SQL Injection (SQLi) - CEH Module 15 Cheat Sheet

# 1. Introduction

SQL Injection (SQLi) is a code injection technique where attackers insert malicious SQL statements into input fields to manipulate the backend database. It can be used to bypass authentication, extract data, modify or delete records, and even take control of the database server.

# 2. Types of SQL Injection

## Error-based SQLi

Exploits database error messages to extract information. Example payload: ' OR 1=1--

## Union-based SQLi

Uses UNION operator to combine results of multiple queries. Example payload: ' UNION SELECT null, username, password FROM users--

## Boolean-based Blind SQLi

Relies on sending queries that return true/false and observing behavior. Example payload: ' AND 1=1-- (true), ' AND 1=2-- (false)

## Time-based Blind SQLi

Uses database sleep/delay functions to infer information. Example payload: ' OR IF(1=1, SLEEP(5), 0)--

## Out-of-Band SQLi

Uses external channels (DNS/HTTP requests) to exfiltrate data. Example: xp\_dirtree in MSSQL to trigger DNS lookup.

# 3. SQLMap - Automated SQLi Tool

sqlmap is one of the most powerful open-source tools for detecting and exploiting SQL Injection.

-u : Target URL (example: -u "http://test.com/page.php?id=1")

--dbs : Enumerate available databases

--tables : List tables in a specific database (requires -D <db>)

--columns : List columns in a table (requires -D <db> -T <table>)

--dump : Extract data from database

--batch : Run in non-interactive mode (auto-choose default answers)

--risk : Risk level of tests (1-3, default 1)

--level : Level of tests (1-5, default 1)

--os-shell : Get an interactive operating system shell

--file-read : Read files from the server

--file-write : Write files to the server

--threads : Number of concurrent threads (speed up)

# 4. Other SQLi Tools

• Havij – Automated SQL Injection tool with GUI  
• jSQL Injection – Java-based SQLi tool  
• SQLNinja – Exploits SQLi on Microsoft SQL Server  
• NoSQLMap – Exploits NoSQL databases (MongoDB, CouchDB)

# 5. Detection & Prevention

• Use parameterized queries (Prepared Statements)  
• Apply stored procedures securely  
• Employ Web Application Firewalls (WAF)  
• Implement least-privilege database accounts  
• Disable verbose error messages  
• Regularly patch and update DBMS & web apps