

Lab 5: SQL Injection Attack

Total Points: 30

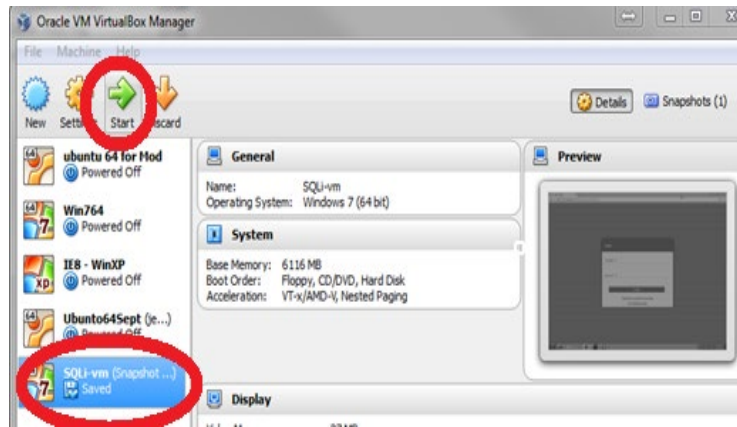
Objective:

The objective of the project is to provide hands-on experience on web security as well as SQL injection attack. We also want to show how such attacks are executed by malicious parties (e.g., hackers) in real settings. For this purpose, we prepared a virtual machine that has a web application that is connected to a database. This provides a safe environment to try and experiment with such attacks. Recall to experiment with these attacks only in such safe and isolated environments.

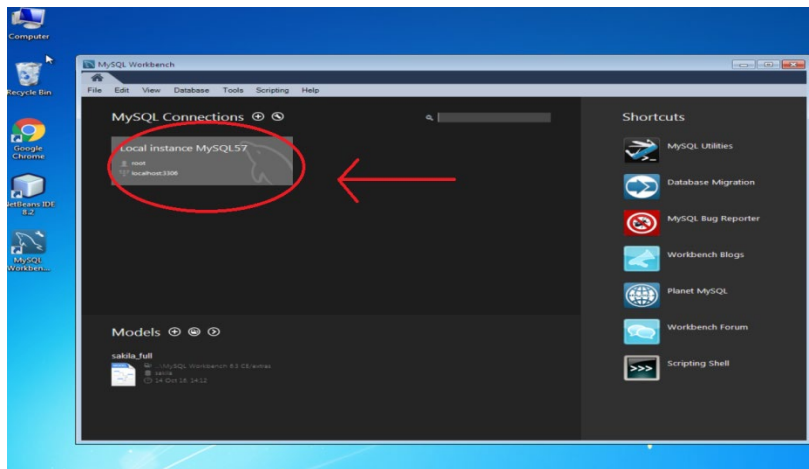
Tasks:

- A. Download and install the **VirtualBox** software.
 - If you are a Windows or Mac user with Intel/AMD processor, just download the latest (relevant) software package from here:
<https://www.virtualbox.org/wiki/Downloads>.
 - If you are using a Mac M1/M2 (containing ARM processor), download the **Developer Preview for MacOS / Arm64 (M1/M2) hosts** package of any release from here: https://www.virtualbox.org/wiki/Download_Old_Builds_7_0. I downloaded the VirtualBox 7.0.8 version for my Mac M1 laptop.
 - **For Mac M1/M2 user:** Use this guideline for the installation if you are confused:
<https://www.makeuseof.com/how-to-install-virtualbox-apple-silicon-mac/>.
- B. Download the **VM instance** that you will be using for this project from this link:
https://drive.google.com/file/d/1XSGzNI4O_y5oOHRnUiQtP8qQS4_kOSCb/view?usp=sharing. The size of this file is > 6.5 GB. So, if possible, use the campus network to download it quickly.
 - If you face downloading this OVA file, or if the file is corrupted, download a zipped version from here:
https://drive.google.com/file/d/1wyrxsbIUHMcqYC1ygFWtD1_Zm-jp7Dt4/view?usp=sharing.
- C. **Setting up the VM instance:**
 - In VirtualBox, go to **File** and choose **Import appliance**

