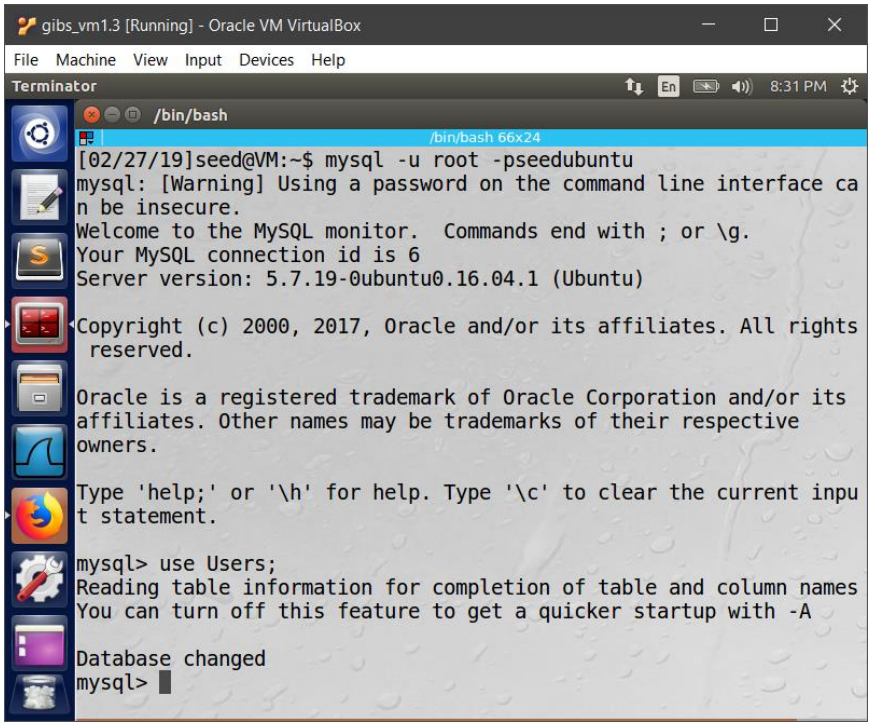
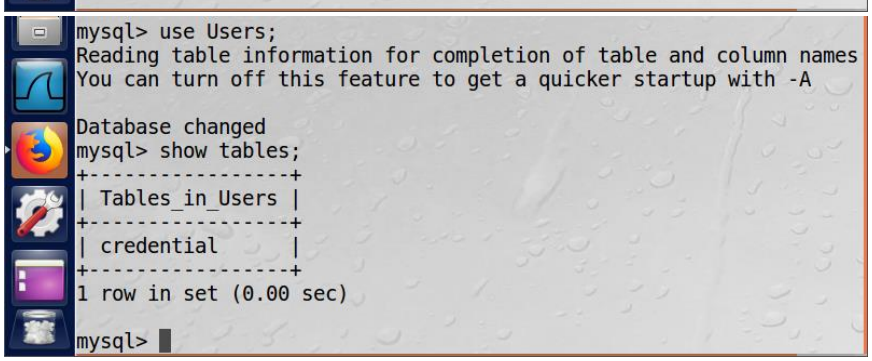
Gabriel Brehm
CS363

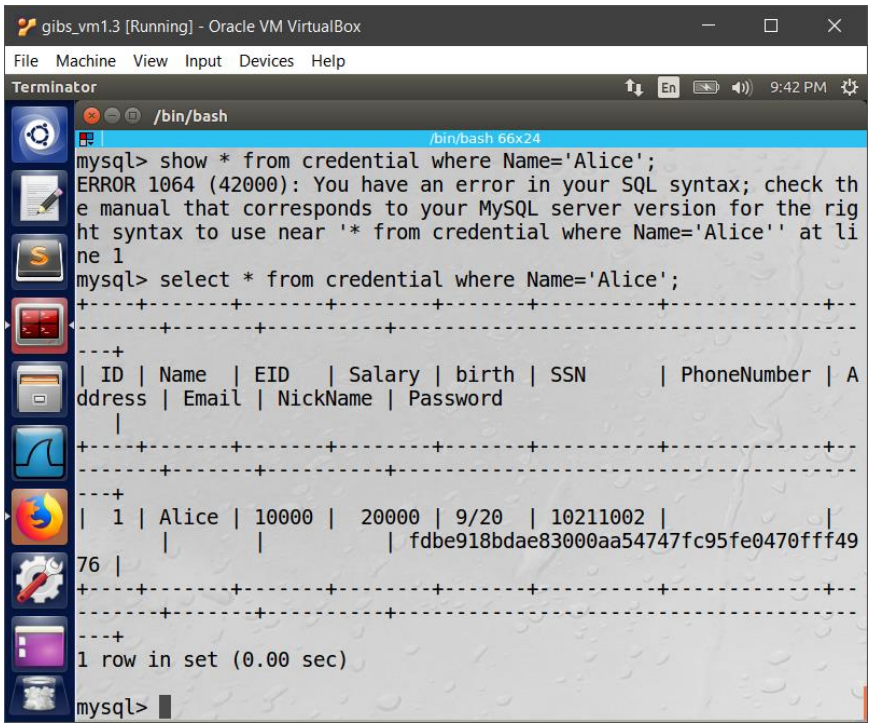
Lab 4

3.1 Task 1: Get Familiar with SQL Statements

Here we logged in to MySQL, used the Users database, showed the tables, and printed the profile information for Alice.



