Lab *04* – SQL Injection Attack Lab

<*Yibiao Liao sgw091*>
IS 4463-CY1 – Summer 2019
<*8/08/2021*>

INTRODUCTION

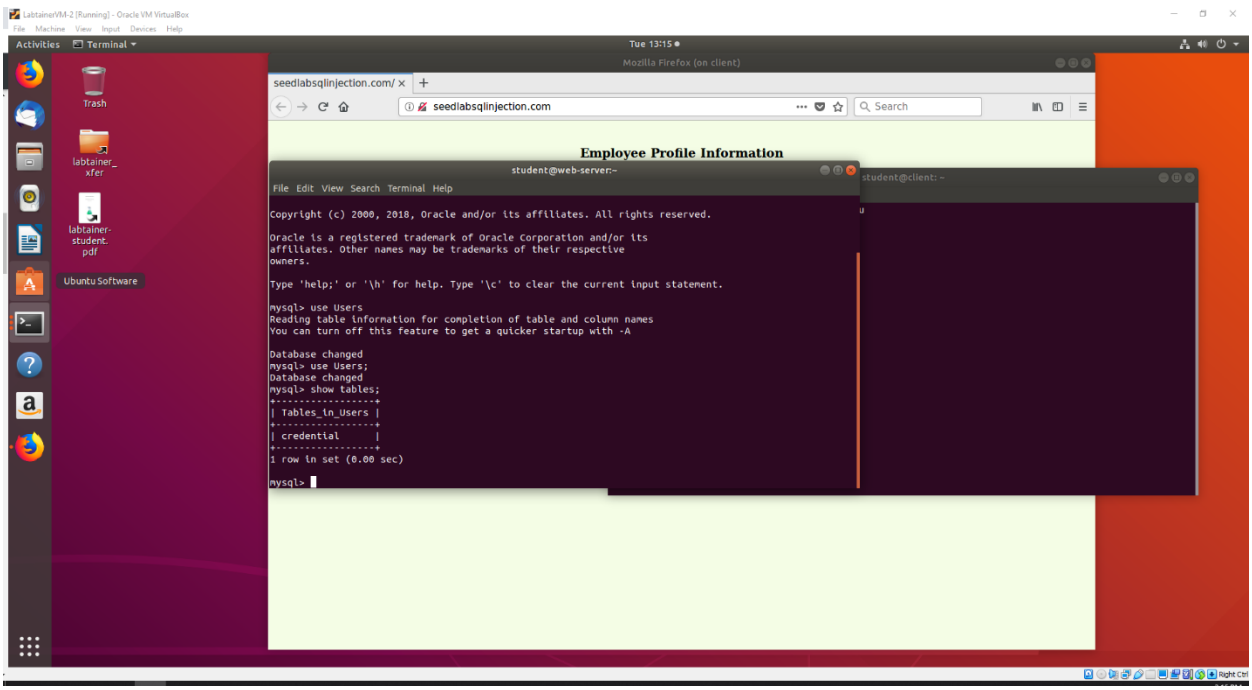
SQL injection attack is an attack by inserting a malicious SQL query or add statement into the input parameters of the application and then parsing and executing it on the backend SQL server, it is currently one of the most common means of hacker attacks on databases.

The main threats posed by SQL injection are as follows

- Guessing the backend database, which is the most exploited way to steal sensitive information from websites.
- Bypassing authentication, such as bypassing authentication to log into the backend of a website.
- Injection can be done with the help of stored procedures in the database for operations such as lifting rights

