CS558 Assignment 3

Due: 11:59pm Nov. 9 (Thursday)

This assignment is done individually. You can use C/C++/Java/Python in this assignment.

Specifications

In this assignment, you will implement a client and a server that utilize **Secure Socket Layer (SSL)** for secure communication. Upon connection, the client prompts the user to enter their ID and password. The client then sends the ID and password securely to the server through the SSL connection.

The server maintains a file **"hashpasswd"** which has the following format;

<ID> <hashed password> <date and time when the password is created>

<ID> is a user ID that contains a sequence of lower-case letters and <hashed password> is a password hashed using the SHA or MD5 algorithm. The password must contain at least 8 characters.

You will write a program genpasswd.c/genpasswd.cpp/GenPasswd.java/genpasswd.py to generate the file "hashpasswd". Your program will be invoked as:

```
./genpasswd (C/C++)
java GenPasswd (Java)
python3 genpasswd.py (Python)
```

Upon execution of the above program, it will prompt the user to enter the ID. It will then check if the entered ID contains only lower-case letters, and if not, prints "The ID should only contain lower-case letters" and prompts the user to re-enter the ID. Otherwise, it will prompt the user to enter the password. Your program should check if the password contains at least 8 characters, and if not, print "The password should contain at least 8 characters" and prompt the user to re-enter the password.

If both the entered ID and password are valid, then your program will compute the hash of the password, and saves the ID, the hashed password, and the date and time when the password is created to file "hashpasswd". You can utilize existing implementations of SHA and MD5 such as those provided in libraries java.security and openssl in your implementation.

Next, your program will display the message "Would you like to enter another ID and Password (Y/N)?". If 'Y' is entered, then your program will prompt the user to enter another ID and password. If 'N' is entered, then your program will terminate. Your program should also check if the entered ID is already in file "hashpasswd". If so, the program will print "the ID already exists" and display "Would you like to enter another ID and password (Y/N)?".

You will implement an SSL server and an SSL client. The SSL server has at least one argument <server_port>, which specifies the port number at which the server accepts the connection request from the client. The port number should be a user-defined number between 1024 and 65535. As multiple students will test their programs on remote.cs, please use a unique port number (e.g., the last 4 digits of your B number) so that it is different from other students' port numbers.

The client has at least two arguments: <server_domain> and <server_port>. <server_domain> is the domain name of the machine on which the server is running. There are seven remote.cs machines: remote01-07.cs.binghamton.edu. For example, If the server runs on remote01.cs.binghamton.edu, then the domain name of the server is remote01.cs.binghamton.edu. When we test your program, we may execute your server and your client on the same or different remote.cs machines. <server_port> is the port number of the server. You can add other arguments to the client and the server if needed. If you use C/C++, your program needs to convert the domain name to the corresponding 32-bit IP address using **gethostbyname** function (https://man7.org/linux/man-pages/man3/gethostbyname.3.html).

If you use C, your Makefile should generate two executables **serv** and **cli**. If you use Java, your Makefile should generate **Serv.**class and **Cli.class**. If you use python, your python files should have name **serv.py** and **cli.py**.

Your server will be invoked as:

serv <*server_port*> (C/C++)
java Serv <*server_port* > (Java)
python3 serv.py <*server_port*> (Python)

Your client will be invoked as:

cli <*server_domain*> <*server_port*> (C/C++)
java Cli <*server_domain*> <*server_port*> (Java)
python3 cli.py <*server_domain*> <*server_port*> (Python)

When the client connects to the server, the client prompts the user to enter their ID and password. The client then sends the ID and password to the server through the SSL connection. After the server receives the ID and the password, the server computes the hash of the password and compares that hashed password against the password stored in file "hashpasswd". If the hashed passwords match, the server sends a string "Correct ID and password" to the client through the SSL connection. The client then prints the string and terminates. Otherwise, the server sends a string "The ID/password is incorrect" to the client, and the client prints the string and prompts the user to re-enter the ID and password. You can keep your server alive after the client terminates.

You need to generate a **public key certificate for the server** to establish the SSL connection. You do not have to generate a certificate for the client. You can use commands or programs to generate the certificate. The certificate generated should be stored as a file, which will be included in the assignment submission. For example, if you use C, you can use the following command to generate the certificate file "cert.pem".

```
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
```

You can use any code available on the web for SSL socket programming and for computing the hash of the password. However, you must write your own code for the rest part of the assignment (e.g. enter and verify ID and password, open/read/write files). You should also generate the certificate by yourself. Please use one of your name to generate the certificate (other information can be forged).

Grading guideline

- Correct execution format: 2'
- Readme: 2'
- Makefile (C/C++/Java): 6'
- Correct implementation of genpasswd: 35'
  - Storing hashed password in file "hashpasswd": 10'

    - Storing date and time in file “hashpasswd”: 5’
    - Error handling -- checking if the ID and password are valid: 10’
    - Error handling -- checking if the ID exists in file “hashpasswd”: 5’
    - Others 5’
- Correct implementation of SSL server and client: 55’ (C/C++/Java), 61 (Python)
    - SSL connection: 35’ (C/C++/Java), 41’ (Python)
    - Generating the public-key certificate for the server: 10’
    - ID & password verification: 10’

Submission guideline

- Create a directory with a unique name (e.g. p3-[userid]), which contains the client program, the server program, the program for generating the password file, the public-key certificate of the server, a makefile (C/C++/Java), 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 execute your program.
    - If any software is needed to be installed on remote.cs to execute your program, please provide the commands for installing the software on remote.cs.
    - (Optional) Briefly describe your algorithm or anything special about your submission that the TA should take note of.
- 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.binghamton.edu.

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 assignment description with other students, however, you must write your OWN programs. Discussing the algorithm or sharing the program is NOT acceptable. Copying an assignment from another student or allowing another student to copy your work may lead to the following:

1. **Report to the CS department and Watson college**
2. **0 in the assignment or F in this course.**

Moss will be used to detect plagiarism in programming assignments. You need to 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.