SQL Injection Tools — Collection Cheat Sheet

Quick, copy-paste ready reference covering the most useful SQLi tools and workflows

sqlmap

Automated CLI tool for detection, fingerprinting, enumeration and exploitation of SQL injection vulnerabilities.

Install / Quick note:

```
pip3 install sqlmap (or use distro package).
```

Quick command	sqlmap -u "http://target/?id=1" --batch --threads=5
Enumerate DBs	sqlmap -u "http://target/?id=1" --dbs --batch
Dump table	sqlmap -u "http://target/?id=1" -D db -T table --dump --batch
POST request	sqlmap -u "http://target/login" --data="user=admin&pass=1" --batch

Tip: Use --tamper scripts when WAF/filters block payloads. Start low-noise (--level=1 --risk=1).

Burp Suite (with extensions)

Intercepting proxy and suite for manual testing. Use Intruder/Repeater for payload testing and extensions for automation.

Install / Quick note:

```
Download from PortSwigger (Pro paid). Community edition available with manual
features.
```

Intercept & Repeater	Capture request → Send to Repeater → craft payloads
Intruder example	Send to Intruder → set payload positions → wordlist or payload generator
Extensions	SQLiPy, active scan rules, sqli extensions (BApp Store)

Tip: Use session handling rules and match/replace to maintain auth. Great for complex logic-based injections.

BBQSQL

Python-based blind SQLi exploitation tool using differential analysis. Good for tough blind/time cases.

Install / Quick note:

```
git clone https://github.com/Neohapsis/bbqsql.git && pip3 install -r
requirements.txt
```

Run	bbqsql -u 'http://target/?id=1' --data 'param=INJECT_HERE' --method POST
Mode	configure payload generator and comparator; useful for boolean/time blind

Tip: Requires tuning of payloads and a reliable comparator; slower but powerful when sqlmap fails.

sqlninja

Focused on Microsoft SQL Server exploitation (xp_cmdshell, file writes, shells).

Install / Quick note:

```
Available in many repos or from project page; configure using its config file before
running.
```

Run	sqlninja -m 1 -u 'http://target/?id=1' # then edit config for payloads
Goal	gain shell via xp_cmdshell or use file-based payloads

Tip: Tailored to MSSQL post-exploitation. Good follow-up after identifying MSSQL with sqlmap.

jSQL Injection

Java-based GUI tool for quick point-and-click SQLi testing across many DBMS.

Install / Quick note:

```
Download the jar from the project and run with `java -jar jsql.jar`.
```

Start	java -jar jsql.jar
Features	auto-detect injection, dump DBs, export results, multiple injection types

Tip: Useful for quick manual checks and demos; easier for novices.

NoSQLMap

Tool specialized in NoSQL (MongoDB, CouchDB, etc.) injection discovery and exploitation.

Install / Quick note:

```
pip3 install nosqlmap or clone project from GitHub.
```

Quick run	nosqlmap -u 'http://target/' --data 'user[$ne]=1'
Focus	NoSQL query injection, different payload patterns than SQL

Tip: Different universe of payloads; use when app uses Mongo/Couch/Redis backends.

SQLsus

Perl/PHP-era tool focused on MySQL exploitation and dumping.

Install / Quick note:

```
Often available in security repos; legacy but still used for some MySQL cases.
```

Run	sqlsus -u 'http://target/?id=1' --search 'password'
Strength	MySQL-focused heuristics and dump automation

Tip: Consider sqlmap first; use SQLsus when tight MySQL-specific techniques are needed.

Havij (historical)

Windows GUI automated SQLi tool; historically popular but flagged by AV and potentially unsafe.

Install / Quick note:

```
Windows binary, often from older sources; use only in isolated labs.
```

Features	auto-exploit, dump, GUI-driven workflows

Tip: Use only in controlled lab environments; avoid on personal machines without sandboxing.

OWASP ZAP

Open-source proxy/scanner similar to Burp; supports scripting and automation for SQLi testing.

Install / Quick note:

```
Install from package manager or download from OWASP ZAP website.
```

Active Scan	Use active scanner to find injection points
Scripting	Write scripts to craft advanced payloads

Tip: Good free alternative to Burp for CI integration and automated scanning.

Impacket (mssqlclient.py)

Collection of Python tools for network protocols; mssqlclient is great for direct MSSQL interaction post-exploit.

Install / Quick note:

```
pip3 install impacket
```

Connect	python3 mssqlclient.py target -windows-auth domain\\user:pass
Usage	use once you have creds or can authenticate; execute queries, run commands

Tip: Powerful for post-exploitation and pivoting in Windows environments.

Metasploit (DB modules)

Framework with modules that support SQLi discovery/exploitation and chaining to payloads.

Install / Quick note:

```
apt install metasploit-framework (Kali) or install from Rapid7.
```

Usage	use auxiliary modules for discovery and exploit modules for DB takeover
Integration	combine with post-ex modules after obtaining access

Tip: Excellent for exploitation chains and post-exploitation automation.

Recommended Workflow & OPSEC

- 1) Passive recon: collect endpoints, parameters and authentication flows (SpiderFoot, Amass, Burp).
- 2) Confirm injection manually using Burp Repeater or ZAP Proxy.
- 3) Use automated tools (sqlmap) for fingerprinting and enumeration; increase level/risk only after permission.
- 4) For blind or time-based issues, try BBQSQL or sqlmap time-based techniques.
- 5) If DBMS is MSSQL and OS commands are possible, use sqlninja/impacket/metasploit for post-exploitation.
- 6) Export results (sqlite/json), save sessions, and document command history for reporting.

Legal & OPSEC Reminder

Only test systems you own or have explicit written permission to test. Run destructive actions (file-write, os-pwn) only when explicitly authorized. Start with low-noise settings and monitor impact.