Faculty of Business, IT, and Management
HACK2200 Hacking and Exploits
Lab 9: Web SQL Injection

Instructions

- This assignment should be completed individually.
- This assignment is designed for the purpose of education and training, but not for any illegal activities including hacking. Beware to only use these exploits on hosts that you have permission to hack.
- When a question asks for screenshots, your screenshots **must**:
    - Include the full window (the application window, or the terminal window, etc…),
    - have the PROMPT setup as per the instructions, including the date and time in the same format provided in the instructions. Screenshots without the prompt setup will receive zero credit,
    - be clearly readable,
    - include all the information required by the question, and
    - not include extra commands, failed attempts, and/or error messages.
- Failure to follow submission instructions will result in marks deduction. There will be marks deduction for including more than what is required in the instructions. Do not replace any screenshot that is not marked for replacement. These screenshots are to guide you only.
- The below instructions are guidelines, you are expected to troubleshoot any errors you run into.
- There will be mark deductions for including more than what is required in the instructions.
- Read and complete the lab instructions below and finish all the tasks. Replace screenshots that are labeled as sample-replace only, and answer the questions where highlighted.

Once completed, submit the Answer File only to the assignment dropbox.

Environment Setup

We will use a fresh copy of SEED Ubuntu 20.04 Virtual Machine available at https://seedsecuritylabs.org/

1- Create a new VM, by downloading Ubuntu 20.04 VM available under **Approach 1: Use a pre-built SEED VM** from the following link https://seedsecuritylabs.org/labsetup.html
2- Follow the lab manual setup instructions to install the SEED VM you downloaded in the previous step on VirtualBox. The lab manual setup is available here: https://github.com/seed-labs/seed-labs/blob/master/manuals/vm/seedvm-manual.md

SQL Tutorial

1. Visit the SQL tutorial available at http://www.w3schools.com/sql/ .
2. Follow the steps of the SQL tutorial.
3. Take the SQL Quiz located at https://www.w3schools.com/sql/sql_quiz.asp
4. Read the SQL Injection section available at https://www.w3schools.com/sql/sql_injection.asp
5. Provide the result of your SQL Quiz. The minimum accepted score is 75%. Take a screenshot of your quiz result and provide it under Screenshot#1 in the answer file.

Lab Tasks

1. Download the SQL Injection Attack Lab available here: https://seedsecuritylabs.org/Labs_20.04/Files/Web_SQL_Injection/Web_SQL_Injection.pdf
2. We will refer to the SQL Injection Attack Lab you downloaded in the previous step as the **Web_SQL_Injection** file.
3. Download the Labsetup file available at https://seedsecuritylabs.org/Labs_20.04/Web/Web_SQL_Injection/ , and browse to where you downloaded and unzipped the file.
4. Read and follow the instructions in the Web_SQL_Injection file to build the 2 containers required for this lab.
   a. Use the `docker-compose build` command to build the container.
   b. Use the `docker-compose up` command to start the container.

   You will need to issue these commands where you unzippzed your Labsetup files.

5. Use the Web_SQL_Injection file to aid you in carrying out the lab tasks below.
6. Build the containers.

```
$ PS1='[`date "+%D"`] yourname [`date "+%r"`] -[~]'
$ docker-compose build
```

```
seed@VM: ~
[02/08/21]seed@VM:~$ PS1='[`date "+%D"`] yourname [`date "+%r"`] -[~]'
[02/08/21] yourname [10:07:21 PM] -[~]cd Desktop/Labsetup/
[02/08/21] yourname [10:07:33 PM] -[~]docker-compose build
Building www
Step 1/5 : FROM handsonsecurity/seed-server:apache-php
apache-php: Pulling from handsonsecurity/seed-server
da7391352a9b: Downloading [>                                                  ]
 293.3kB/28.56MBlling fs layer
14428a6d4bcd: Downloading [==================>                                ]
14428a6d4bcd: Downloading [==================================================>]
da7391352a9b: Downloading [==>                                                ]
da7391352a9b: Downloading [=========>                                         ]
da7391352a9b: Downloading [=============>                                     ]
da7391352a9b: Downloading [====================>                              ]
da7391352a9b: Downloading [============================>                      ]
da7391352a9b: Downloading [================================>                  ]
 18.51MB/28.56MB
da7391352a9b: Downloading [=====================================>             ]
 20.27MB/28.56MB
da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
d801bb9d0b6c: Pull complete
Digest: sha256:fb3b6a03575af14b6a59ada1d7a272a61bc0f2d975d0776dba98eff0948de275
```