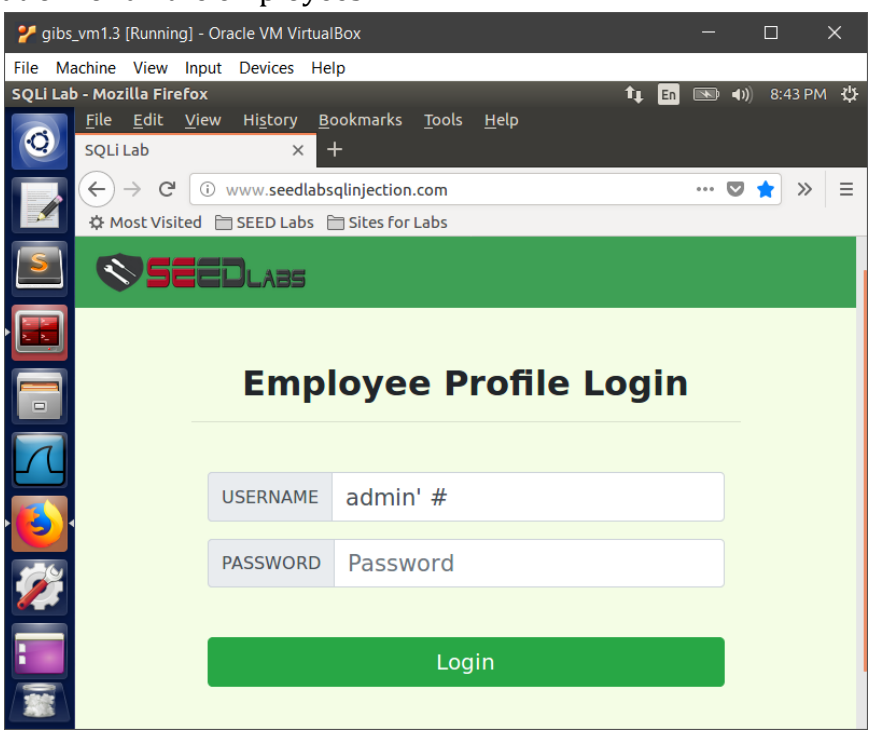


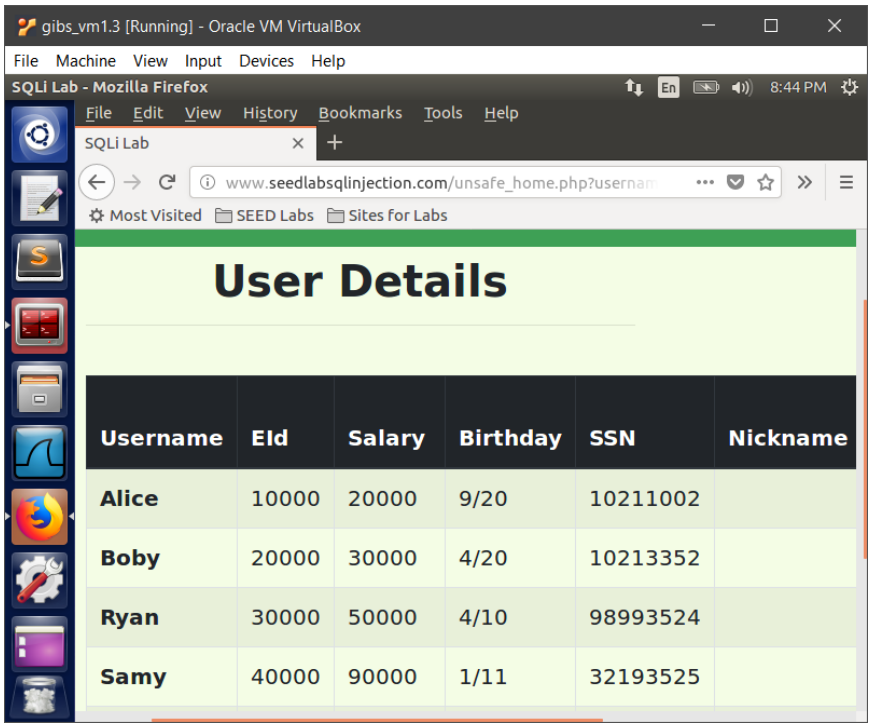


3.2 Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

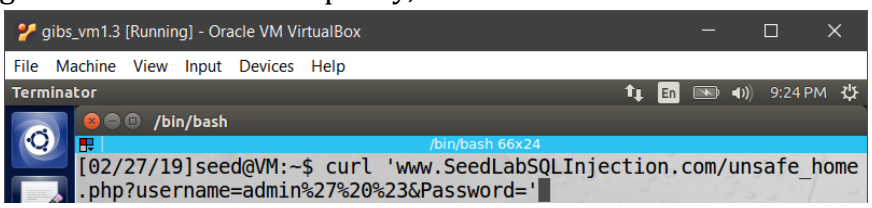
Here we type "admin' #" into the username box, and we can see we get all of the profile information for all the employees.

