## Lab 10: Automating SQL injection using SQLmap

### Scenario

You are hired as a certified cybersecurity practitioner to test some database vulnerabilities. You are required to simulate SQL injection attacks to retrieve hidden data. You can use manual methods to find web application vulnerabilities in SQL injection (SQLi).

### Solution

You must use some techniques to find SQLi vulnerabilities in online applications. You will need to inject some SQL queries with the help of the commenting functionality in SQL. You will use different payloads for this; hence, if one payload does not work, you can try another.

|  |
| --- |
| 1. The following full URL is used as an example for the lab: [**http://testphp.vulnweb.com/listproducts.php?category=2**](http://testphp.vulnweb.com/listproducts.php?category=2)    2. Right-click on the page and select **Inspect (Q)**.    3. A new window appears at the bottom. Select **Console**. On the console tab, write **document.cookie** and press enter. It will show the cookie value; copy this value.    4. On ParrotOS, open the terminal and write the command **sudo su** to operate in root mode.  5. Use the command pattern **sqlmap -u "URL" --cookie="value" –dbs to launch sqlmap attack.** Copy the URL from **Step#1** and the cookie value from **Step#3** and paste it here.    In this step, **-u** specifies the target URL. The URL contains a parameter **cat=2** that is being tested for SQL injection. The **--cookie** option sends a custom HTTP cookie to simulate an authenticated session. Here, the cookie value is **login=test%2Ftest**, which may bypass authentication checks. The **--dbs** option tells SQLMap to enumerate and list all available **databases** on the target server.    6. If it asks to keep testing, enter **y** and press **Enter.**    7. Hence, it provides the databases and their information.    8. Use the command **sqlmap -u <URL> -D db\_name --tables** command to scan the tables in the database.    9. It shows how many tables are present in the database and their names.    10. Use the command **sqlmap -u <URL> --cookie=<value> -D <db name> -T**  **<table name> --dump** command to scan the tables in the database.    11. The result shows that there is one entry in this particular table that shows all the details, such as name, password, email, phone number, etc.    12. You can also use the command **sqlmap -u <URL> --cookie=<value> --os-shell** to retrieve the hostnames of the machine on which the target web application is running.    13. This concludes the end of SQLmap. |