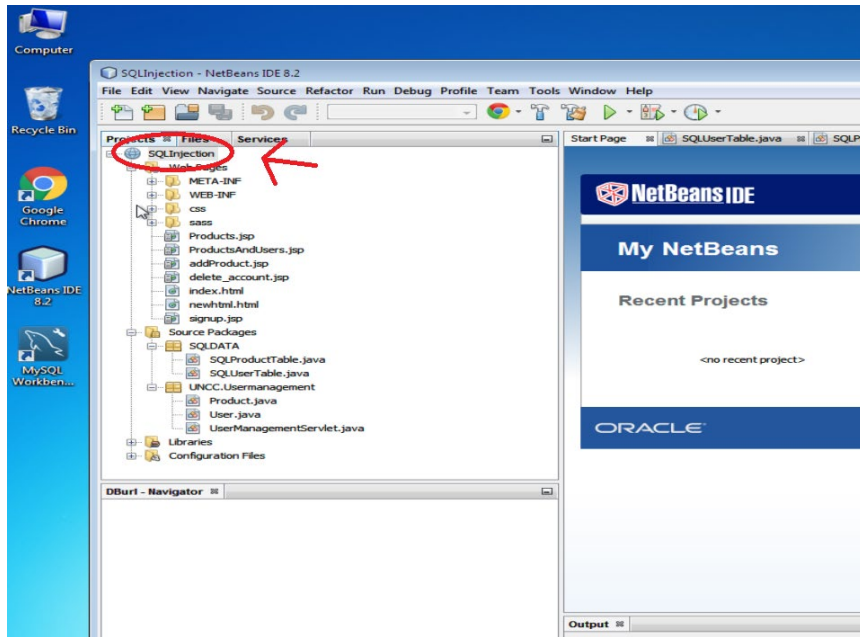
- Navigate to the ***SQLi-fvm32.ova*** file you downloaded earlier and perform the necessary tasks (e.g., clicking *next*) to successfully import the instance.
- Once loaded, select the virtual machine instance, and click **start**:



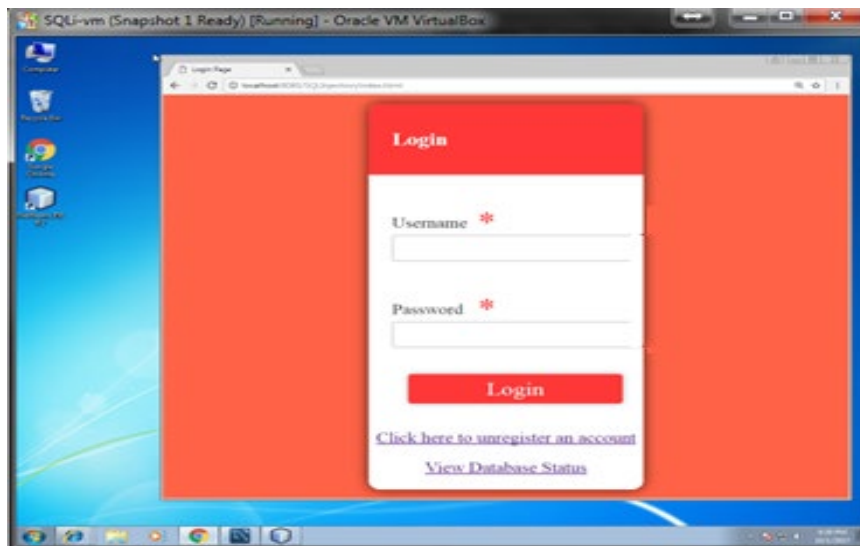
- D. After the Virtual machine finishes loading, you should be able to see the desktop. There is an icon for **MySQL Workbench** on the desktop, double click on it to start the application. Once it starts, double click on the '**local instance**' and enter the password "**root**" if asked.



- E. The desktop has another icon for **NetBeans IDE**, you can double click on it to start the application. Then, right click on the **SQLInjection** project and choose "run".



