CS558 Assignment 3

Due: 11:59pm June 26 (Sunday)

This assignment is done individually.

This assignment consists of two parts and students need to complete BOTH parts.

Part 1 (20% of the total score):

1. **[30 points]** Consider the following simple hash function H, which computes the hash code H(M) of a message consisting of n blocks b1, b2, …, bn.

   $$hi = bi1 \oplus bi2 \oplus \ldots \oplus bin$$

   - hi: the ith bit of the hash code
   - bij: the ith bit of the jth block

   Assume that the block size is 4-bits and Alice sends message M = 0110 1001 1101 0101 to Bob. Answer the following two questions:

   (1) what is the hash code H(M)?

   (2) show that this hash function is not secure by creating another message M' so that H(M') = H(M).

2. [10 points] Answer the following questions:

   (1) Give one difference between virus and worm.

   (2) Give one difference between user-mode rootkits and kernel rootkits

3. [30 points] Consider the following protocol. Kpub is the public key of B and Kpua is the public key of A. $\mathbf{N_A}$ and $\mathbf{N_B}$ are nonces.

   A → B $\mathbf{E(K_{pub}, [N_A, A])}$

   B → A $\mathbf{E(K_{pua}, [N_A, N_B])}$

   A → B $\mathbf{E(K_{pub}, N_B)}$

This protocol is vulnerable to an attack. Describe the attack.

Part 2 (80% of the total score):

Part 2 can be done using C/C++/Java/Python.

You will implement a client and a server using the **Secure Socket Layer (SSL)**. SSL enables to establish a secure connection between the server and the client.

Upon connection, the client prints "Connected to the server". Next, the client prompts the user to enter a message and sends the message to the server through the SSL connection. After the server receives the message, the server prints the message. Both the server and the client then terminate.

The server has at least one argument <port_number>, which specifies the port number used by the server. The port number is used to identify the server process on the machine, and is specified as a number between 1024 and 65535.

The client has at least two arguments: <server_domain> and <port_number>. <server_domain> specifies the server's domain name. The domain name is the name of the machine on which the server is running. After you log into remote.cs.binghamton.edu, it will show "remote01:~>", "remote02:~>" etc. If the server runs on remote01, then the domain name of the server is remote01.cs.binghamton.edu. If the server runs on remote02, then the domain name of the server is remote02.cs.binghamton.edu. <port_number> specifies the port number of the server. You can add other arguments to the client and the server if needed.

For example, if you use C, the server and the client can be invoked as:

```
./ser 2345
./cli remote01.cs.binghamton.edu 2345
```

If you use Java, the server and the client can be invoked as (you can use additional parameters if needed):

```
java SslServ 2345
java SslCli remote01.cs.binghamton.edu 2345
```

In addition, you need to generate a public key certificate (covered in Lecture 8) in order to establish the ssh connection. E.g., in C, you can use the following command to generate certificate.

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365
```

If you use java, you can use keytool to generate the certificate.

To compile your C program, please use the following commands:

```
gcc -Wall -w -o sslcli sslcli.c -I/usr/local/ssl/include/ -L/usr/local/ssl/lib -lssl -lcrypto
gcc -Wall -w -o sslserv sslserv.c -I/usr/local/ssl/include/ -L/usr/local/ssl/lib -lssl -lcrypto
```

Important Note

You can use any code available on the web for SSL socket programming. However, you must write your own code for the rest part of the assignment (i.e., prompting the user to enter the message, sending and receiving the message, and printing the message).   You should also generate the certificate by yourself. Please use your name to generate the certificate (other information can be forged).

Submission guideline

- Create a directory with a unique name (e.g. p3-[userid]), which contains the source code, the certificate, and a README file.
- **README** file (text file, please do not submit a .doc file) contains
    - ➢ Your name and email address.
    - ➢ Whether your code was tested on remote.cs.
    - ➢ How to compile and execute your program.
    - ➢ (Optional) Briefly describe your algorithm or anything special about your submission.

-Tar the contents of this directory using the following command.

tar –cvf p3-[userid].tar p3-[userid]

E.g. tar -cvf p3-pyang.tar p3-pyang/

- Upload the tared file you create above to brightspace.

Academic Honesty:

All students should follow Student Academic Honesty Code (**if you have not already read it, please read it carefully**). All forms of cheating will be treated with utmost seriousness. You may discuss the problems with other students, however, you must write your OWN codes and solutions. Discussing solutions to the problem is NOT acceptable. Copying an assignment from another student or allowing other students to copy your work may lead to an 0 in the assignment or an F in the course. Moss will be used to detect plagiarism in programming assignments. You need ensure that your code and documentation are protected and not accessible to other students. Use **chmod 700** command to change the permissions of your working directories before you start working on the assignments. If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate.