CSYE6225 Fall 2018 - Assignment 10

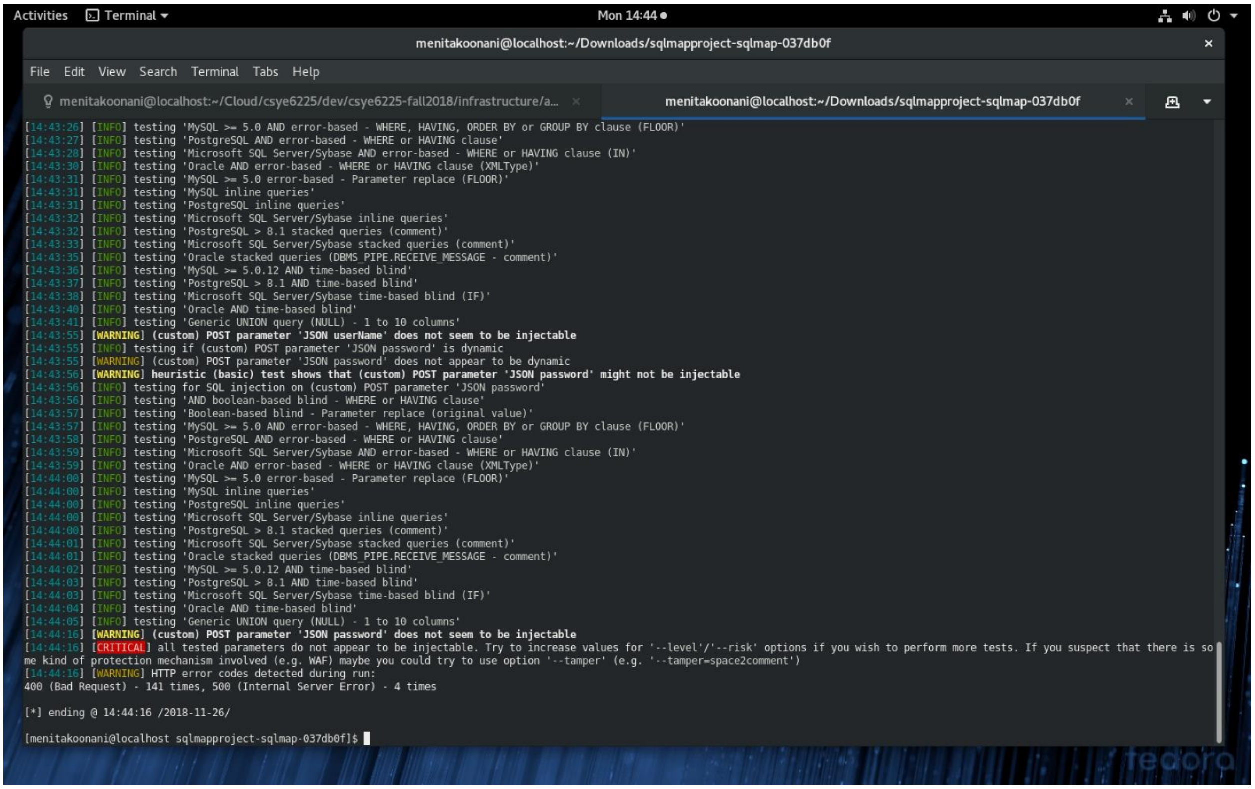
Penetration Testing

We have used below attack vectors to test our application running on EC2 instances

Below are the attack vectors we chose to test on our application and we have documented the results as below:

1. SQL Injection :

   Attack Vector : SQL Injection through SQL map
   SQL injection for URL endpoint user/register
   command used: python sqlmap.py -u "https://csye6225-fall2018-koonanim.me/user/register" --method=POST --data='{ "userName": "menitakoonani@gmail.com", "password": "menita" }'
   Result :



Activities
Terminal
Mon 14:44
menitakoonani@localhost:~/Downloads/sqlmapproject-sqlmap-037db0f
File
Edit
View
Search
Terminal
Tabs
Help
menitakoonani@localhost:~/Cloud/csye6225/dev/csye6225-fall2018/infrastructure/a...
menitakoonani@localhost:~/Downloads/sqlmapproject-sqlmap-037db0f

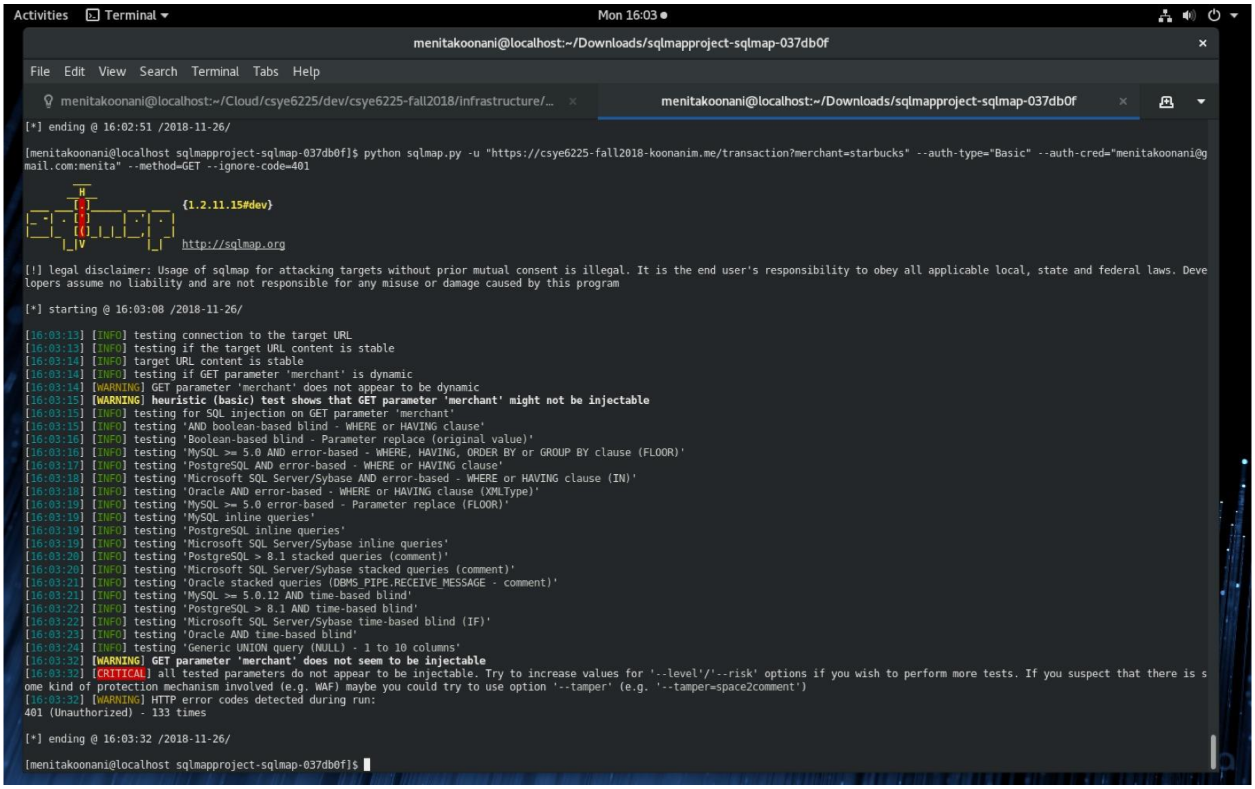
```
[14:43:26] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[14:43:27] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[14:43:28] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[14:43:30] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[14:43:31] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[14:43:31] [INFO] testing 'MySQL inline queries'
[14:43:31] [INFO] testing 'PostgreSQL inline queries'
[14:43:32] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[14:43:32] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[14:43:33] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[14:43:35] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[14:43:36] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[14:43:37] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[14:43:38] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[14:43:40] [INFO] testing 'Oracle AND time-based blind'
[14:43:41] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:43:55] [WARNING] (custom) POST parameter 'JSON userName' does not seem to be injectable
[14:43:55] [INFO] testing if (custom) POST parameter 'JSON password' is dynamic
[14:43:55] [WARNING] (custom) POST parameter 'JSON password' does not appear to be dynamic
[14:43:56] [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON password' might not be injectable
[14:43:56] [INFO] testing for SQL injection on (custom) POST parameter 'JSON password'
[14:43:56] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:43:57] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:43:57] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[14:43:58] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[14:43:59] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[14:43:59] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[14:44:00] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[14:44:00] [INFO] testing 'MySQL inline queries'
[14:44:00] [INFO] testing 'PostgreSQL inline queries'
[14:44:00] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[14:44:00] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[14:44:01] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[14:44:01] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[14:44:02] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[14:44:03] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[14:44:03] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[14:44:04] [INFO] testing 'Oracle AND time-based blind'
[14:44:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:44:16] [WARNING] (custom) POST parameter 'JSON password' does not seem to be injectable
[14:44:16] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is so
me kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment')
[14:44:16] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 141 times, 500 (Internal Server Error) - 4 times

[*] ending @ 14:44:16 /2018-11-26/

[menitakoonani@localhost sqlmapproject-sqlmap-037db0f]$
```