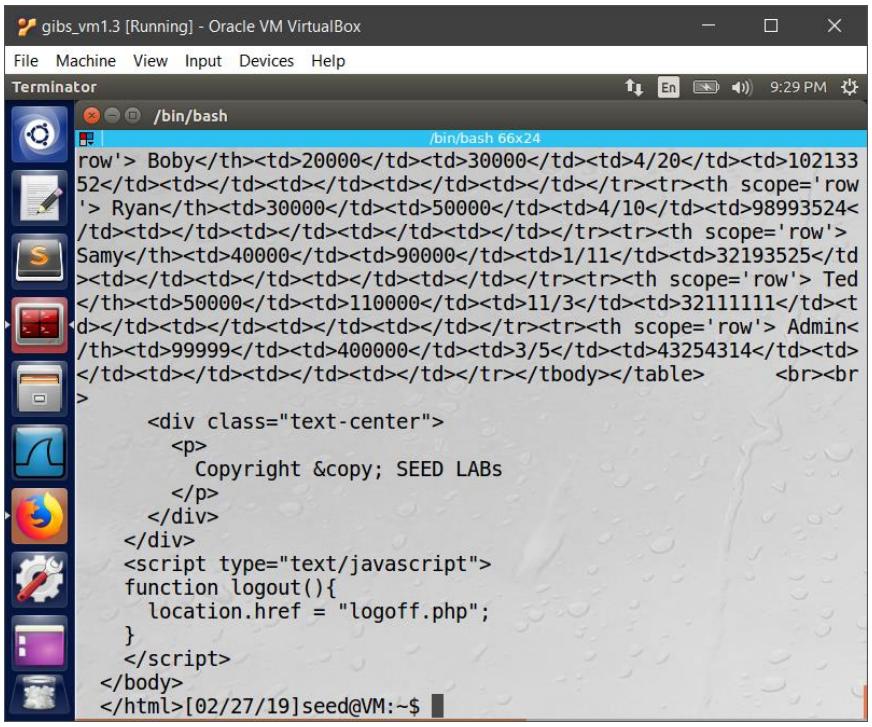




Task 2.2: SQL Injection Attack from command line

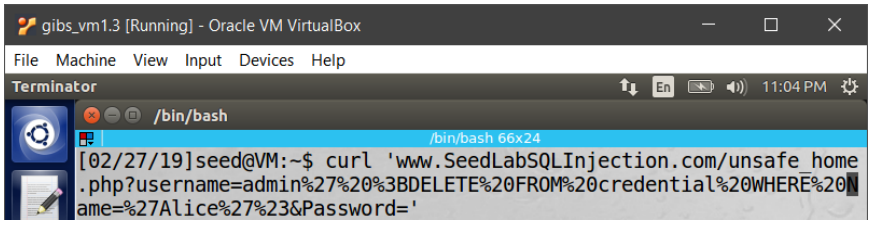
Here we did the same thing as the previous step, but we used curl instead of using the webpage. The format is not so pretty, but the information is there.

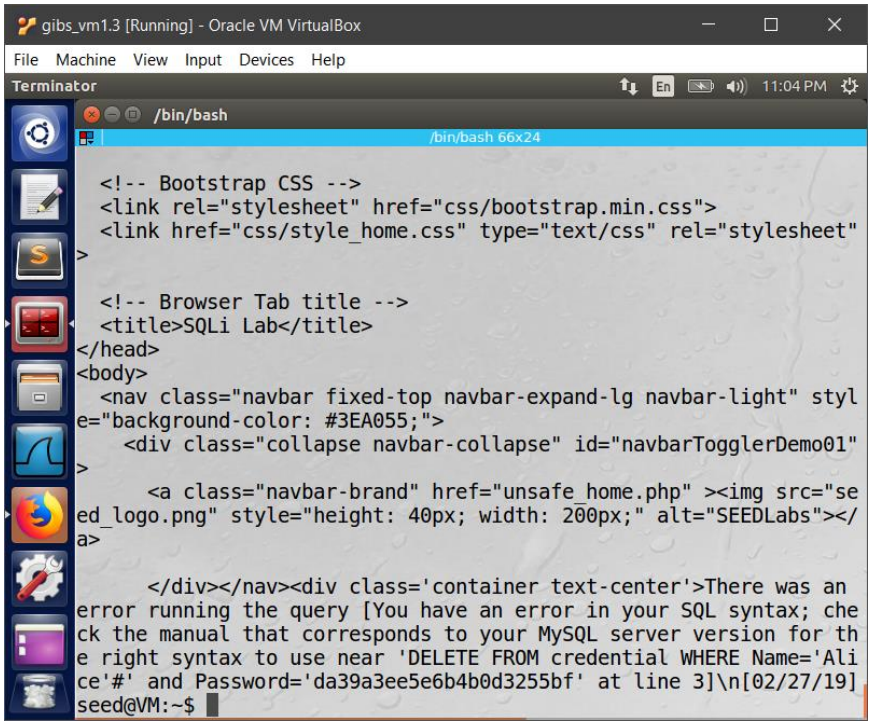




Task 2.3: Append a new SQL statement

Here we attempted to append an additional SQL statement, but we get the error seen at the bottom of the second picture. Multiple types of SQL statements were appended, but none of them worked. The textbook seems to imply that this was supposed to happen.