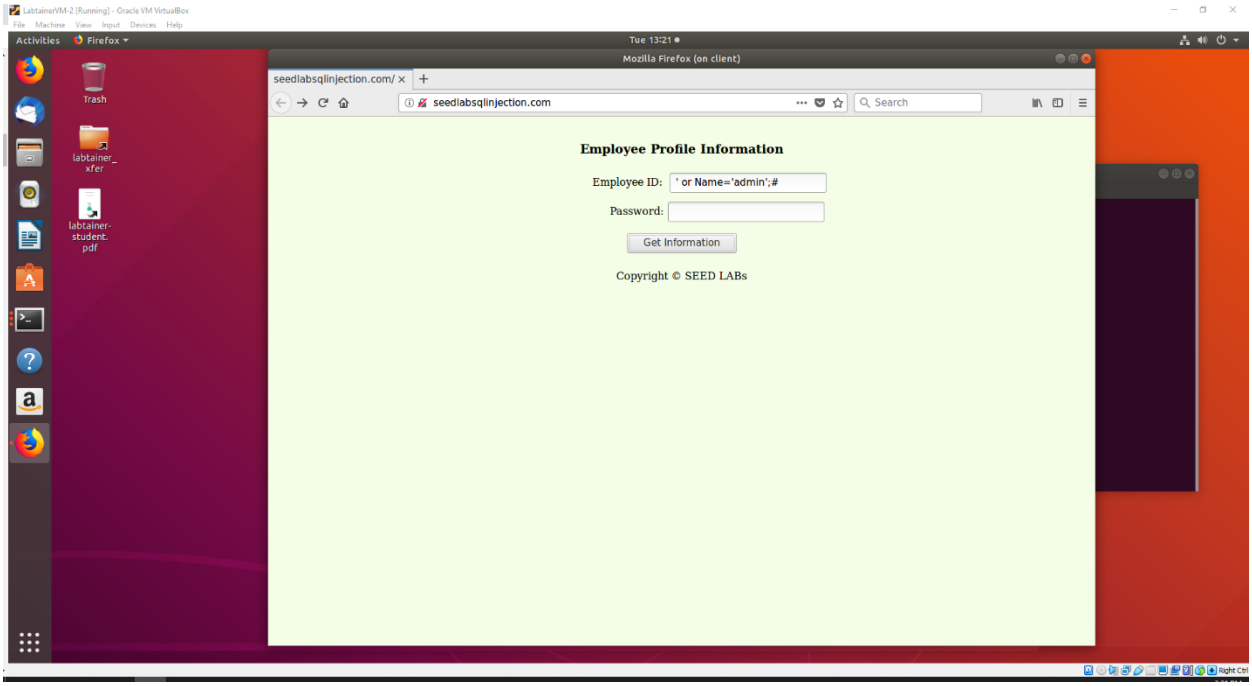
PROCEDURE

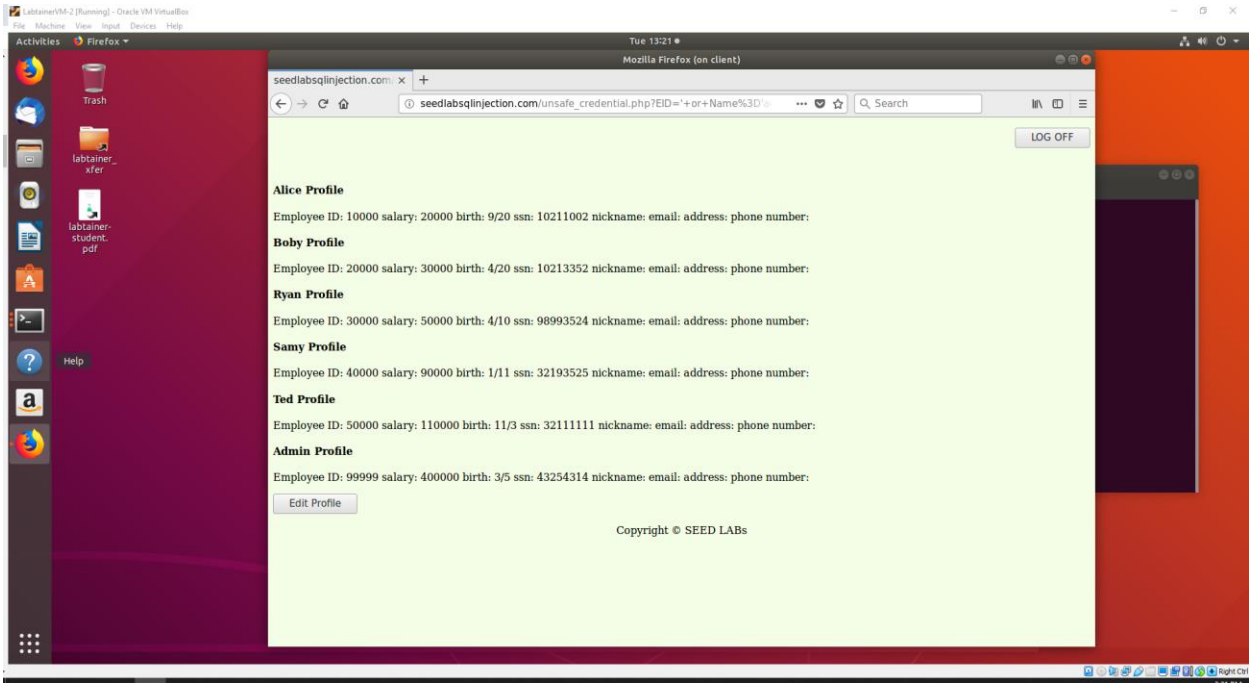
Task 1: MySQL Console



Task 2: SQL Injection Attack on SELECT Statement

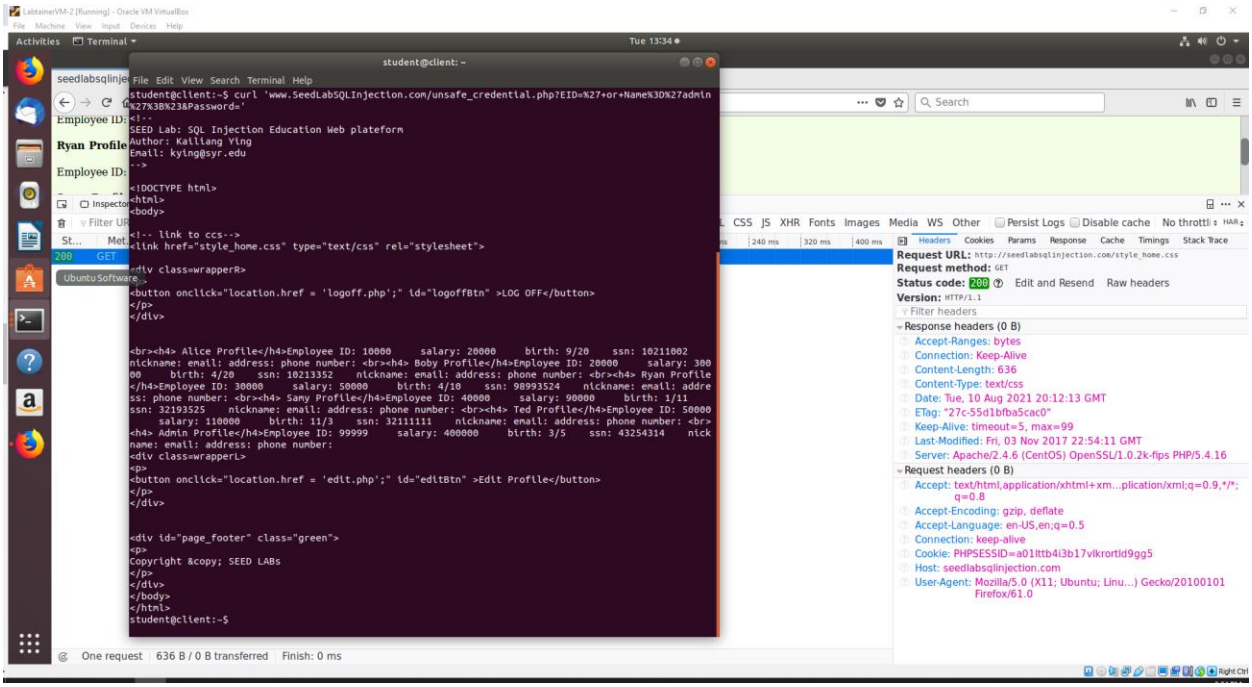
2.1: SQL Injection Attack from webpage





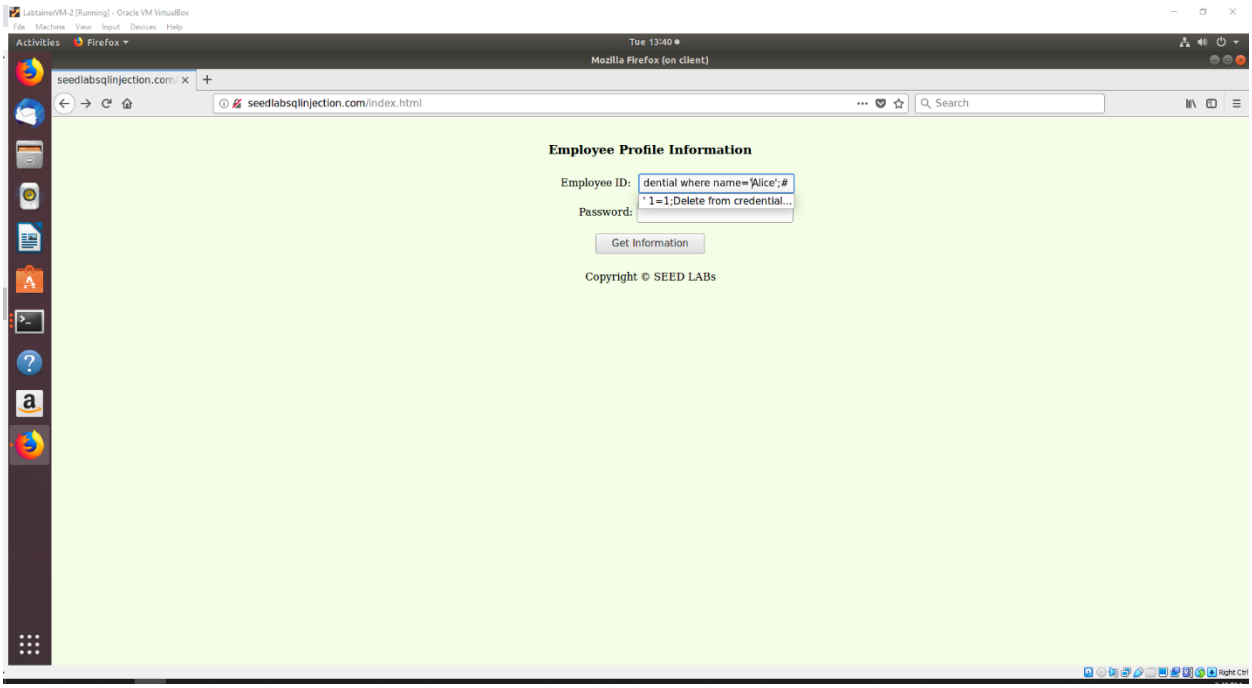
Use the code ‘ or Name=’admin’;# to exploit the attack. The single quotation closes the input id parameter, and the OR statement that follows permits us to login as admin. The # is added at the end to comment out everything that comes after it, allowing the password input to be bypassed.