SQL injection with query string

command used: python sqlmap.py -u "https://csye6225-fall2018-koonanim.me/transaction?merchant=starbucks" --auth-type="Basic" --auth-cred="menitakoonani@gmail.com:menita" --method=GET --ignore-code=401



Activities
Terminal
Mon 16:03
menitakoonani@localhost:~/Downloads/sqlmapproject-sqlmap-037db0f
File
Edit
View
Search
Terminal
Tabs
Help
menitakoonani@localhost:~/Cloud/csye6225/dev/csye6225-fall2018/infrastructure/...
menitakoonani@localhost:~/Downloads/sqlmapproject-sqlmap-037db0f

```
[*] ending @ 16:02:51 /2018-11-26/

[menitakoonani@localhost sqlmapproject-sqlmap-037db0f]$ python sqlmap.py -u "https://csye6225-fall2018-koonanim.me/transaction?merchant=starbucks" --auth-type="Basic" --auth-cred="menitakoonani@g
mail.com:menita" --method=GET --ignore-code=401
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.2.11.15#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. Deve
lopers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 16:03:08 /2018-11-26/

[16:03:13] [INFO] testing connection to the target URL
[16:03:13] [INFO] testing if the target URL content is stable
[16:03:14] [INFO] target URL content is stable
[16:03:14] [INFO] testing if GET parameter 'merchant' is dynamic
[16:03:14] [WARNING] GET parameter 'merchant' does not appear to be dynamic
[16:03:15] [WARNING] heuristic (basic) test shows that GET parameter 'merchant' might not be injectable
[16:03:15] [INFO] testing for SQL injection on GET parameter 'merchant'
[16:03:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:03:16] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[16:03:16] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[16:03:17] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[16:03:18] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[16:03:18] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[16:03:19] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[16:03:19] [INFO] testing 'MySQL inline queries'
[16:03:19] [INFO] testing 'PostgreSQL inline queries'
[16:03:19] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[16:03:20] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[16:03:20] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[16:03:21] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[16:03:21] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[16:03:22] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[16:03:22] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[16:03:23] [INFO] testing 'Oracle AND time-based blind'
[16:03:24] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[16:03:32] [WARNING] GET parameter 'merchant' does not seem to be injectable
[16:03:32] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is s
ome kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment')
[16:03:32] [WARNING] HTTP error codes detected during run:
401 (Unauthorized) - 133 times

[*] ending @ 16:03:32 /2018-11-26/

[menitakoonani@localhost sqlmapproject-sqlmap-037db0f]$
```

Reasons:

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.