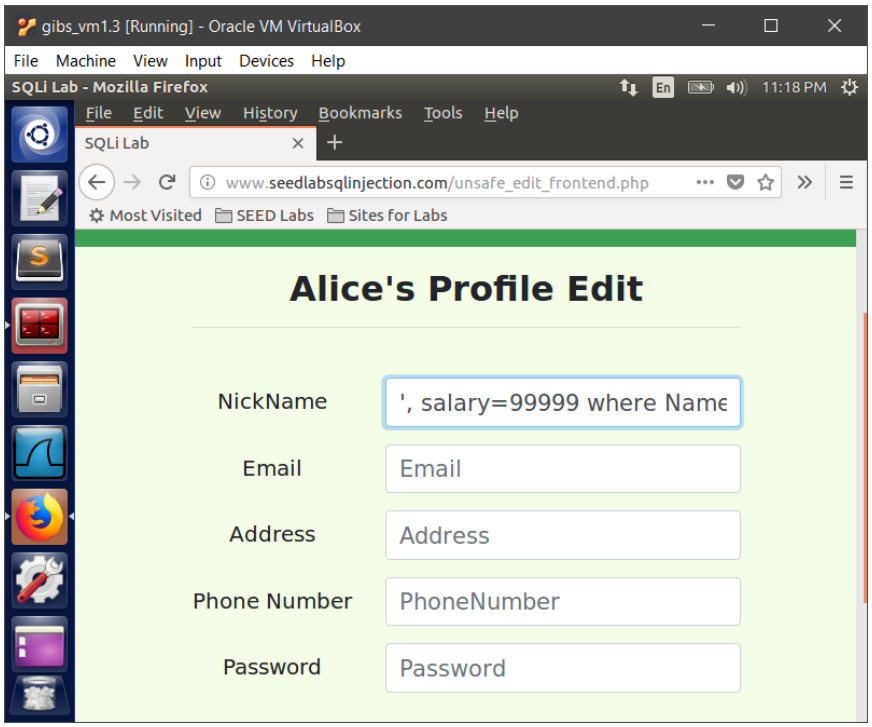


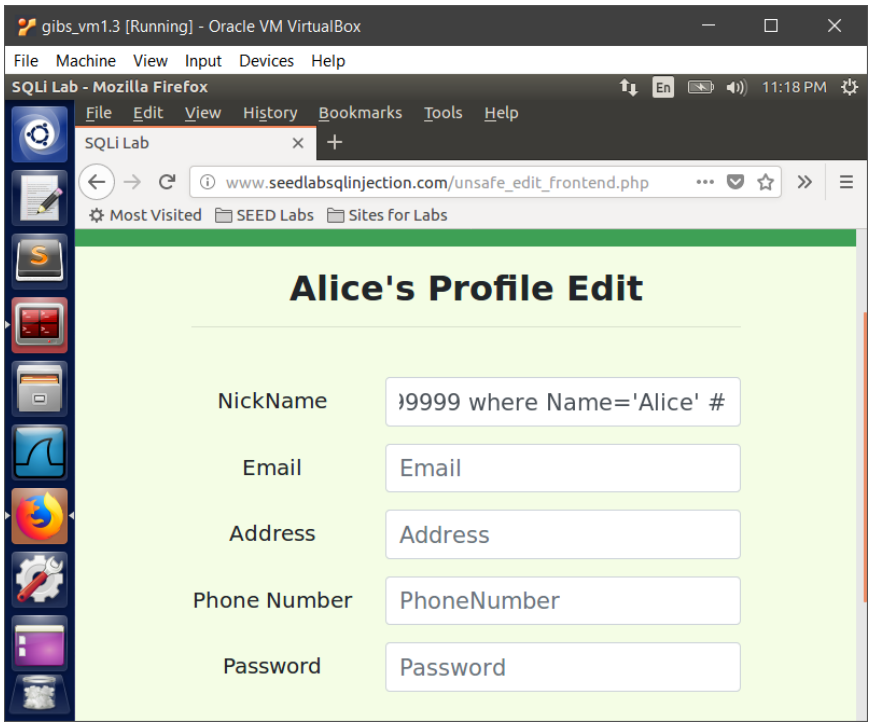


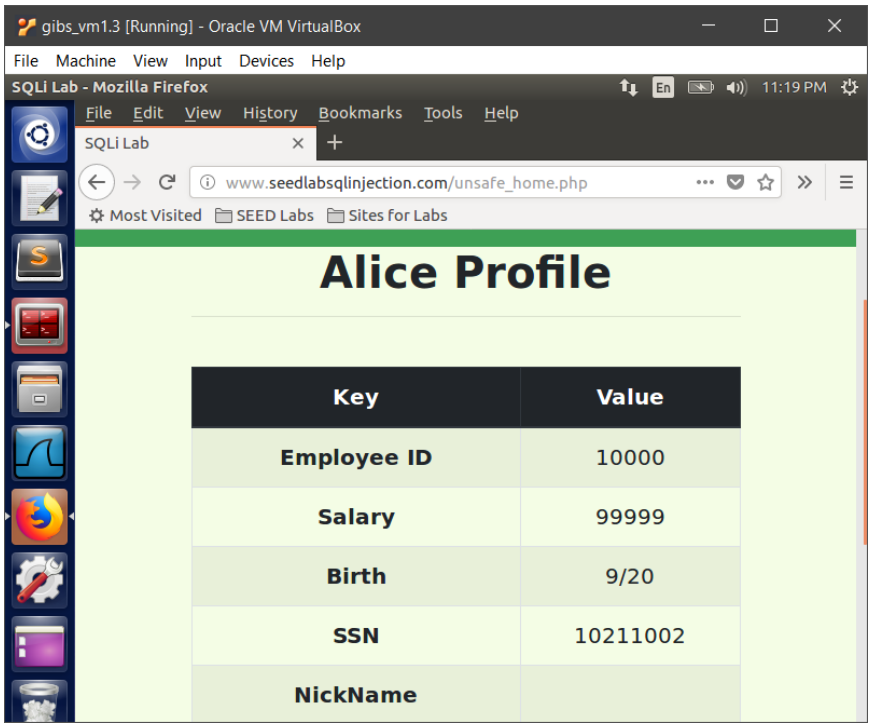
3.3 Task 3: SQL Injection Attack on UPDATE Statement

Task 3.1: Modify your own salary

Here we simply set the nickname to "', salary=00000 where Name='Alice' #" which resets Alice's salary to 99999. As one can see based on the third image, the salary was indeed changed.

