

SQLMap

Jaylan Howden





Overview



SQL Injection

- Conducting SQL injection attacks manually can be a time consuming and tedious process.
- As one of the most common and dangerous vulnerabilities in web applications, SQL injection is a critical concern for security professionals.
- SQLMap is an open-source penetration testing tool that automates the detection and exploitation of SQL injection vulnerabilities.
- Web applications can greatly improve their safety precautions and avoid data breaches by knowing how what SQL injection is and how to prevent it..
- SQL injection attacks threaten the CIA Triad by compromising the confidentiality, integrity, and availability of data and systems, leading to unauthorized access, data manipulation, and service disruptions.



What is SQL Injection

- SQL Injection is a type of security vulnerability that occurs when an attacker is able to manipulate with the queries that an application makes to into database.
- SQL Injection can lead to unauthorized access to sensitive data such as personal information, intellectual property, and financial details.
- It allows attackers to view data that they are not normally able to retrieve from regular queries.
- SQL Injection is one of the most widespread and critical security threats to web applications.

A login form with a pink header bar containing the text "Please sign-in". Below the header, there are two input fields. The first field is labeled "Username" and contains the text "' or 1=1 --". The second field is labeled "Password" and is empty. Below the input fields is a blue "Login" button with a red hand cursor icon pointing at it.

Please sign-in

Username

Password

Login



- ```

root@kali:~# sqlmap -hh

{1.1.6#stable}
http://sqlmap.org

Usage: python sqlmap [options]

Options:
 -h, --help Show basic help message and exit
 -hh Show advanced help message and exit
 --version Show program's version number and exit
 -v VERBOSE Verbosity level: 0-6 (default 1)

Target:
 At least one of these options has to be provided to define the
 target(s)

 -d DIRECT Connection string for direct database connection
 -u URL, --url=URL Target URL (e.g. "http://www.site.com/vuln.php?id=1")
 -l LOGFILE Parse target(s) from Burp or WebScarab proxy log file
 -s SITEMAPURL Parse target(s) from remote sitemap(.xml) file
 -m BULKFILE Scan multiple targets given in a textual file
 -r REQUESTFILE Load HTTP request from a file
 -g GOOGLEDORK Process Google dork results as target URLs
 -c CONFIGFILE Load options from a configuration INI file

Request:
 These options can be used to specify how to connect to the target URL

 --method=METHOD Force usage of given HTTP method (e.g. PUT)
 --data=DATA Data string to be sent through POST
 --param-del=PARAM_DEL.. Character used for splitting parameter values
 --cookie=COOKIE HTTP Cookie header value
 --cookie-del=COOKIE_DEL.. Character used for splitting cookie values

```



# Tools



- SQLMap on linux
- Damn Vulnerable Web Application
- Virtual Machine with Apache Guacamole on Ubuntu
- Burp Suite (Linux)
- Web Browser (Firefox)
- Linux Command Line Terminal

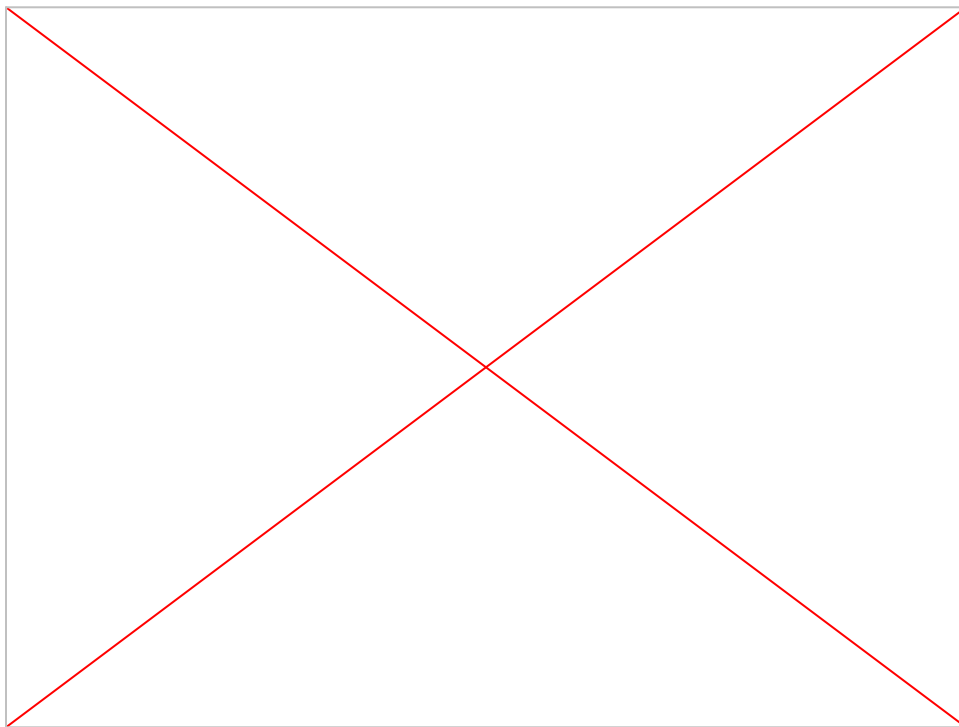


# Demonstration Preview

- Setting up DVWA(Damn Vulnerable Web Application) and testing queries
- Using the Burp Suite proxy to get HTTP request
- Setting parameters on SQLMap to start the SQL injection
- SQLMap uses the parameters to inject and attack the SQL database
- SQLMap then launches the attack and can get data such as database names, table names, usernames, and password hashes



# SQLMap Demonstration





# Demonstration Summary

- Using SQLmap on DVWA it shows how SQLMap can automate an attack on a vulnerable web application
- Using the HTTP proxy on Burp Suite you can see GET request which shows the request from the web application which SQLMap uses for the injection
- SQLMAP then launches the injection which enumerates the databases, dumps the tables in the database, and dumps the data from the table which includes usernames and password hashes which can be cracked.
- In Summary SQLMap can be a very useful tool for penetration testers when testing for web vulnerabilities and also be used by Security Analysts to identify and remediate SQL vulnerabilities on a web application.





# Mitigation Against SQL Injection

- Input validation and sanitization
- Web application firewalls
- Prepared Statements and Parameterized Queries which treat statements as data, not as executable code
- Keeping Web applications and databases up to date with updates and patches
- Provide thorough security awareness training to all relevant employees, including as system administrators, software developers, quality assurance teams, and DevOps engineers.
- Having a Incident response plan if a SQL Injection attack were to happen



**The End**

**Thank You**