eJPT Certification Section

Section: Cross Site Scripting

09/21/20

**Learning Objectives:**

* Steal credentials
* Full control on a web app
* Gain unrestricted access to web application data

**Notes:**

**SQL Injections**

* Most web apps use a database to store data
* To interact with the data within the DB you use a language called SQL (structured query language.

A close up of a computer

Description automatically generated

* SQL Injection (SQLi) attacks allow an unauthorized user to take control over SQL statements used by a web application.

**Understanding SQL Syntax**

* **A screenshot of a social media post

  Description automatically generated**SQL Statements:
  + In this SQL statement we are searching for the name and description of a record that exists in table called products. In this table the selected record will have an id value equal to 9.
* You can use two types of symbols to add comments to SQL statements:
  + #
  + –

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generatedSELECT Example**

**A screenshot of a cell phone

Description automatically generatedUnion Example**

**SQL Queries Inside Web Applications**

* The following is an example of a static query inside of a PHP page.

**A screenshot of a cell phone

Description automatically generated**

* + **$connection** is a n object referencing the connection to the database.
  + **$query** contains the query
  + **$mysqli\_query()** is a function which submits the query to the databse.
  + **display\_results()** function renders the data from the database to the attacker

**A screenshot of a cell phone

Description automatically generated**

* **Vulnerable Dynamic Queries**
  + **A screenshot of a cell phone

    Description automatically generated**Most of the time queries are not static as shown above in the previous example, but rather they are dynamically built.
  + **$id** is a parameter which receives a value dynamically via GET request.
* A screenshot of a cell phone

  Description automatically generated**Example of Vulnerable Dynamic Query**

**A close up of text on a white background

Description automatically generated**

**Example of SQL UNION**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Find SQL Injections**

* SQL injection vulnerabilities can be found where there are user inputs:
  + GET parameters
  + POST parameters
  + HTTP Headers
    - User-agent
    - Cookie
    - Accept
    - **A picture containing table

      Description automatically generated**Etc
  + Check one injection at a time.