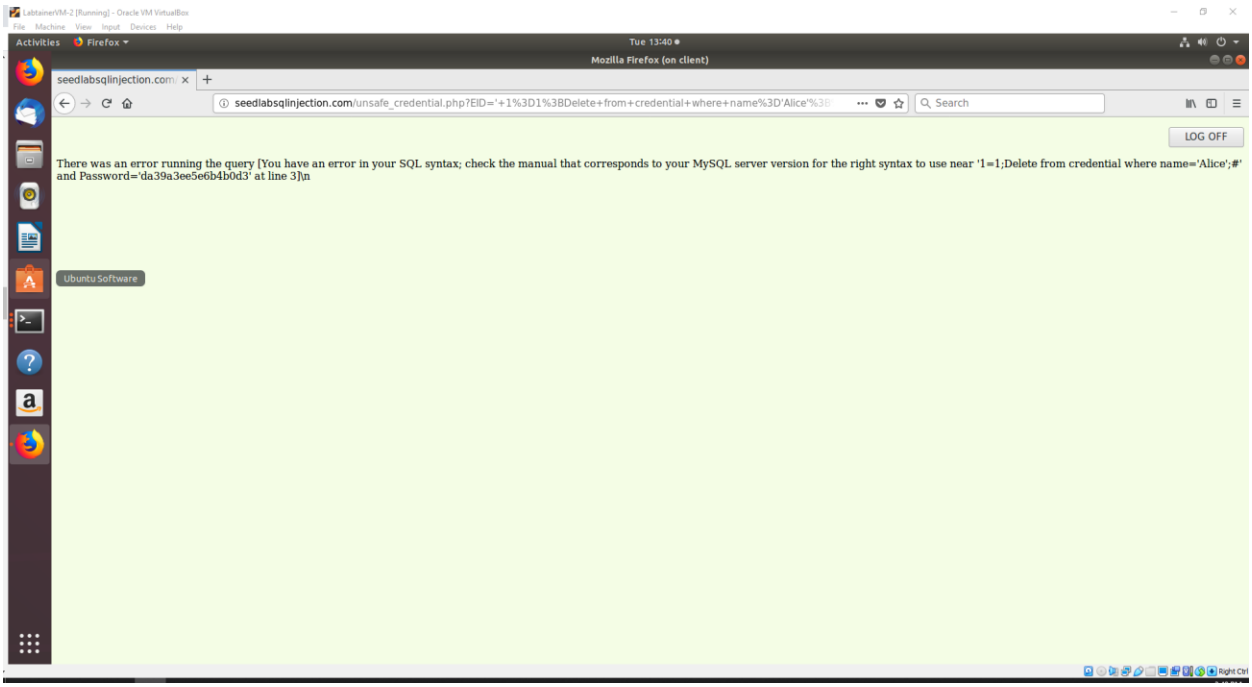
2.2: SQL Injection Attack from command line.



We need to encode special characters to run the attack from the command line. So, when conducting the attack from the webpage, we may retrieve the URL by looking at the "LiveHTTPHeaders." If the assault is successful, the command prompt will show all the information.

2.3: Append a new SQL statement.