- F. Once you run the project, you should be able to see the new virtual environment and the **login screen** of the web application.

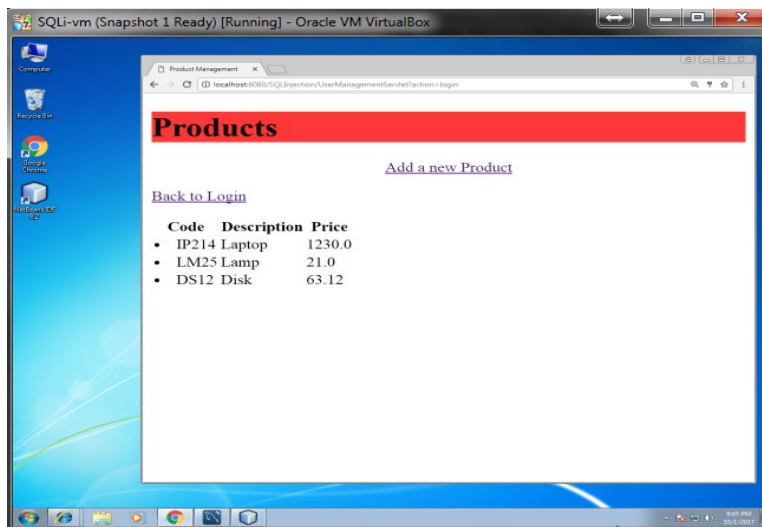


- You should be able to login using the following **credentials**:

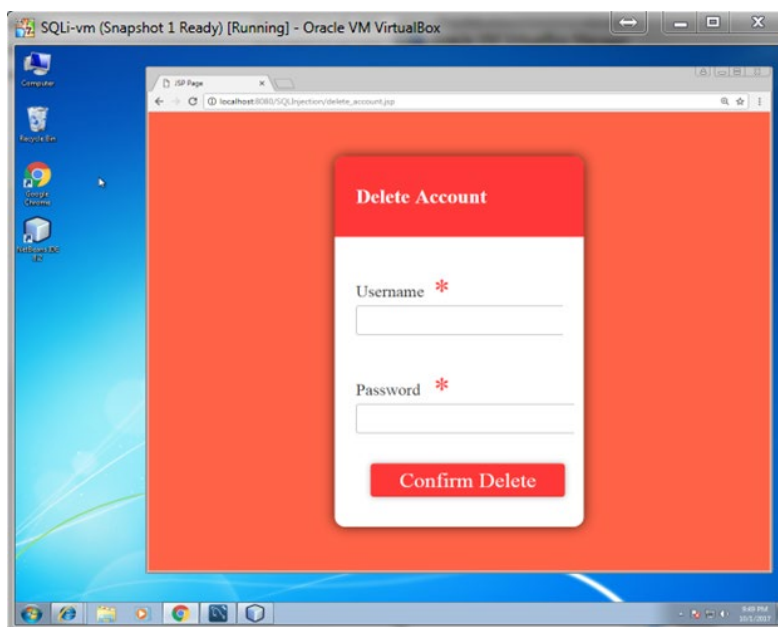
Username: JohnConnor

Password: skynet

When you successfully login, you will be able to **view a list of products**, and you can add new products through the link ***add a new product***.



You can also delete an account from the database using ***click here to unregister***, where you will be asked for your username and a password:



- G. [important] The ***view database status*** link at the main screen (login page) will show you the SQL tables in the database and their structures. Note that understanding tables' structures is vital to launch SQL attacks.
- H. This website shows many variations to SQL commands:
<https://www.w3schools.com/sql/>. You can explore some examples there.

Hacking the website:

- (1) **Bypass the login screen:** Without using a valid username and password, try to hack into the website login page using the appropriate script or command injection. Try the following script in the **Username** box and keep the **Password** box **empty**.

