections with IMPLICIT mode (which is a separate server listening with TLS always at port 990 by default) and EXPLICIT mode (which is the same server waiting on port 21 for special commands to turn TLS on and off)? What are the "it depends" part of the tradeoffs?

EXTRA CREDIT (worth up to 25 points) – implement one of the two modes (look at openssl() and other objects for sockets to implement).

Part E – Analyzing the conversation

Question to be answered:

5. Why is the 3 person method of FTP (as originally defined in the RFC) really insecure? Think of what you could do to cause trouble with the approach and what you can do in your clients and servers to stop that from happening. Do you have to do any checking in your program(s) with PORT/PASV to make sure that it isn't happening (that YOU ARE talking to the same host)? Think about the data channel as well as the control channel.

EXTRA CREDIT (worth up to 10 points): Think of the conversation of FTP and compare it to other file transfer protocols

- SFTP – offers the service on port 22 and data and commands share the same channel – better or worse?
- BitTorrent – offers files from a large number of hosts.

What are the good points and bad points of each approach (FTP, SFTP, BitTorrent)?

Part F – Analyzing the operation of the server

Question to be answered:

6. Do you think there are events that you've logged which show that someone is trying to break into your server? Do you have to add other log entries and checking to make sure that attacks aren't happening? Brainstorm some attacks against your FTP server. For at least one of them, implement additional code to check for them and log appropriate entries.

Your submission

Your submission (All client and server code) MUST contain the following:

- Well documented code (VERY WELL documented code) that should compile correctly. Please resubmit ***ALL*** code for the server, not just what was modified.
- A README file detailing instructions to compile your code or the use of your makefile and how to run your code. Please include the name of the compiler and the operating system (and/or system) that you tested it on. Please also include the list of commands that can be used from your UI.
- Server log from a sample run using both your client and the released FTP client.
- The answers to the questions above (and JUSTIFY YOUR ANSWERS!!!!)
- Any other information you deem important (like your name, etc.)

Point Sheet

Item	**Points**
Part A - implementation	15
Part A – question #1	10
Part A – question #2	10
Part B - implementation	10
Part C - implementation	15
Part C – question #3	10
Part D – question #4	10
Part E – question #5	10
Part E – question #6	10
Total	**100**
Part D – extra credit	Up to 25 pts
Part E – extra credit	Up to 10 pts