CSC427 SQL Injections and sqlmap Tutorial

You will be using two VM's in this tutorial: Kali and Metasploitable.
Kali: /virtual/csc427/kali
Metasploitable VM: /virtual/csc427/metasploitable2
Copy these into your own virtual folders

Instructions for Setting up the Metasploitable VM:

Make sure the VM is in bridged mode.
Start the Metasploitable 2 VM and login with `username: msfadmin, password: msfadmin`

Run `sudo nano /var/www/mutillidae/config.inc` and update the config file to have the dbname "owasp10". Press Ctrl+X, y, then Enter to save the file.

Restart the VM.

Run `ifconfig` to determine the IP address of the VM.

Instructions for the Kali Linux VM:

Use the following login information: Login: root
Password: toor

Visit http://[IP address of Metasploitable VM]/mutillidae/index.php in a web browser to access the Mutillidae Web application.

Open the command line and use `sqlmap [options]` to run sqlmap on it.

Tutorial:

The purpose of this tutorial is to perform SQL injections on the Mutillidae web application.

Answer the following questions in answers.txt.

Hint 1: Use https://github.com/sqlmapproject/sqlmap/wiki/Usage as a reference for command options.

Hint 2: Use the --batch option to use default behavior rather than asking for user input.

1. List at least 2 urls in Mutillidae that may be vulnerable to sql injection.

2. On the Mutillidae VM find all the tables inside the **owasp10** database.
   a. Submit your command and **the list of tables**.
   b. Which parameters did sqlmap inject and which type of technique (Union, error etc) did sqlmap use?

   HINT:
   - Don't use --dump (one of the tables is very large and this will take a long time).
   - Look at sqlmap documentation to determine how you can tell sqlmap to only target the owasp database
   - Use **--forms** so that sqlmap injects into form data and not HTTP headers like User-Agent

3. What is the database version running on the application?

4. How would you discover all the username and passwords? Submit your command and the **csv** sqlmap generates of the database table.
   HINT: Don't dump the entire database, instead use an sqlmap command to only dump a single table!

5. Are you able to use the --sql-shell to **insert** data such as a new user into the database? Why or why not?

6. View the logs page in Mutillidae. How many requests are in the log? Why are there so many requests? This information is visible on the site, no need for sqlmap.

7. What are the privileges of the current database user? Submit at least three privileges and the command(s) you ran, in the following format:

   > Command 1
   > Command 2
   > Command …
   [Current user name] privileges: X, Y, Z

8. Create a file "payload.txt" with text of your choice (I recommend writing "*Hacker voice: I'm in" in the file) in the /tmp directory of the Metasploitable VM. Use sqlmap on Kali Linux to read the contents of the file. Submit the commands you ran in answers.txt, and a screenshot of sqlmap output showing the read was successful in screenshot.png or screenshot.jpg.
   NOTE: If you restart the VM the payload.txt you created will disappear and you will need to re-create payload.txt (in /tmp)

Zip the folder containing answers.txt and Question 8's screenshot. Submit it as sql.zip.

Congratulations! You now know how to use sqlmap to perform penetration testing and devastating SQL injections.

Work from home:

Install the Metasploitable VM: https://sourceforge.net/projects/metasploitable/. Set it up as described in **Instructions for Setting up the Metasploitable VM** at the top of the document.

You can either install sqlmap on your own computer or Kali Linux on your computer:

Install sqlmap to your own computer (from the buttons on the top-right of http://sqlmap.org/), note that you need Python to run it. OR Install the Kali Linux VM on your machine, which comes pre-installed with sqlmap. Then, Set it up as described in **Instructions for Setting up the Kali Linux VM** at the top of the document.