# Vulnerability Assessment Using sqlmap

***Abstract*—sqlmap is an open-source penetration testing tool that automates the detection and exploitation of SQL injection vulnerabilities in web applications. sqlmap is widely used in vulnerability assessments to help identify and mitigate risks associated with SQL injection attacks due to its powerful automation capabilities.**

## I. Introduction

sqlmap is a powerful open-source tool specifically designed for detecting SQL injection vulnerabilities in web applications. By automating the process of identifying and exploiting SQL injection points, sqlmap has become valuable for cybersecurity professionals. It supports a wide range of databases, injection techniques, and post-exploitation activities, making it an essential tool for vulnerability assessments.

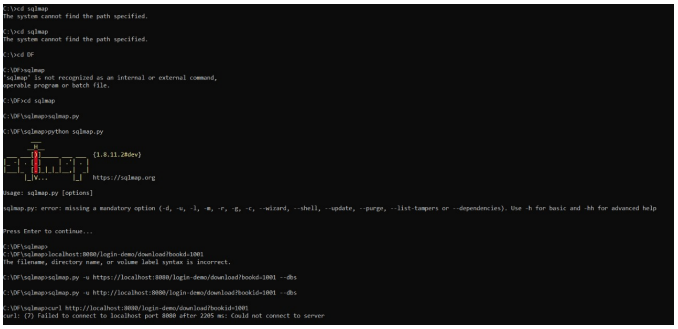

Fig. 1. Overview of sqlmap and its applications in vulnerability assessment

### A. Key Features

- Supports various database management systems, including MySQL, PostgreSQL, Oracle, and more.
- Automates detection and exploitation of SQL injection vulnerabilities.
- Supports blind, time-based, and error-based SQL injection techniques.
- Enables data retrieval from compromised databases.

## II. Installation

sqlmap can be installed on Linux, macOS, and Windows systems.

### A. On Linux (Ubuntu/Kali)

To install sqlmap on Linux, use:

```
sudo apt update
sudo apt install sqlmap -y
```

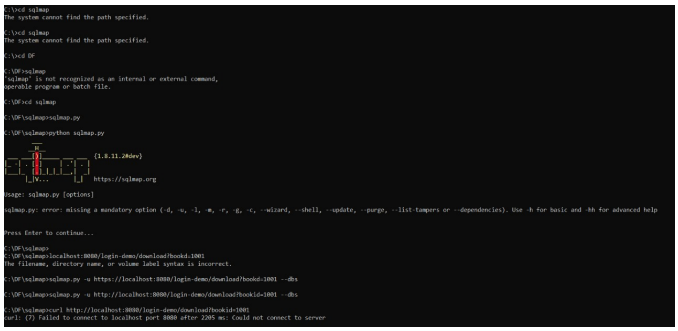

Fig. 2. Installing sqlmap on Linux (Ubuntu/Kali)

### B. On macOS (using Homebrew)

Install sqlmap on macOS with:

```
brew install sqlmap
```

### C. On Windows

- Download sqlmap from the official repository on GitHub.
- Execute sqlmap commands with Python.

## III. Using sqlmap: Commands and Examples

sqlmap provides various command-line options to customize scans. Here are some essential commands.

### A. Basic Syntax

The basic syntax for sqlmap is:

```
python sqlmap.py -u <target URL>
```

This command initiates a scan for SQL injection vulnerabilities on the specified URL.

### B. Examples and Key Options

1) **Basic Scan**: Scans a target URL for SQL injection points.

   ```
   python sqlmap.py -u http://testphp.vulnweb.com
   ```

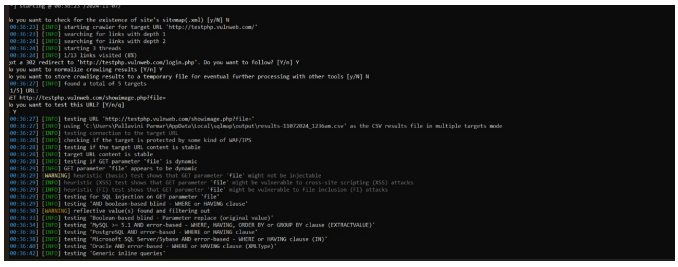

Fig. 3. Basic scan command example

2) **Crawl and Scan**: Crawls the site up to 2 levels deep and uses 3 threads.

```
python sqlmap.py -u http://testphp.vulnweb.com/ --crawl=2 --threads=3
```

Fig. 4. Using –crawl and –threads options with sqlmap

3) **Batch Mode**: Use –batch for non-interactive scanning.

```
python sqlmap.py -u http://testphp.v
```

4) **Database Table Extraction**: Lists all tables from a vulnerable database.

```
python sqlmap.py -u http://testphp.v
```

Fig. 5. Extracting tables from a vulnerable database

5) **Dump Data**: Extracts data from the database once tables are identified.

```
python sqlmap.py -u http://testphp.vulnweb.com --dump
```

## IV. Managing Output Files

sqlmap can save results in formats like TXT and XML for documentation and analysis.

### A. Saving Output to a File

To save scan results, use:

```
C:\Users\Pallavini_Parmar\AppData\Local\sqlmap\output\results-11072024_1240am.csv
```

## V. Generating a Vulnerability Assessment Report

A sqlmap vulnerability assessment report should include:

### A. Executive Summary

Overview of assessment results, highlighting key vulnerabilities and risks.

### B. Scan Details

- **Target URL**: Target website details.
- **Scan Date**: Date and time of the scan.
- **Tool and Version**: sqlmap version.
- **Command Options**: Options used in the scan.

### C. Vulnerability Findings

- List of detected vulnerabilities, including injection points.
- Severity ratings and descriptions.
- Recommended actions.

### D. Conclusion and Recommendations

Summarize findings with actionable recommendations for mitigation.

Fig. 6. Using –crawl and –threads options with sqlmap

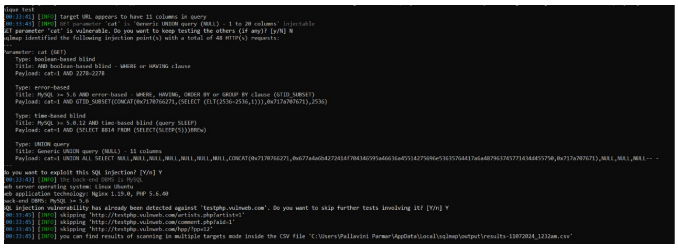

Fig. 7. Using –crawl and –threads options with sqlmap

## VI. Conclusion

sqlmap is a robust and efficient tool for identifying SQL injection vulnerabilities in web applications. It provides extensive automation options, ideal for quick assessments. However, responsible and ethical use is crucial.