CS 483/5583 Software Security

Assignment #2

This assignment is based on the SimpleWebServer program discussed in class. If you choose to use a different programming language, it is your responsibility to translate the Java code correctly. Turn in your answers and source code as **a single zip** file.

1. What if a client connects to SimpleWebServer, but never sends any data or disconnects? What type of attack would such a client be able to conduct? (10 points)

The attacker could conduct a Denial of Service or DoS attack. This is because whenever a connection request is received, the server creates a new socket and waits for user input from the client. Because of this, if the client never enters any data or disconnects, the socket on the server is left waiting forever for input and occupies that particular socket. If conducted at scale or with multiple requests, this could take up all sockets in the server leading to DoS for other clients who need to access the server.

To solve this, a timeout for the request could be implemented in the run method to timeout and disconnect sockets if they take a long amount of time.

1. Modify SimpleWebServer.java to allow the client to upload files through a PUT command (e.g., PUT <destination\_path> <file\_content>) and save all client requests into a log file. Also modify SimpleWebClient.java to allow the user to upload a given file to the destination path on the server. The sample methods for text file storage and logging are given below. You may update storeFile to deal with binary files if needed. (30 points)

public void storeFile(BufferedReader br, OutputStreamWriter osw, String pathname) throws Exception {

FileWriter fw = null;

Scanner sc = new Scanner(br);

try {

fw = new FileWriter(pathname);

String s = sc.nextLine();

while(!s.isEmpty() && s != null) {

fw.write(s+"\n");

s = sc.nextLine();

}

fw.close();

sc.close();

osw.write("HTP/1.0 201 Created");

} catch(Exception e) {

osw.write("HTTP/1.0 500 Internal Server Error");

}

}

public void logEntry(String filename, String record){

FileWriter fw = new FileWriter(filename, true);

fw.write((new Date()).toString()+" "+record);

fw.close();

}

1. Rewrite the serveFile method such that it imposes a maximum file size limit. If a user attempts to download a file larger than the maximum allowed size, write a log entry to a file called error\_log.txt and return a “403 Forbidden” HTTP response code. (20 points).
2. Describe and implement: (a) an attack that defaces (overwrites) the index.html homepage and (b) an attack that removes the log data. Attach screenshots of your attacks. (20 points)
3. Suppose you are the attacker who has got hold of the complied SimpleWebServer.class. Describe an attack such that, after SimpleWebServer is re-started (e.g., because of an exception by another attack), the functionality in (3) is disabled. (20 points)

Notes: attach a copy of your source code and screenshots of the program execution.