

Incident Report: MR-10189-OffensiveUserLogin

Date: 10-7-2021

Executive summary:

I was tasked with logging into a user's account without the password. This task did not require any special tools. I chose to go without any tools because I know that SQL Injections are doable without any tools.

Results

Application Details:

- **Application URL:** <https://mr-ttabookstore-104.herokuapp.com/#/login>
- **User accounts:** bender@tta.bookstore


Conclusion:

As stated earlier, I was tasked with gaining access to a user's account without the password. Because I was able to previously log into the admin account I knew what the email syntax looked like. For example, the admin's email was: admin@tta.bookstore. Knowing this info was VERY helpful. I knew the name of the user's account I was trying to compromise, which was Bender. Knowing that much I was able to make an educated guess and assume that Bender's email was bender@tta.bookstore. So in the email field, I inputted bender@tta.bookstore' AND 1=1 --. Because that was indeed Bender's email, that worked. The ' AND 1=1 -- allowed me to tell the database to select all accounts with the email bender@tta.bookstore. The first quote closed the statement after the email. At first, I kept trying to input OR instead of AND. This just kept logging me into the admin's account. Using AND instead told the database that it was looking for Bender's account AND 1 had to equal 1, which we all know is always true. In the password field, you can input whatever you want because the two dashes comment out everything that follows so the database ignores what was inputted into the password

field. Here is the flag I got: 5ff5052e879e6fef64124e64c82c84ebc809c6c4. Below is a screenshot of the flag as well.

You successfully solved a challenge: Login Bender (Log in with Bender's user account.)

 5ff5052e879e6fef64124e64c82c84ebc809c6c4

 Copy to clipboard