Tutorial: websqliattack

Overview

websqliattack is an automated security testing tool designed to detect SQL Injection vulnerabilities in web applications. It scrapes target websites, finds input fields, and applies both blind and non-blind SQL injection techniques. The project also includes vulnerable and protected backend applications to demonstrate both exploitation and secure coding practices.

Chapter 1: Vulnerable Backend Application

This demonstration backend is intentionally insecure. It features a simple login form that directly injects user input into SQL queries without validation. It allows attackers to bypass authentication and extract sensitive data. This backend is built with Node.js, Express, and MySQL.

Chapter 2: Protected Backend Application

The protected backend demonstrates best practices for secure coding. Protections include input validation using express-validator and parameterized queries with mysql2. This ensures that user inputs are always treated as data, not executable code.

Chapter 3: Attack Orchestrator

The orchestrator acts as the brain of websqliattack. It identifies the target web application type (React or non-React), scrapes for forms, and launches the appropriate attack modules. It also maintains a detailed log of each attack attempt.

Chapter 4: Web Application Scraper

The scraper identifies input fields and form submission points. It uses BeautifulSoup for static HTML websites and Selenium for dynamic applications built with React or similar frameworks. This ensures no potential injection point is missed.

Chapter 5: SQL Injection Attack (Non-Blind)

In non-blind attacks, payloads reveal immediate feedback, such as error messages or unexpected data. The tool leverages UNION SELECT queries to extract schema details like table and column names from the database.

Chapter 6: SQL Injection Attack (Blind)

In blind SQL injection, no direct errors or extra data are shown. Instead, websqliattack infers information character by character through boolean conditions and timing-based techniques. This allows gradual extraction of hidden values, such as passwords.

Chapter 7: Payload and Configuration Management

Payloads and common database names are centrally managed. This enables consistent reuse of attack strings and schema targets. Configuration files make it easy to extend the system with new payloads or adapt to different environments.

Chapter 8: Logging System

Every scan generates a timestamped log file that records start and end times, discovered vulnerabilities, and extracted schema details. These logs are crucial for reviewing results and improving attack coverage.

Disclaimer

This tool is provided for educational purposes only. Do not use websqliattack on systems you do not own or do not have explicit authorization to test. Unauthorized use may be illegal and subject to penalties.