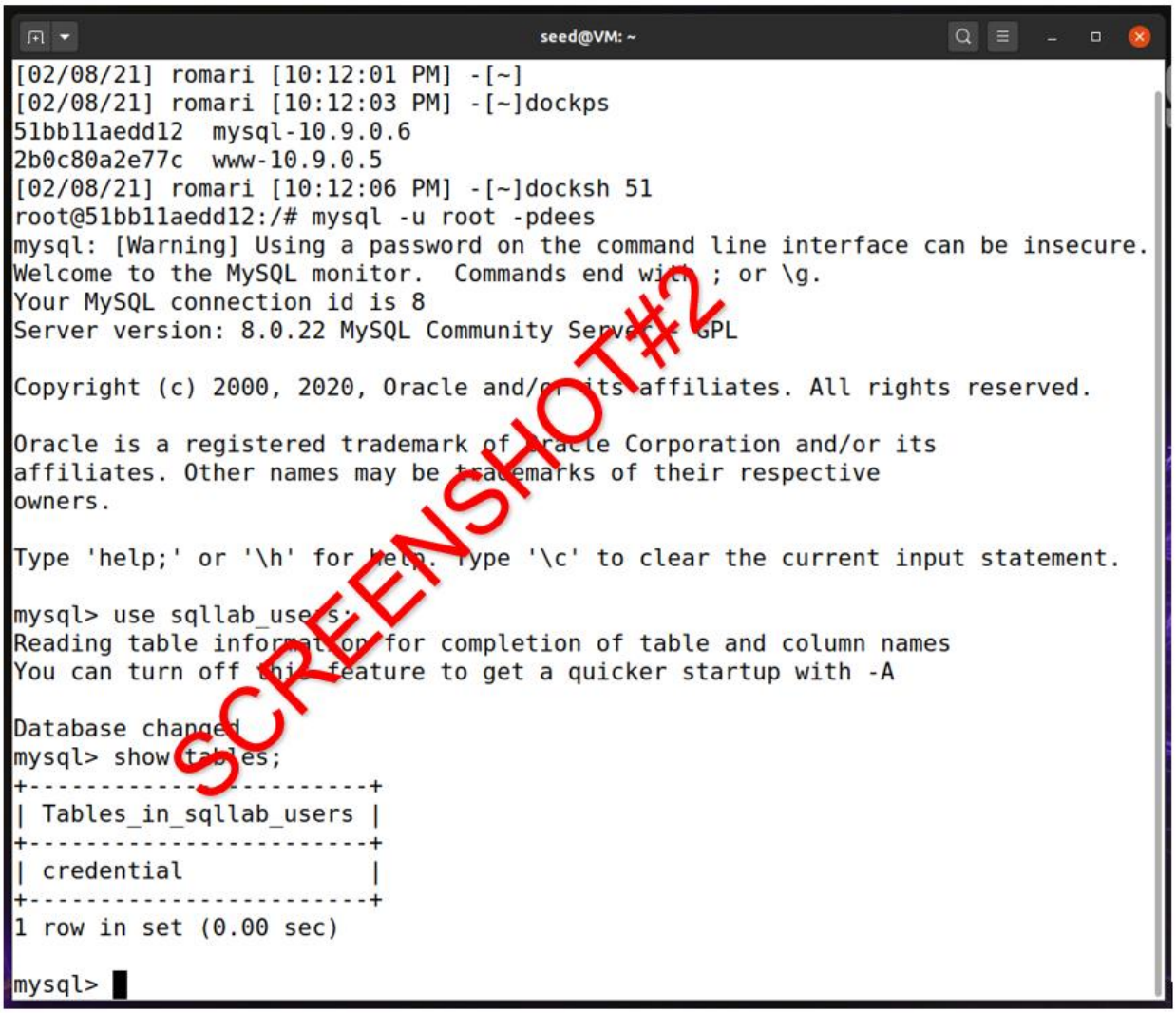

7. Start the containers:

```
$ docker-compose up
```

```
seed@VM: ~
mysql-10.9.0.6 | 2021-02-09 03:10:29+00:00 [Note] [Entrypoint]: MySQL init proce
ss done. Ready for start up.
mysql-10.9.0.6 |
mysql-10.9.0.6 | 2021-02-09T03:10:29.724545Z 0 [System] [MY-010116] [Server] /us
r/sbin/mysqld (mysqld 8.0.22) starting as process 1
mysql-10.9.0.6 | 2021-02-09T03:10:29.738536Z 1 [System] [MY-013576] [InnoDB] Inn
oDB initialization has started.
mysql-10.9.0.6 | 2021-02-09T03:10:30.191889Z 1 [System] [MY-013577] [InnoDB] Inn
oDB initialization has ended.
mysql-10.9.0.6 | 2021-02-09T03:10:30.336238Z 0 [System] [MY-011323] [Server] X P
lugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/my
sqld/mysqlx.sock
mysql-10.9.0.6 | 2021-02-09T03:10:30.444369Z 0 [Warning] [MY-010068] [Server] CA
 certificate ca.pem is self signed.
mysql-10.9.0.6 | 2021-02-09T03:10:30.444682Z 0 [System] [MY-013602] [Server] Cha
nnel mysql_main configured to support TLS. Encrypted connections are now support
ed for this channel.
mysql-10.9.0.6 | 2021-02-09T03:10:30.448805Z 0 [Warning] [MY-011810] [Server] In
secure configuration for --pid-file: Location '/var/run/mysqld' in the path is a
ccessible to all OS users. Consider choosing a different directory.
mysql-10.9.0.6 | 2021-02-09T03:10:30.500533Z 0 [System] [MY-010931] [Server] /us
r/sbin/mysqld: ready for connections. Version: '8.0.22'  socket: '/var/run/mysql
d/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
```

Task 1: Get Familiar with SQL Statements

1. Start a new terminal, and follow the commands in Task 1 in the Web_SQL_Injection file to get familiar with SQL Statements. Note that all the commands are run inside the container.
2. Take a screenshot of the terminal, and place it under Screenshot#2 in the answer file.



Task 2: SQL Injection Attack on SELECT Statement

Study how authentication is implemented in the web application, specifically go through the `unsafe_home.php` file. Then, carry the following tasks.

