4417

tcet COMP ENGINEERING

TCET

DEPARTMENT OF COMPUTER ENGINEERING (COMP)

(Accredited by NBA for 3 years, 3rd Cycle Accreditation w.e.f. 1st July 2019)

Choice Based Credit Grading Scheme (CBCGS)

Under TCET Autonomy

Experiment no. 8

Aim: Simulation of SQL Injection Attack

Objectives: To understand about SQL Injection Attack

Outcomes: The learner will be able to understand and simulate SQL Injection Attack

Hardware / Software Required: Unix/Linux, SQLmap

Theory:

SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements (also commonly referred to as a malicious payload) that control a web application's database server (also commonly referred to as a Relational Database Management System – RDBMS). Since an SQL injection vulnerability could possibly affect any website or web application that makes use of an SQL-based database, the vulnerability is one of the oldest, most prevalent and most dangerous of web application vulnerabilities.

By leveraging an SQL injection vulnerability, given the right circumstances, an attacker can use it to bypass a web application's authentication and authorization mechanisms and retrieve the contents of an entire database. SQL injection can also be used to add, modify and delete records in a database, affecting data integrity.

To such an extent, SQL injection can provide an attacker with unauthorized access to sensitive data including, customer data, personally identifiable information (PII), trade secrets, intellectual property and other sensitive information.

How SQL Injection works

In order to run malicious SQL queries against a database server, an attacker must first find an input within the web application that is included inside of an SQL query.

In order for an SQL injection attack to take place, the vulnerable website needs to directly include user input within an SQL statement. An attacker can then insert a payload that will be included as part of the SQL query and run against the database server.

SQLMAP: 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.

Step 1: Installation of sqlmap

```
$ sudo apt-get install sqlmap
```

Step 2 : List information about the existing databases

To check access to a database, - - dbs option can be used. - - dbs lists all the available databases.

It notifies vulnerability in parameter cat, various payloads executed, name of backend database, its version and list of all available databases. Here, two databases: acuart and information_schema are listed.

```
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
```

Step 3: Listing tables present in Database

Each of the database can further explored to get tables information from them. Option -D can be used to specify the name of the database we need to explore. If access to the database is allowed, we can access the tables using --tables option along with name of database. Here, acuart database is accessed and all available tables in that database are listed as an output of the following command.

```
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
```

Step 4: List column information of a particular table

Columns of a particular table can be viewed by specifying -T option before table name and --columns option to query the column names. Access to table and its column for table "products" is displayed by following command.

```
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T products –
columns
```

Step 5: Dump the data from the columns

Information from specific column can be retrieved and displayed using -C. Multiple column can also be listed separated by a comma and the –dump query retrieves the data. Flowing command shows all Domain values of column name from product table from acuart database.

```
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T products -C
name -dump
```