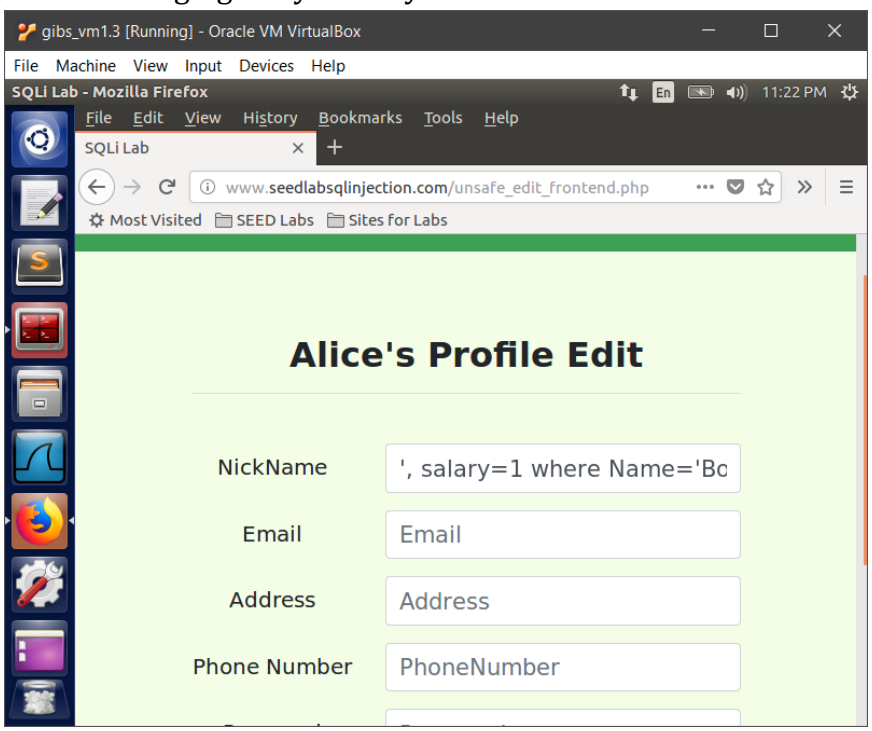

gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SQLi Lab - Mozilla Firefox
11:18 PM
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
Most Visited SEED Labs Sites for Labs
Alice's Profile Edit
NickName
', salary=99999 where Name
Email
Email
Address
Address
Phone Number
PhoneNumber
Password
Password

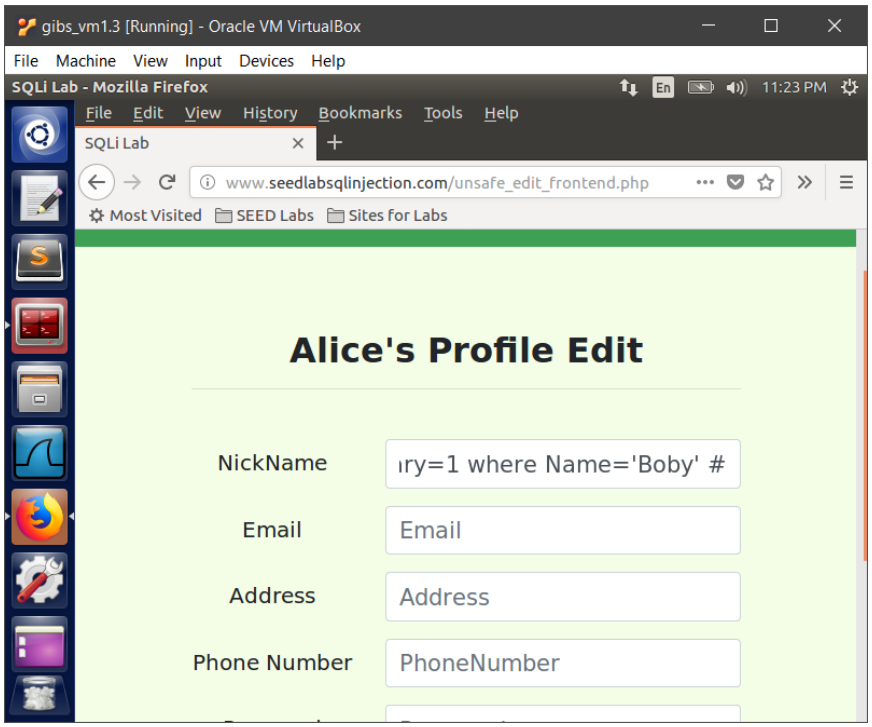

gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SQLi Lab - Mozilla Firefox
11:18 PM
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
Most Visited SEED Labs Sites for Labs
Alice's Profile Edit
NickName
)9999 where Name='Alice' #
Email
Email
Address
Address
Phone Number
PhoneNumber
Password
Password

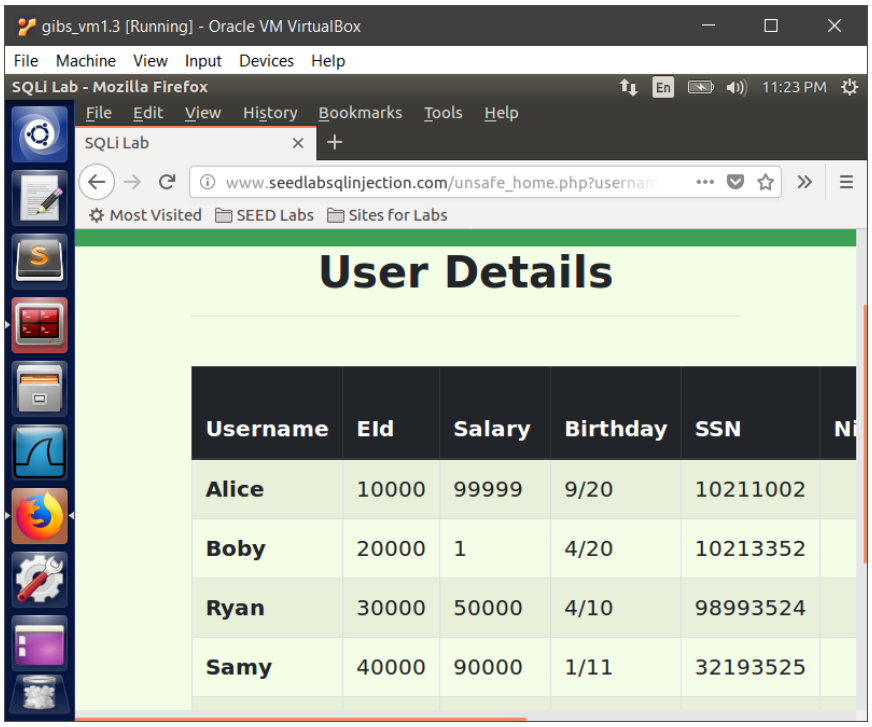


Task 3.2: Modify other people's salary

Here we used almost the same code as above, except instead of changing Alice's salary to 99999, we are changing Boby's salary to 1.

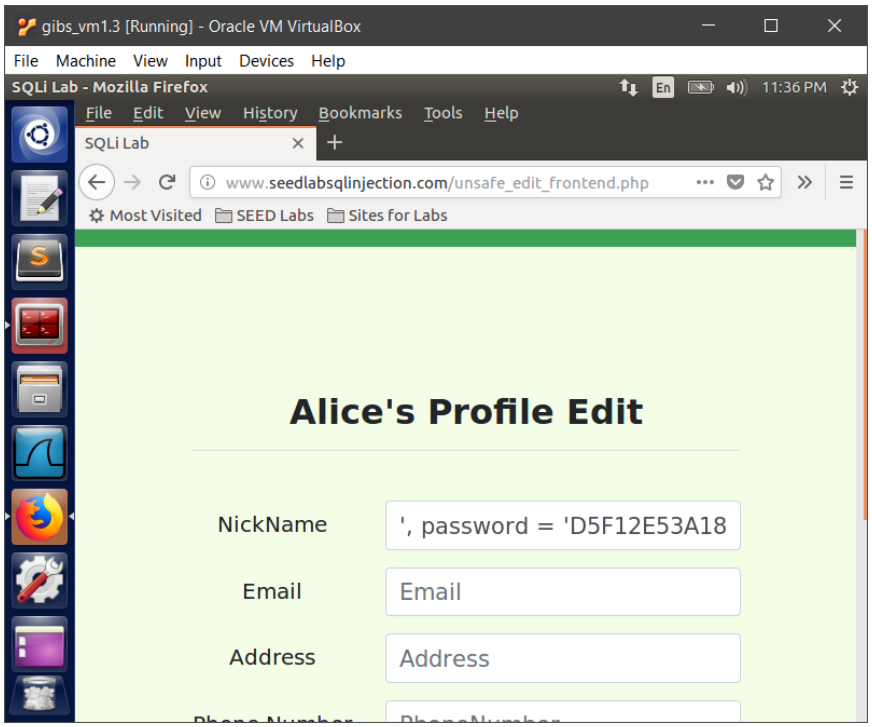


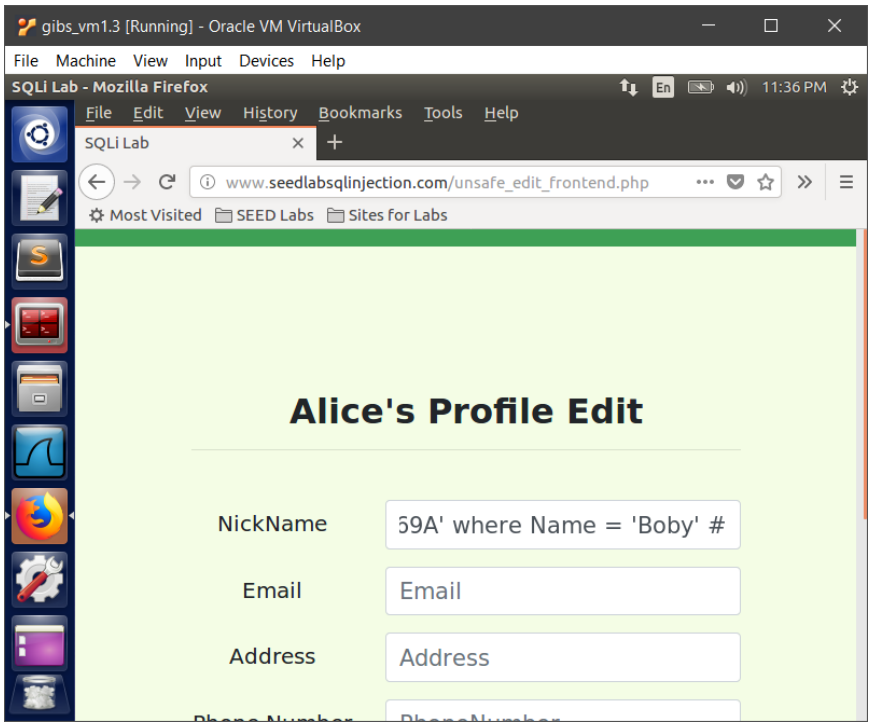


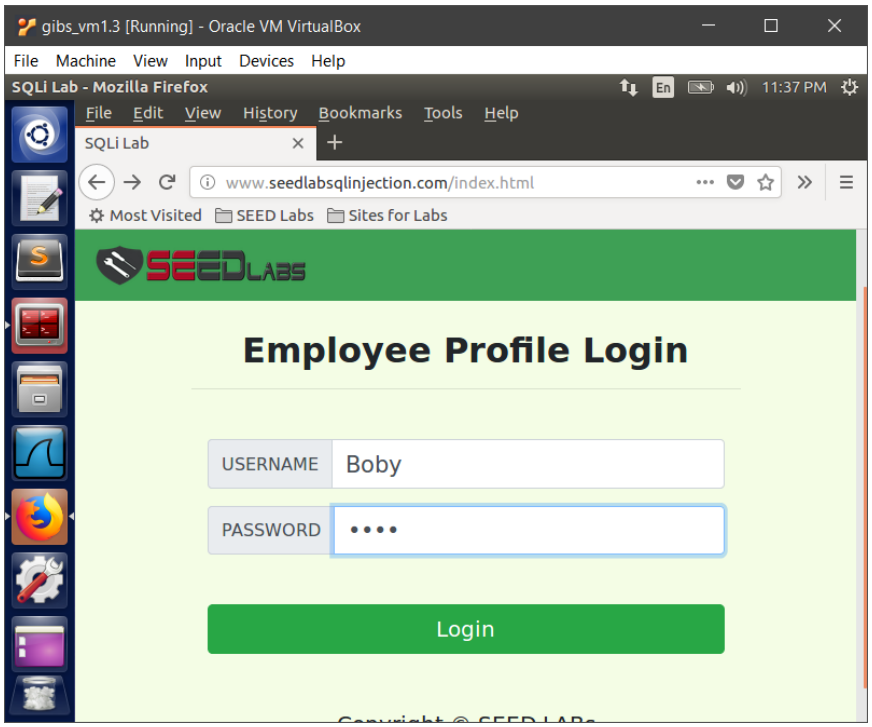


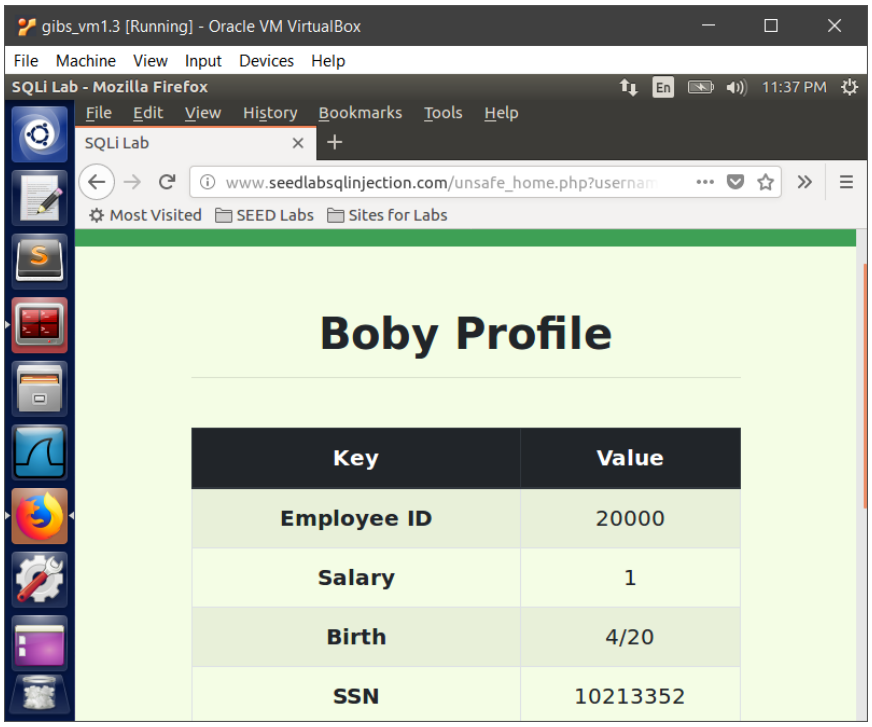
Task 3.3: Modify other people's password

Adding insult to injury, we are here changing Boby's password to 4321. We had to use an online SHA1 hasher so we could use the password we were putting in later. The hashed code for 4321 is D5F12E53A182C062B6BF30C1445153FAFF12269A, which is what we entered using a similar code to earlier except we used "password" instead of "salary."


gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SQLi Lab - Mozilla Firefox
11:36 PM
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
Most Visited SEED Labs Sites for Labs
Alice's Profile Edit
NickName
', password = 'D5F12E53A18
Email
Email
Address
Address

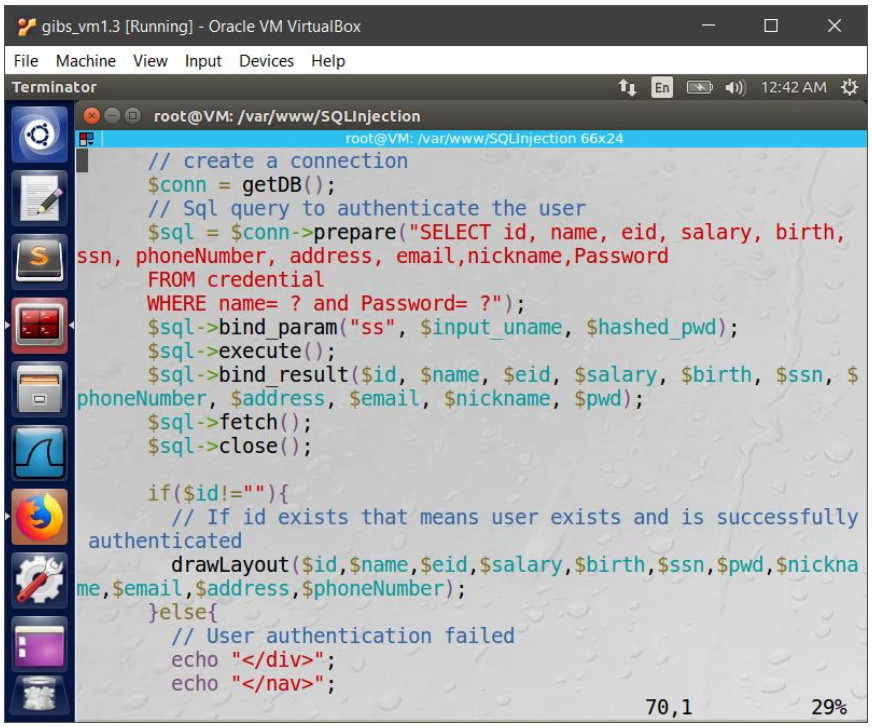

gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SQLi Lab - Mozilla Firefox
11:36 PM
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
Most Visited SEED Labs Sites for Labs
Alice's Profile Edit
NickName
59A' where Name = 'Boby' #
Email
Email
Address
Address

