# **Module 15: SQL Injection - Practical Guide**

This guide focuses on the technical tools and step-by-step procedures used to detect, verify, and exploit SQL injection vulnerabilities in web applications.

## **1. Comprehensive SQLi ToolsetQ**s**Automation & Exploitation**= or**SQLmap: The "Gold Standard." A powerful open-source tool that automates detecting and exploiting SQL injection flaws and taking over database servers.**

1. **Havij:** A well-known automated SQLi tool with a user-friendly GUI. Though legacy, it is often referenced for its easee.band SQLi

The attacker uses a different channel to retrieve results (e.g., forcing the database to make a DNS or HTTP request to a server the attacker controls). This is rare and used when the web server is too slow or unstable for Blind SQLi.

## 3. SQLi Hacking Methodology

1. **Information Gathering:** Identify all entry points (URL parameters, POST data, Cookies, HTTP Headers).
2. **Detection:** Use a single quote ' or double quote " to see if the server returns a database error.
3. **Exploitation:**
   * Identify the number of columns using ORDER BY.
   * Find which columns are vulnerable to display data.
   * Extract database names, table names, and column names.
   * Dump the data (Usernames/Passwords).

## 4. Specialized SQLi Tools

| **Tool** | **Focus** | **Description** |
| --- | --- | --- |
| **SQLmap** | Automation | The #1 tool for detecting and exploiting SQLi. |
| **Havij** | Windows GUI | A legacy automated SQLi tool with a user-friendly interface. |
| **BSQL Hacker** | Blind SQLi | Specifically optimized for automated Blind SQLi attacks. |
| **Mole** | Automation | An automated SQLi exploitation tool that relies on a CLI. |
| **Burp Suite** | Manual | Used for intercepting and manually fuzzing parameters with payloads. |

## 5. Countermeasures (The "Defense" Section)

* **Prepared Statements (Parameterized Queries):** The gold standard. Separates the query logic from the user data.
* **Stored Procedures:** Similar to prepared statements, but stored on the DB server.
* **Input Validation:** Use a "Whitelist" to only allow expected characters.
* **Principle of Least Privilege:** Ensure the DB user for the web app cannot run administrative commands (like DROP TABLE or SHUTDOWN).
* **Web Application Firewall (WAF):** To detect and block common SQLi patterns in real-time.