

---

# Sécurité Web 1

Olivier LASNE

2021-01-18

## Sécurité Web

Pour ce TP nous utiliserons la machine **OWASP Broken Web Apps** que vous avez déjà. Et l'iso "**From SLQI to Shell**" que vous pouvez télécharger ici :

[https://pentesterlab.com/exercises/from\\_sqli\\_to\\_shell/iso](https://pentesterlab.com/exercises/from_sqli_to_shell/iso)

### Rappel SQL

SQL est un langage de requêtes de base de données.

Vous pouvez vous connecter en SSH à votre VM OWASP Broken Web Apps en SSH. `root/owaspbwa`.

`ssh root@192.168.56.101` (remplacer avec l'IP de la machine)

On peut y lancer MySQL avec la commande suivante : `mysql -u root -powaspbwa`

Cela lance un shell MySQL. On obtient de l'aide avec la command `help` ou `\h`.

```
1 mysql> help
2
3 For information about MySQL products and services, visit:
4   http://www.mysql.com/
5 For developer information, including the MySQL Reference Manual, visit:
6   http://dev.mysql.com/
7 To buy MySQL Enterprise support, training, or other products, visit:
8   https://shop.mysql.com/
9
10 List of all MySQL commands:
11 Note that all text commands must be first on line and end with ';'
12 ?          (\?) Synonym for 'help'.
13 clear      (\c) Clear the current input statement.
14 connect    (\r) Reconnect to the server. Optional arguments are db and
      host.
15 delimiter (\d) Set statement delimiter.
16 edit       (\e) Edit command with $EDITOR.
17 ego        (\G) Send command to mysql server, display result vertically.
18 exit       (\q) Exit mysql. Same as quit.
19 go         (\g) Send command to mysql server.
20 help       (\h) Display this help.
21 nopager    (\n) Disable pager, print to stdout.
22 notee      (\t) Don't write into outfile.
23 pager      (\P) Set PAGER [to_pager]. Print the query results via PAGER.
24 print      (\p) Print current command.
25 prompt     (\R) Change your mysql prompt.
26 quit       (\q) Quit mysql.
27 rehash     (\#) Rebuild completion hash.
28 source     (\.) Execute an SQL script file. Takes a file name as an
      argument.
```

```
29 status      (\s) Get status information from the server.
30 system      (\!) Execute a system shell command.
31 tee         (\T) Set outfile [to_outfile]. Append everything into given
      outfile.
32 use         (\u) Use another database. Takes database name as argument.
33 charset     (\C) Switch to another charset. Might be needed for
      processing binlog with multi-byte charsets.
34 warnings    (\W) Show warnings after every statement.
35 nowarning   (\w) Don't show warnings after every statement.
36
37 For server side help, type 'help contents'
```

On peut voir les bases de données avec la commande `show databases;`

```
1 mysql> show databases;
2 +-----+
3 | Database |
4 +-----+
5 | information_schema |
6 | .svn |
7 | bricks |
8 | bwapp |
9 | citizens |
10 | cryptomg |
11 | dvwa |
12 | gallery2 |
13 | getboo |
14 | ghost |
15 | gtd-php |
16 | hex |
17 | isp |
18 | joomla |
19 | mutillidae |
20 | mysql |
21 | nowasp |
22 | orangehrm |
23 | personalblog |
24 | peruggia |
25 | phpbb |
26 | phpmyadmin |
27 | proxy |
28 | rentnet |
29 | sqlol |
30 | tikiwiki |
31 | vicnum |
32 | wackopicko |
33 | wavsepdb |
34 | webcal |
35 | webgoat_coins |
36 | wordpress |
37 | wraithlogin |
```

```
38 | yazd |
39 +-----+
40 34 rows in set (0.00 sec)
```

On sélectionne une base avec la commande **use** :

```
1 mysql> use peruggia;
2 Reading table information for completion of table and column names
3 You can turn off this feature to get a quicker startup with -A
4
5 Database changed
```

On peut ensuite lister les tables avec la commande **show tables** ; :

```
1 mysql> show tables;
2 +-----+
3 | Tables_in_peruggia |
4 +-----+
5 | picdata             |
6 | users               |
7 +-----+
8 2 rows in set (0.00 sec)
```

**!/ Les commandes **show** et **help** sont des commandes du SHELL MySQL. Il ne s'agit pas de requêtes SQL valides.**

On peut sélectionner l'ensemble des champs d'une table avec la requête **SELECT \* FROM nom\_de\_la\_table**.

Le caractère **\*** signifie *tout les champs*.

```
1 mysql> SELECT * FROM users;
2 +-----+-----+-----+
3 | ID | username | password |
4 +-----+-----+-----+
5 | 1 | admin    | 21232f297a57a5a743894a0e4a801fc3 |
6 | 2 | user     | ee11cbb19052e40b07aac0ca060c23ee |
7 +-----+-----+-----+
8 2 rows in set (0.00 sec)
```

On peut sélectionner un seul certains champs, en les listant séparés par des virgules.

```
1 mysql> SELECT ID, username FROM users;
2 +-----+-----+
3 | ID | username |
4 +-----+-----+
5 | 1 | admin    |
6 | 2 | user     |
7 +-----+-----+
8 2 rows in set (0.00 sec)
```

Note : il n'est pas nécessaire de mettre `SELECT` et `FROM` en majuscule. Néanmoins il s'agit de la convention prise dans la plupart des cas de façon à distinguer les champs des opérateurs.

On peut utiliser le mot-clé `WHERE` pour filtrer les éléments sélectionnés.

```
1 mysql> SELECT password FROM users WHERE username = 'admin';
2 +-----+
3 | password |
4 +-----+
5 | 21232f297a57a5a743894a0e4a801fc3 |
6 +-----+
7 1 row in set (0.00 sec)
```

**Exercice :** Sélectionner le nom de l'utilisateur avec l'ID 2.