Username: 'or'7'='7';#

Could you bypass the login hurdle and access the **Products** page? Please take a screenshot of the login page when you entered the script and a screenshot of the **Products** page after bypassing the login page. Include these two screenshots in your submission. [4 points]

Now, try to replace the script in the **Username** box with 'or'7'='8';# and try accessing the **Products** page. Did you find any difference here? Again, please take a screenshot of the login page when you entered the script and a screenshot of the **Products** page after bypassing the login page. Include these two screenshots in your submission. [4 points]

In your own words, explain the reasons behind the difference caused by the 2nd script. [2 points]

- (2) **Open a backdoor:** Once hackers are in, they immediately open backdoors (a way that can be used later to log into the system without hacking it again, such as creating a new account). Let's create a new user account and keep it as a backdoor for the future.

From the **Login** page, click on the link **Click here to unregister an account**, and it will bring you to the **Delete Account** page. In this page, write the following script in the **Username** box and keep the **Password** box empty as usual.

Username: 'or'7'='8'; INSERT INTO users(id, firstName, lastName, email, password) VALUES(9, "your_first_name", "your_last_name", "your_email_address", "your_password"); #

Note that you should replace the fields **your_first_name**, **your_last_name**, **your_email_address**, and **your_password** with your actual first name, last name, email address and a custom password. Take a screenshot of the **Delete Account** page with the given script written in the **Username** box and attach it into your submission. [4 points]

Click on the **Confirm Delete** button and use the view **database status link** to view the **Products** page. Take a screenshot of the **Products** page and highlight the newly added account information there. [2 points]

Now, using the similar script, add another user with the first name **John** and last name **Smith**. Note that you cannot keep the *id*, *email*, and *password* fields empty. So, choose anything that comes to your mind. After adding the user John Smith, your next task is to remove the user with the **id = 9**.

To do that, from the **Login** page, click on the link ***Click here to unregister an account***, and it will bring you to the **Delete Account** page. In this page, write the following script in the *Username* box and keep the *Password* box empty.

Username: 'or'7'='8'; DELETE FROM users WHERE id = 9; #

Take a screenshot of the *Delete Account* page with the given script written in the *Username* box. Take a screenshot of the **Products** page also after following the link *view database status*. Attach both screenshots into your submission. [4 points]

- (3) Take over all customer accounts in the website by setting all of their passwords to '123':** Once a backdoor is created, now you need to attack other customers and hijack their accounts, set all of their passwords to a single value so that you can log into their accounts whenever you please.

To do that, write the following script in the *Username* box in the **Delete Account** page and keep the *Password* box empty:

Username: 'or'7'='8'; UPDATE users SET password = '12345'; #

Take a screenshot of the *Delete Account* page with the given script written in the *Username* box. Take a screenshot of the **Products** page also after following the link *view database status*. Attach both screenshots into your submission. [5 points]

Note that the password field for all the user accounts now should be updated into **'12345'**. Try logging in using a random username from the *Products* page with the password field set at **'12345'**.

- (4) Use XSS attack to run script on a user (victim) if he goes to view products page.**

An XSS attack is like planting a trap, you plant it, and then you wait for a victim to step on it. So, if you add a new product that has a XSS in its name, when another user logs in and views all products, he will be caught by your trap. In other words, your script in the XSS will run on his machine. In this task, we will try to plant XSS in the product list by adding a new product that has a script in its name.

First, try logging in as the user John Connor. Now, if you try the password **skynet** for the username **JohnConnor**, you will see an error. That's because you already updated the password for all usernames into '**12345**' in the previous task. So, you should try logging in with the following credentials:

Username: JohnConnor
Password: 12345

Once you are logged in, try to add a new product following the link **Add a new product**. Use the following information to add the product:

Product Name: Table <script>alert("Ha ha! This is a trap!")</script>
Product Price: 200

Take a screenshot of this page and click on the **Add Product** button to add the product. Now, try to login again as Alan Turing with the following credentials:

Username: AlanTuring
Password: 12345

Do you see a pop-up message saying "**Ha ha! This is a trap!**"? If yes, take a screenshot of the page showing the whole message and include it into your submission along with the previous screenshot. [5 points]