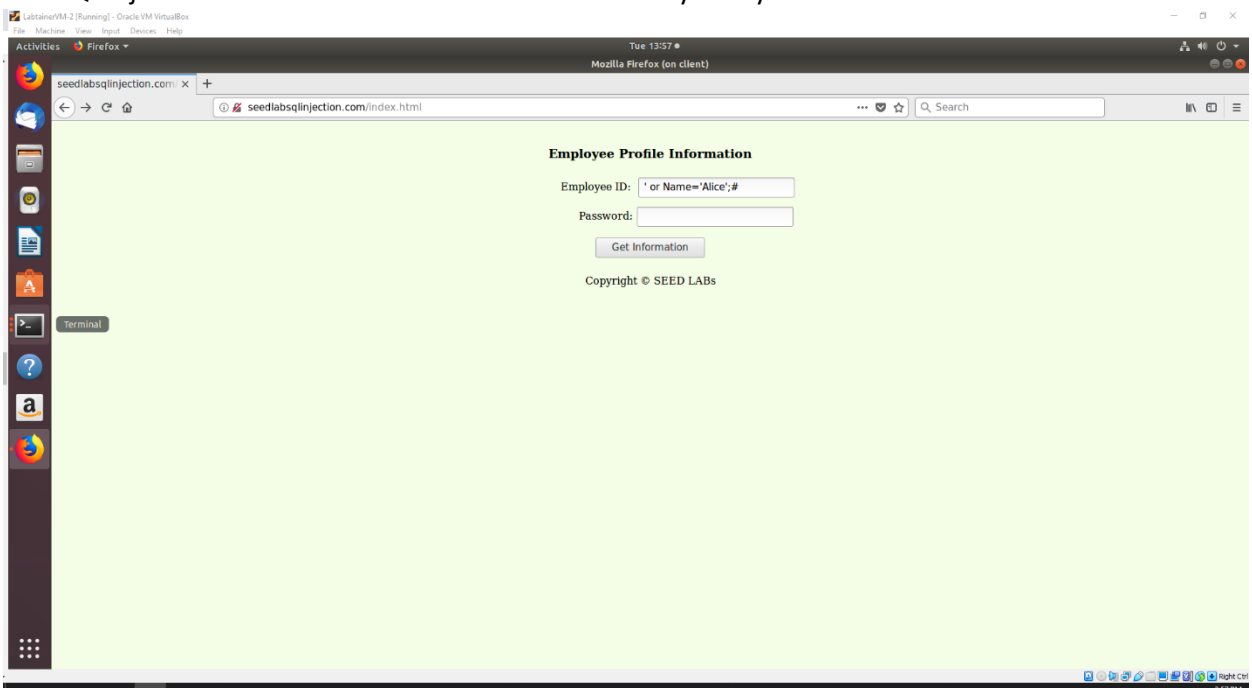


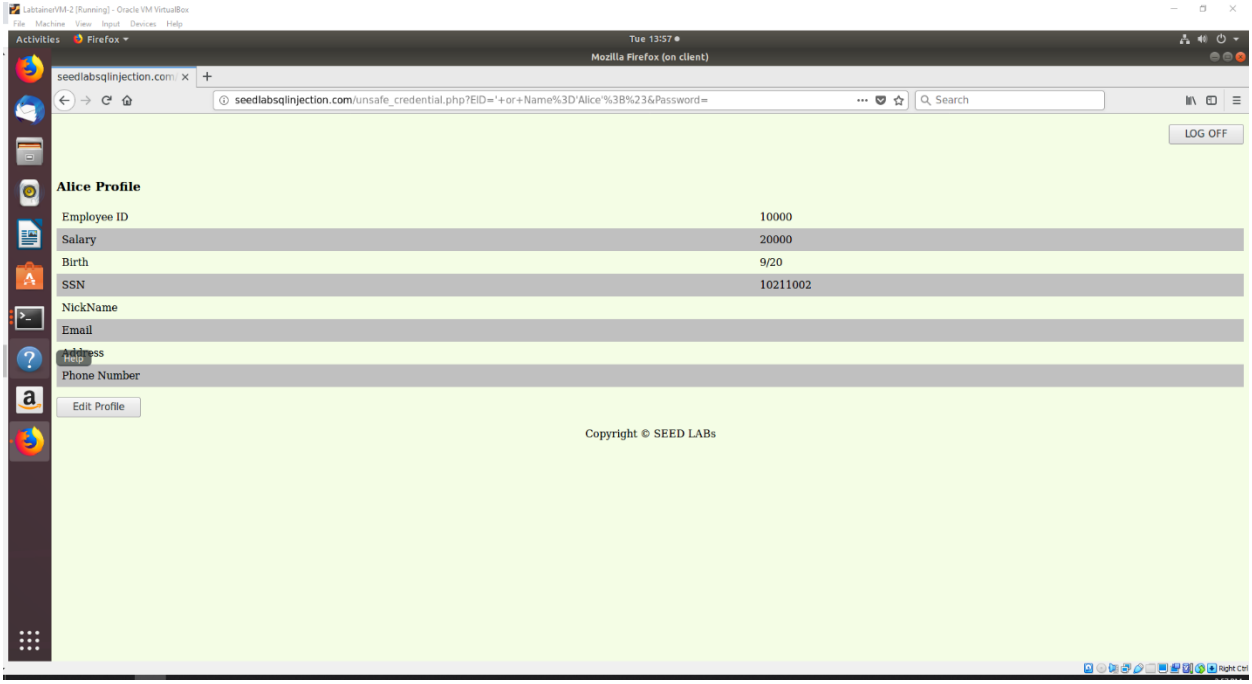


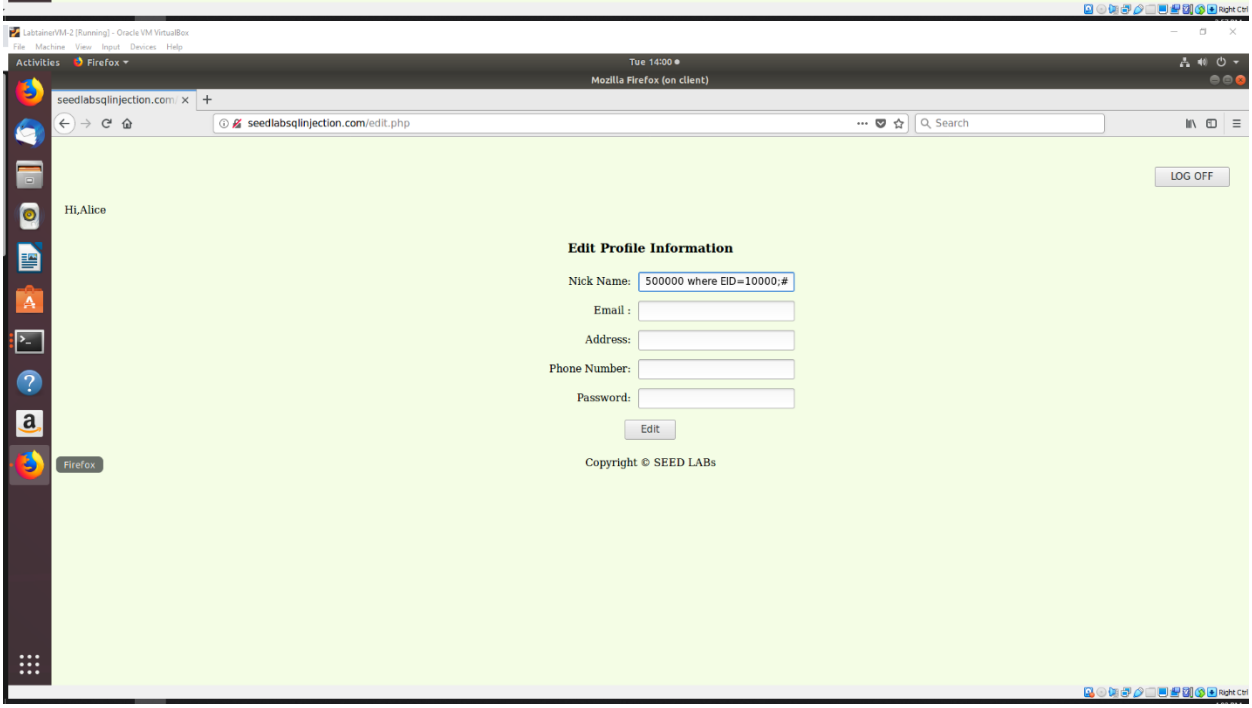
The attack fails due to a MySQL countermeasure