ID: 180917

Course: Web & Mobile Application Development II

Date: March 22, 2019

Class Summary

In the class of the course shown above the instructor taught us how the flask web service can be improved by adding features such as user authentication and SSL certificates. Using user authentication enables the server to perform advanced functionalities such as data restriction whereas using SSL certificates ensures that the data being transferred between clients and the server is kept secure. The level of security of a particular certificate is dependent on how many bits it comprises of in the not too distant future 2048 bit (and even 4096 bit) certificates will become breakable by supercomputers and the average PC as technology advances and breakthroughs occur.

Certificates are used to provide some level of protection against the all infamous Man-In-The-Middle attack, granted that the attacker does now see the initial handshake. This is why it is very important that users use their own private network access point to prevent the possibility of said attack. Encryption with the use certificates works by providing connecting users with a public key. Only this public key can be used to both decrypt specific data sent by the server, However once data is encrypted with the public key it can only be decrypted by the private key which the server contains. Public keys are also used to encrypt data, thus resulting in a bidirectional encryption session. Currently, the only known way to compromise an encrypted message is through the use of brute force (trying all possible combinations until the data has meaning) however with today’s technology this will take an astronomically long time, even for a supercomputer given that the certificate is at least 2048 bits.

By design, browsers are shipped with a list of trusted certificate authorities to help differentiate between valid and invalid certificates. Although self-signed certificates work, they are not suitable for commercial use because of the possibility of fraud.

Class exercise tasks:

1. Apply the ID -> URL to all the GET/POST/PUT/DELETE functions so that the client always sees URLs instead of IDs, test if all works
2. Apply the http\_auth to all functions, test if the requirement is met
3. Apply HTTP 403 code instead of 401, test it
4. Try HTTPS + authorization, see what kind of certificate you have, and what are accepted by the browser
5. Apply for a real, free certificate and embed it in your web service

In order to apply for a free certificate on LetsEncrypt a registered domain name is required. After the domain name has been bought some the user needs to inject a verification link in the route. Once this has been done the certificate will be generated successfully.

1. Prepare your database, e.g. MySQL, for storing data (username/password/hash/…)

The sql file with the database information was created.