Raymond Glier

Software Engineering

02/05/2018

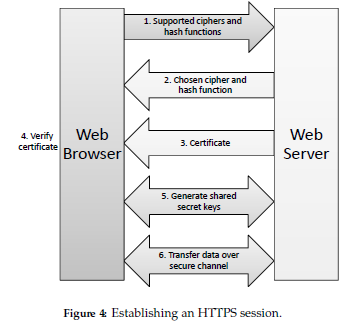
Kickoff Sprint

**User Encryption Research**

* We need to make sure ports on our server are associated with different security protocols. For example: the service provided by port X could be associated with the File Transfer Protocol (FTP). Port Y could be associated with Hypertext Transfer Protocol (HTTP). Port Z could be associated with the Hypertext Transfer Protocol over TLS/SSL (HTTPS). X, Y, and Z are just variables for actual port numbers that we don’t know yet.
* A careful look at the hosting service/server architecture that we decide to use will be required to make optimized security decisions.

**HTTPS Specifics**

* Using HTTPS as opposed to HTTP would be an optimal choice because an extra layer of security is crucial; users will not want to visit our site to make purchases unless they feel that we have done everything that we can to make sure their bank accounts/credit cards are safe during any and all transactions.
* HTTPS protocol solves problems concerning user confidentiality that HTTP does not solve. HTTPS provides another security layer called SSL (Secure Socket Layer) which we could use.
* There is also a newer implementation called TLS (Transport Layer Security).
* Both of these methods require a certificate which is used to confirm a server’s identity. This also creates a channel of communication between the web server and web browser. This channel is encrypted. Below is a diagram that illustrates specifics concerning the encrypted channel of communication between the browser and server.



* **More details concerning an HTTPS connection:**
* Browser sends HTTPS request to server, provides list of crypto ciphers/hash functions that are supported by the client program.
* Then, server searches through the list to find the strongest/most secure cipher and hash function. The cipher and hash function must be supported by both the browser and server.
* Next, the browser is notified of the choice made by the server concerning which specific encryption method was chosen, and sends back a certificate that includes the server’s public encryption key. The authenticity of the certificate is then verified by the browser.
* The browser will then encrypt a number at random using the public encryption key provided by the server. This can ONLY be decrypted using the private key held by the server.
* Starting at the number randomly generated, the server and client program create secret keys that encrypt/authenticate messages using a symmetric cryptosystem and MAC (message authentication code). \
* This concludes the details concerning the establishment of a secure channel. Once this secure channel is implemented, the HTTP communication works normally. A MAC is added to each HTTP message. Then the message is considered “authenticated” and is then able to be encrypted.

**Notes/Source Links and Information**

* We must figure out what hosting service we want to use, because we will not be able to plan out the specific implementation of an HTTPS protocol for our website until this is finished.
* All information from the book *Introduction to Computer Security* by Michael Goodrich and Roberto Tamassia, First Edition, Pearson New International Edition