that prevents numerous statements from being executed when called from PHP.

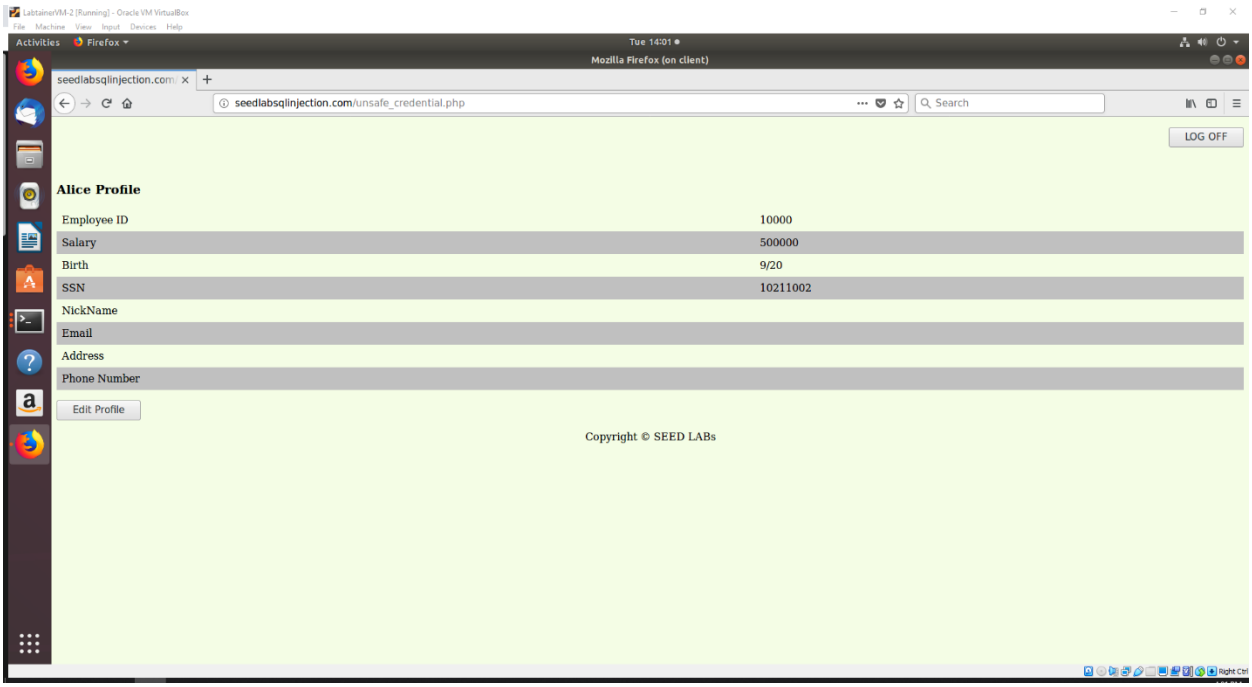
Task 3: SQL Injection Attack on UPDATE Statement

