Basic Injection

ctflearn.com

Jakub Kazimierski February 2023

1 Introduction

This sheet meant to be a walkthrough of CTF challenge from ctf [4]. Here I try to explain step by step what metodology for solving this problem was taken along with actions which were needed. Some of steps will be showed on screenshots and some just explained along with commands used.

OS which was used for this presentation is Kali Linux in version 2022.4

2 Problem overview

Description of this task is short, we should exploit site: [5]. When we open link we get home page:

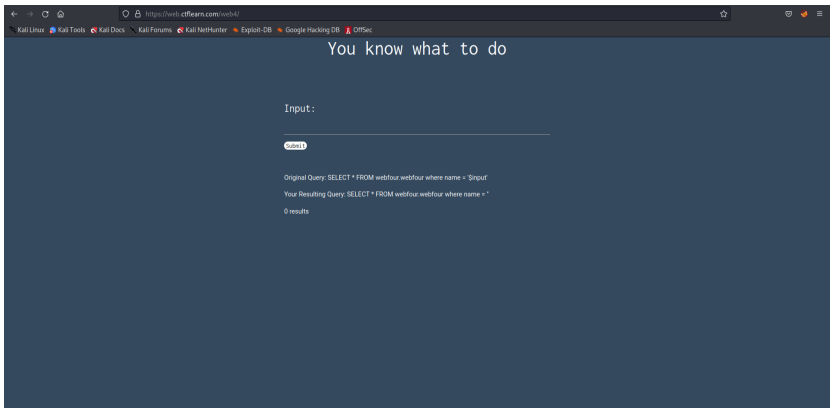

Figure 1: Site home page

It seems like we should insert some kind of data to obtain list of results. Let's inspect this page so maybe we will find some more informations.

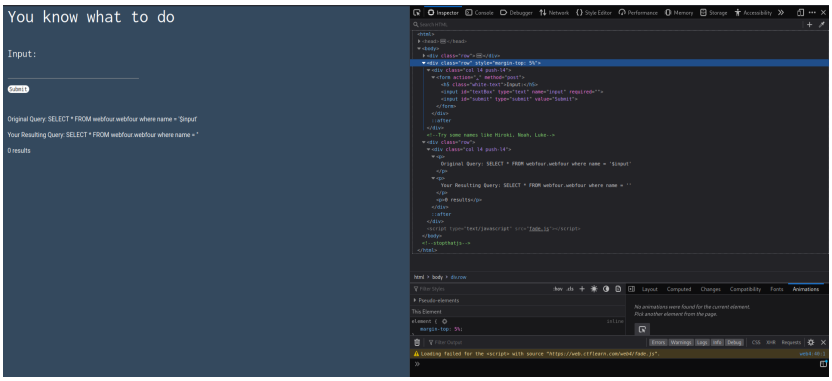

Figure 2: Inspecting page

Ok in suh case let's try some of those names in comment, I inserted **Luke**

Figure 3: Returned output

We get two columns with data as output, probably in this table flag is hidden somewhere, but we dont know what value **$input** should have to get it. Let's look again at query which is returning data.

```
SELECT * FROM webfour.webfour where name = '$input'
```

Listing 1: Executed query

There is possibility that if **$input** is not validated [1] or script which executes query on database does not have preprared statements [3], we can execute SQL Injection [6] on this database. Shortly this kind of attack is based on inserting

piece of SQL query instead of valid input data which were expected. How can we applay it in our case? At this point take note, that syntax below is for MySQL, as this is one of most popular relational database management system which uses SQL. Let's see below:

```
SELECT * FROM webfour.webfour where name = '' OR 1=1 --
```

Listing 2: Injected query

First we escape quotation mark: ' , so after that next part of our input will not be taken as name. The OR operator displays a record if any of the conditions separated by OR is true. 1=1 is always true condition. – Is comment, to make rest of this query invalid, make sure that after sign – space is inserted, because according to MySQL documentation [2] it requires at least one whitespace character after the double dash for it to be registered as a comment. And after we input this we should get all rows from this table:

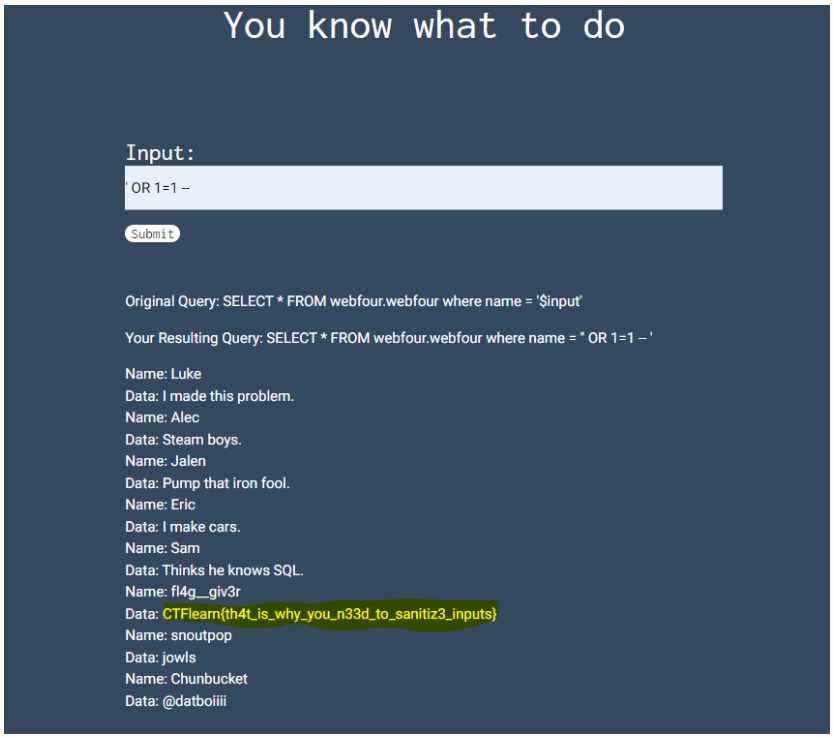

Figure 4: Flag

It appears that we obtained a flag which is: **CTFlearn{th4t_is_why_you_n33d_to_sanitiz3_inputs}**

References

[1] Input validation. https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html.

[2] Mysql documentation. https://dev.mysql.com/doc/refman/5.7/en/comments.html.

[3] Prepared statements. https://www.w3schools.com/php/php_mysql_prepared_statements.asp.

[4] Reference to site with this challenge. https://ctflearn.com/.

[5] Site to exploit. https://web.ctflearn.com/web4/.

[6] Sql injection. https://www.w3schools.com/sql/sql_injection.asp.