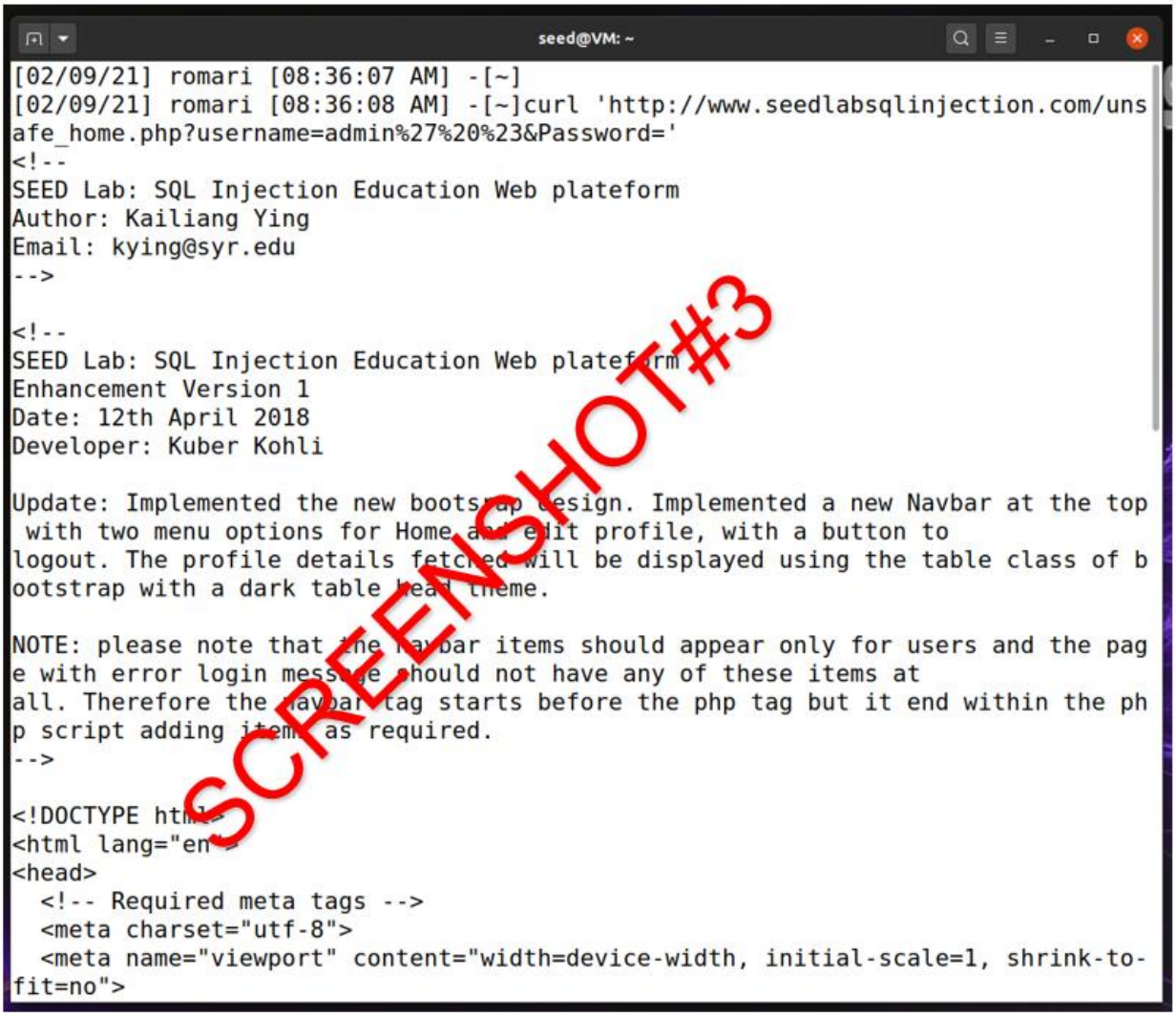
Task 2.1: SQL Injection Attack from webpage

Carry out task 2.1 as described in the Web_SQL_Injection file, and answer the following question:

Question#1: Type the SQL code you used to login as an admin, without knowing the admin's password.

Task 2.2: SQL Injection Attack from the command line.

Carry out task 2.2 as described in the Web_SQL_Injection file. Take a screenshot of your terminal issuing the curl command, and place it under Screenshot#3 in the answer file.



Task 3: SQL Injection on UPDATE Statements

Study the PHP code implemented in `unsafe_edit_backend.php` file, and perform the following tasks.

Task 3.1: Modify your own salary.

Carry out task 3.1 as described in the Web_SQL_Injection file, and answer the following question.

Question#2: Type the SQL code you used to update your own salary, and in which field did you type that code (i.e., username, password, etc..)?

Task 3.2: Modify other people' salary.

Carry out task 3.2 as described in the Web_SQL_Injection file, and answer the following question.

Question#3: Type the SQL code you used to update Boby's salary, and the field where you typed that code.

Task 3.3: Modify other people's passwords.

Carry out task 3.3 as described in the Web_SQL_Injection file, and answer the following question.

Question#4: Type the SQL code you used to change Boby's password, and the field that you used to type that code.

Question#5: What did you change Boby's password to? (5 Marks)