3.1: SQL Injection Attack on UPDATE Statement — modify salary



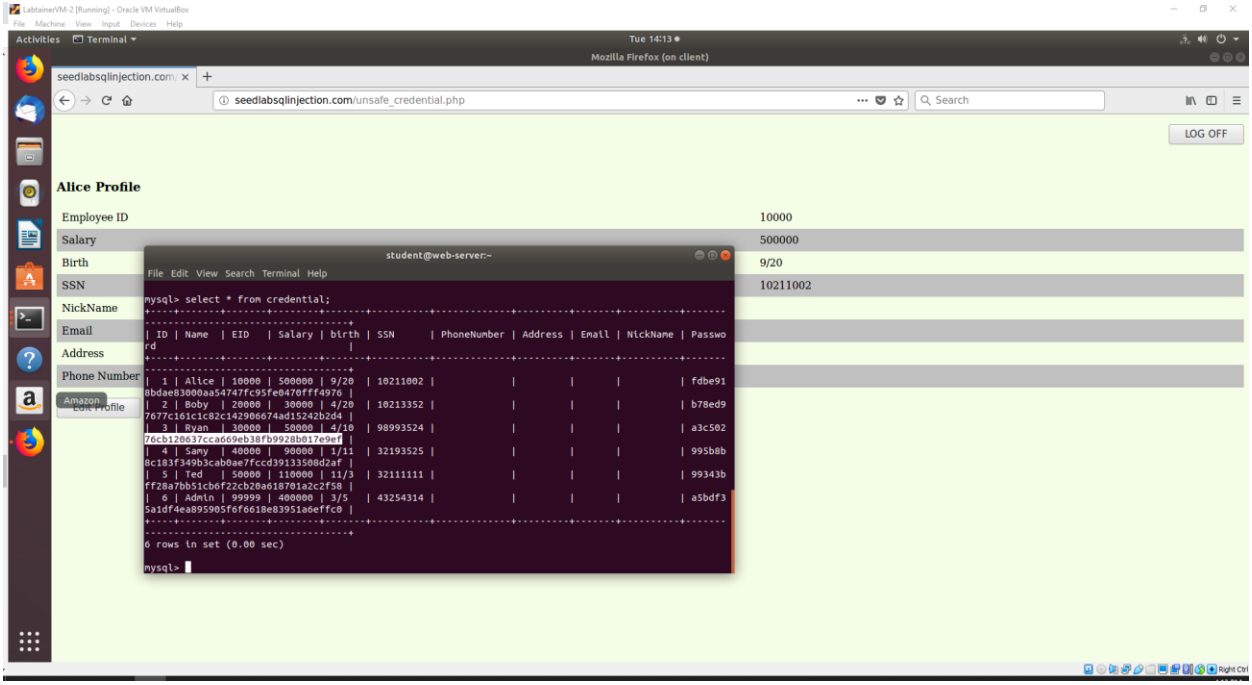

Activities
Firefox
Tue 13:57
Mozilla Firefox (on client)
seedlabsqlinjection.com
seedlabsqlinjection.com/unsafe_credential.php?EID='+or+Name%3D'Alice'%3B%23&Password=
Search
LOG OFF
Alice Profile
Employee ID 10000
Salary 20000
Birth 9/20
SSN 10211002
NickName
Email
Address
Phone Number
Edit Profile
Copyright © SEED LABs