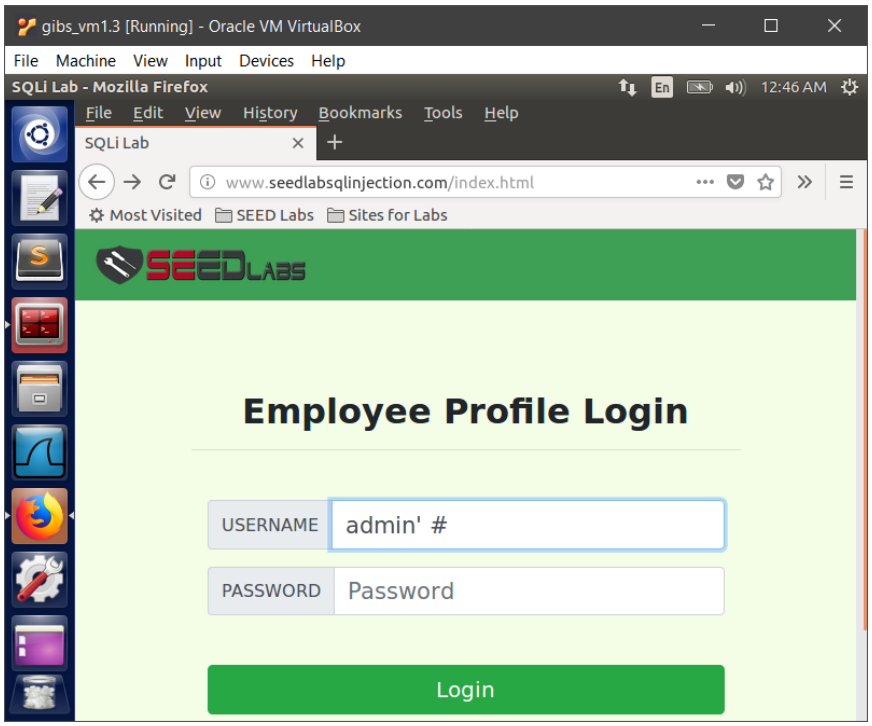


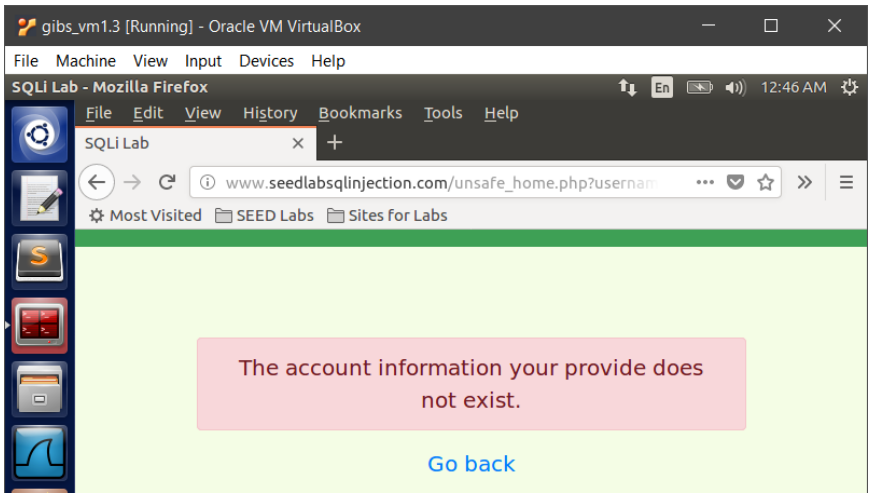


3.4 Task 4: Countermeasure – Prepared Statement

Here we went into the directory where the unsafe_home.php is stored, and made the necessary changes to it so we could not use an SQL Injection Attack on the website any longer. Below is a picture of the altered .php file, or at least the part of it we altered. Beneath that picture is two pictures demonstrating that with the fixed code, an SQL Injection Attack does not work.


gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
12:42 AM
root@VM: /var/www/SQLInjection
root@VM: /var/www/SQLInjection 66x24
// create a connection
$conn = getDB();
// Sql query to authenticate the user
$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= ? and Password= ?");
$sql->bind_param("ss", $input_uname, $hashed_pwd);
$sql->execute();
$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $nickname, $pwd);
$sql->fetch();
$sql->close();
if($id!=""){
// If id exists that means user exists and is successfully authenticated
drawLayout($id,$name,$eid,$salary,$birth,$ssn,$pwd,$nickname,$email,$address,$phoneNumber);
}else{
// User authentication failed
echo " ";
echo "</nav>";
70,1
29%


gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SQLi Lab - Mozilla Firefox
12:46 AM
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/index.html
Most Visited
SEED Labs
Sites for Labs
SEEDLABS
Employee Profile Login
USERNAME
admin' #
PASSWORD
Password
Login


gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SQLi Lab - Mozilla Firefox
12:46 AM
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/unsafe_home.php?usernam
Most Visited
SEED Labs
Sites for Labs
The account information your provide does not exist.
Go back