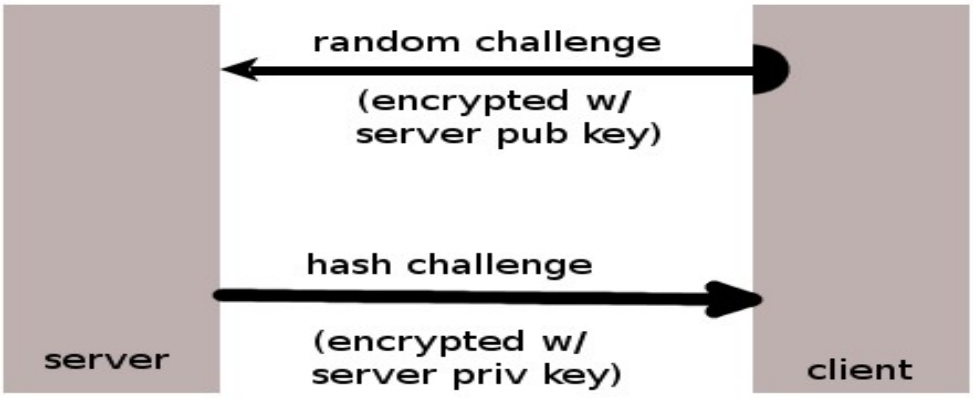
CS 165, Fall 2014
Cole Murray
12/08/14

CS 165 Computer Security Assignment 2

Summary:

For this assignment, I was given the task of implementing a simple file server using the openSSL library. The server supports both storage and retrieval of files. I also implemented a client that can send and received files to be stored at the server.

Structure:



Implementation:

client.c

Usage: ./client –serveraddress=00.11.22.33 –portnum=5555 –send ./file
./client –serveraddress=00.11.22.33 –portnum=5555 –receive ./file

Client generates a random challenge.

Client forms an SSL connection with the server and completes a handshake.

Client encrypts this challenge with the server's public key and sends to server.

Client receives server's hashed encrypted challenge.

Client decrypts server's hash with servers public key.

Client compares hash with own generated hash.

If the values match, proceed to handle file exchange.

Else, terminate the connection

server.c
Usage: ./client –portnum=5555

Server accepts an incoming connection
Server receives encrypted challenge from client

Server decrypts challenge.

Server hashes decrypted challenge, and re-encrypts using private key.

Server sends encrypted hash to client.

If hashes match, proceed to handle file exchange

else, terminate

Tests:

To test this project, I built it in modules. I individually tested the input/output with these modules using printf statements. This allowed me to ensure I was receiving and sending what I had intended.

Also, I used valgrind to resolve memory leaks.

Epilogue:

The most difficult part of this project was definitely the lack of documentation from the openSSL library. I was forced to rely on experimentation and google to discover more about the openSSL library. Also, using C as the primary language made things increasingly more difficult. I was forced to manage my own memory, and deal with the leaks present in the openSSL library.