Chapter 6

A Practical Guide to Database Hacking with SQL Map

Welcome to the practical world of database hacking. In this chapter, we'll move beyond theory and dive into using SQL injection to hack and dump a live website's database using SQL Map. If you missed the previous one, where we introduced SQL Map basics, make sure to check it out. Now, let's begin.

WHAT IS A DATABASE

A database is an organized system for storing and managing data. Think of it as a notebook where different types of information are stored in sections. For example, a school's database named "ADV School" could have three tables:

1. **Students Table**: Contains data like student names, classes, and roll numbers.
2. **Teachers Table**: Holds teacher-related information.

3. **Admin Table**: Stores admin credentials and settings.

Each table is like a section in the notebook, holding relevant data in an organized way. Our goal is to access and extract (dump) this data.

STEP 1: IDENTIFYING SQL INJECTION VULNERABILITIES

Here are the steps you need to follow:

Choose the Target Website

Select the URL of the website you want to test for vulnerabilities.

Write the Initial Command

Open Notepad and write the following command:

```
sqlmap -u "target_website_url" --crawl 2 --random-agent
--threads 6
```

- *-u*: Specifies the target URL.
- *--crawl 2*: Tells SQL Map to scan pages linked to the main website. The number (1-5) determines the depth; higher numbers scan deeper but take more time.
- *--random-agent*: Sends requests using random user-agents to avoid detection.
- *--threads 6*: Increases scanning speed (but keep it below 6 to avoid triggering firewalls).

Run the Command

Copy the command into the terminal of Kali Linux and hit Enter.

Responding to Prompts

- SQL Map may ask if you want to test for redirections—press n (no).
- It will ask if you want to normalize crawling—press y (yes).
- You might be asked to save detected links—press n (no).

Analyze the Output

SQL Map will provide a list of links that could be vulnerable to SQL injection. Review these links carefully before proceeding.

STEP 2: FINDING THE DATABASE NAME

To find the database name, you need to follow the following steps:

Modify the Command

Use one of the vulnerable links found earlier and update the command to:

```
sqlmap -u "vulnerable_link" --dbs
```

--dbs: This tells SQL Map to retrieve the names of all databases on the website.

Run the Command and Answer Prompts

- If asked about cookies, press y (yes).
- SQL Map may detect the database type (e.g., MySQL) and ask if it should focus only on it—press y (yes).

Review Database Names

SQL Map will display the names of detected databases. Ignore default databases (which often contain no useful data) and focus on the target database.

STEP 3: DUMPING THE DATABASE

Here are the steps to dump the database:

Write the Dump Command

Update the command to include the name of the target database:

```
sqlmap -u "vulnerable_link" -D database_name --dump
```

- *-D*: Specifies the database name.
- *--dump*: Dumps all the data from the specified database.

Specify the Output Location

If you want to save the dumped data in a specific folder, add this to your command:
--output-dir="desired_location"

Automate Responses

Add the *--batch* command to skip SQL Map's questions and let it choose default answers:

```
sqlmap -u "vulnerable_link" -D database_name --dump
--batch
```

Run the Command

Copy the full command into your terminal and hit Enter. SQL Map will begin dumping the data. Depending on the size of the database, this may take some time.

STEP 4: ACCESSING THE DUMPED DATA

Follow these simple steps to access the dumped data:

Open the Dump File

SQL Map saves the dumped data in a specific folder. By default, the location is displayed in the terminal after the dump is complete.

Move the File

To access the file more easily, you can move it to a preferred folder. For example, if you want to save it to your desktop, copy the desktop folder path and add this command to the previous one:
--output-dir="/path_to_desktop"

Analyze the data

Open the dumped file in Firefox or any text editor. You'll find sensitive information like usernames, passwords, and other database entries.

IMPORTANT TIPS

- **Batch Mode**: The `--batch` command is useful for automating responses during repetitive scans.
- **Database Names**: Always verify the database name before dumping to avoid unnecessary scanning.
- **Ethical Use**: Hacking without permission is illegal. Use these techniques responsibly, such as in penetration testing with proper authorization.

WRAPPING UP

In this chapter, we successfully identified SQL injection vulnerabilities, extracted database names, and dumped sensitive data using SQL Map. This process has shown you how attackers can exploit weak security measures to access and steal critical information from websites. However, as an ethical hacker, your responsibility is to use this knowledge for defensive purposes—to find and fix these vulnerabilities before malicious actors can exploit them. Always remember, hacking without proper authorization is illegal and unethical.

By ScripterJee