Coursework

Alexander Stradnic - 119377263

Practical 1

Number of different passwords = x passwords length 6, y len 7, z len 8

Permutations length 6 = Character permutations using all characters - (Permutations with only letters + Permutations with only numbers)
= (26*2 + 10)^6 - (52^6 + 10^6) = 37,028,625,920

Permutations length 7 = (26*2 + 10)^7 - (52^7 + 10^7) = 2,493,532,903,680

Permutations length 8 = (26*2 + 10)^8 - (52^8 + 10^8) = 164,880,277,053,440

Total = 6 + 7 + 8 = 167,410,838,583,040 combinations

Practical 2

A JSON Web Token (JWT) consists of 3 parts :

- A header, encoded in Base64
- `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9`
- The header defines the type of token being used (in this case JWT), as well as the encryption algorithm to be used.

- A payload, encoded in Base64
- `eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ`
- The payload consists of a set of claims, such as a user's ID, as well as other information such as expiration of the token.

- A signature, encoded in the algorithm specified in the header
- `SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c`
- The signature takes both the header and payload, already encoded in Base64, as well as a secret, and encodes them using the algorithm defined in the header.

Practical 3

Prepared statements separate data from any actual SQL statements. This processing prevents SQL injection from outside by processing data into text and then combining it with the SQL statement. An example of a prepared statement in PHP is below.

```
$statement = $connection->prepare('SELECT * FROM students WHERE name =
:name');
    $statement->execute([
        ':name' => 'Mark',
    ]);
```

Practical 4

Cross Site Scripting attacks occur when an attacker gains the ability to inject malicious scripts into an otherwise benign website. If they gain access, they are able to scrape sensitive information from an unsuspecting viewer, or from the website itself. The browser thinks that the script is from a trusted source and thus has no way to realise what is happening. A XSS Script has the ability to read sensitive data, change the HTML page, and much more, and there are many types of flaws that could allow a XSS attack to succeed.

Stored XSS Attack

A stored XSS attack begins when an attacker gains access to a web server. They then inject a script, which is then stored on the web server itself, and is executed when requests from users are made to the site.

Reflective XSS Attack

A reflective XSS attack is when an attacker makes a link, posted as a message or otherwise, that takes advantage of situations where user input is immediately rendered back to the user instead of being processed. Elaborate pages can be constructed this way, potentially tricking a user into submitting personal data that would then be sent to the attacker instead of to the website.

Practical 5

Cross Site Request Forgeries trick a user into submitting a malicious request to a website. They target state changes on the server, instead of retrieving data, as the attacker cannot retrieve the user's data with CSRF.

Preventing CSRF Attacks

There are a few different ways that CSRF attacks be prevented, such as :

- Using unpredictable Tokens
- A CSRF token is sent out to the user, at least once per session if not more. This token should be hidden in the body of the request, instead of being included in the URL of a request where it is more exposed and likely to be exploited

- Requiring the user to reauthenticate, or to prove they are a user, such as using credentials or CAPTCHA
- This forces the user to enter their details, which an attacker wouldn't know, or some other user-only action such as a CAPTCHA. This prevents requests from going out without the user knowing about them

Example using Token Authentication

Bob is checking his email. He clicks on a malicious email that contains a 0px Image with src=" http://site.com?submitPost='<a%20href=<malicious_link>>click%20here' ". However, the developers of site.com know about CSRF attacks and require their users to be authenticated with a CSRF token in order to submit a post. Thus the attack fails and no post from Bob's account is posted.