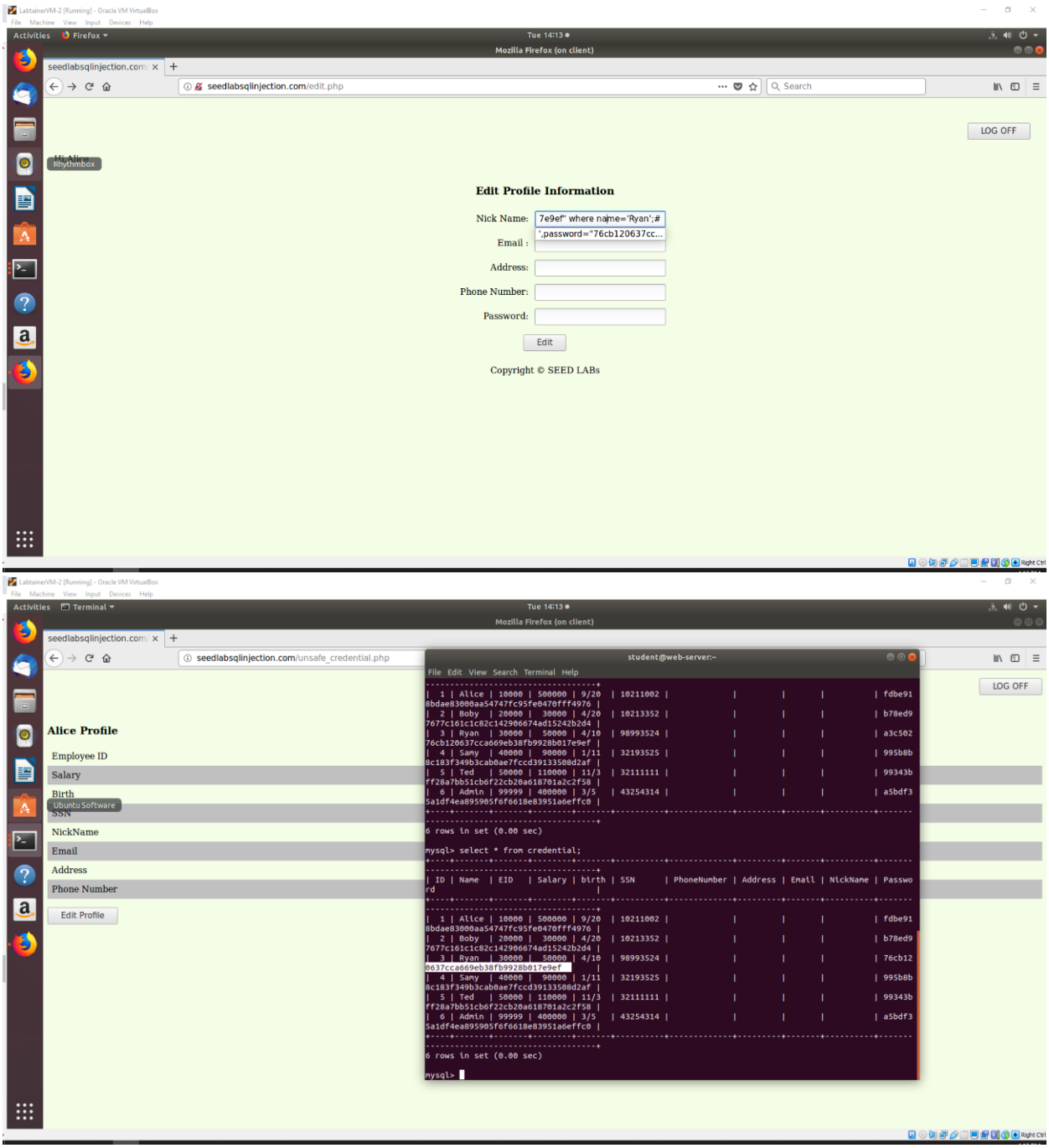

Activities
Firefox
Tue 14:00
Mozilla Firefox (on client)
seedlabsqlinjection.com
seedlabsqlinjection.com/edit.php
Search
LOG OFF
Hi,Alice
Edit Profile Information
Nick Name: 500000 where EID=10000;#
Email :
Address:
Phone Number:
Password:
Edit
Copyright © SEED LABs



We're attempting to attack a SQL injection vulnerability by injecting code into the edit profile page to change the current employee's pay. We use a "#" at the end to comment out any subsequent values, ensuring that we don't run into any issues with null or erroneous input values from other fields. And we carry out this attack and change the salary field, which is hidden since the employee is not permitted to alter it. It can only be edited by the administrator. The pay of Alice has been adjusted because of the successful attack.

3.2: SQL Injection Attack on UPDATE Statement





The update command is used to change the password of a different account (Ryan) from another account (Alice). The SQL Injection vulnerability is exposed because of this. This demonstrates how hazardous it may be. We attempt to edit Alice's profile by logging into her account. The password of Ryan is changed when we enter the attack vector into the nickname box, and if the attack is successful. The update statement is used on the edit profile page to update the fields of an account; however, we utilize the injected code to change the information of another account. The # sign at the conclusion of the attack vector is used to comment out all subsequent code in the original code, ensuring that the attack is not hampered.

CONCLUSION

Please provide a conclusion based on these prompts:

- What are the goals of the techniques and tools used in this lab? How are they relevant to real-world security?
  MySQL and HTML are both involved in this lab. As the most used database language, knowing how to control the safety of the database is crucial.
- How effective are these techniques and tools, in your opinion?
  Very effective, but some extra research were needed
- What is your opinion of this lab section in terms of difficulty and relevance? What other aspects of this topic would you like to explore?
  Its challenging.

REFERENCES

Internet Resources

https://www.youtube.com/watch?v=_P8HCLkDInA&feature=youtu.be