

LAB EXPERIMENT : Testing Authentication Weaknesses and Session Management Using Kali Linux & DVWA

AIM

To identify and analyze authentication weaknesses and session management vulnerabilities using DVWA in Kali Linux.

REQUIREMENTS

Software

- Kali Linux
- DVWA (Pre-installed on lab systems)
- Web Browser (Firefox)

Hardware

- Computer System with Internet Disabled (Lab Setup)

THEORY

Authentication Weakness

Authentication ensures that only valid users can log in. Weak authentication occurs due to:

- Weak passwords
- No account lockout
- Brute force vulnerability
- Default credentials

Session Management

Session management handles user sessions using session IDs. Improper session handling leads to:

- Session hijacking
- Session fixation
- Reuse of old session IDs
- Insecure cookies

PROCEDURE

PART A: Launch DVWA

Step 1: Start Required Services

Open terminal and start Apache and MySQL:

```
sudo service apache2 start
```

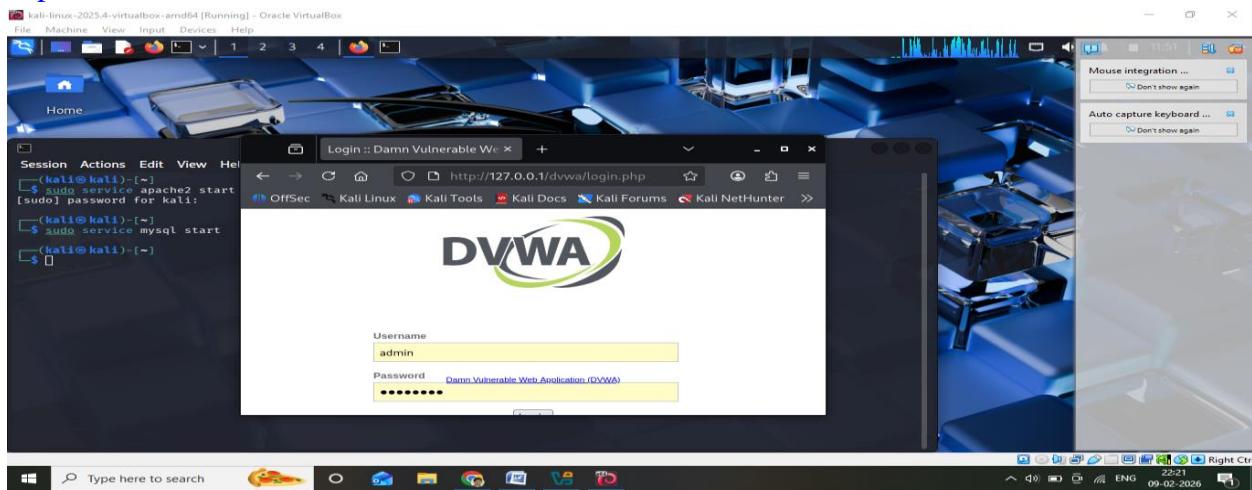
```
sudo service mysql start
```



Step 2: Open DVWA in Browser

Open Firefox and enter:

<http://127.0.0.1/dvwa>

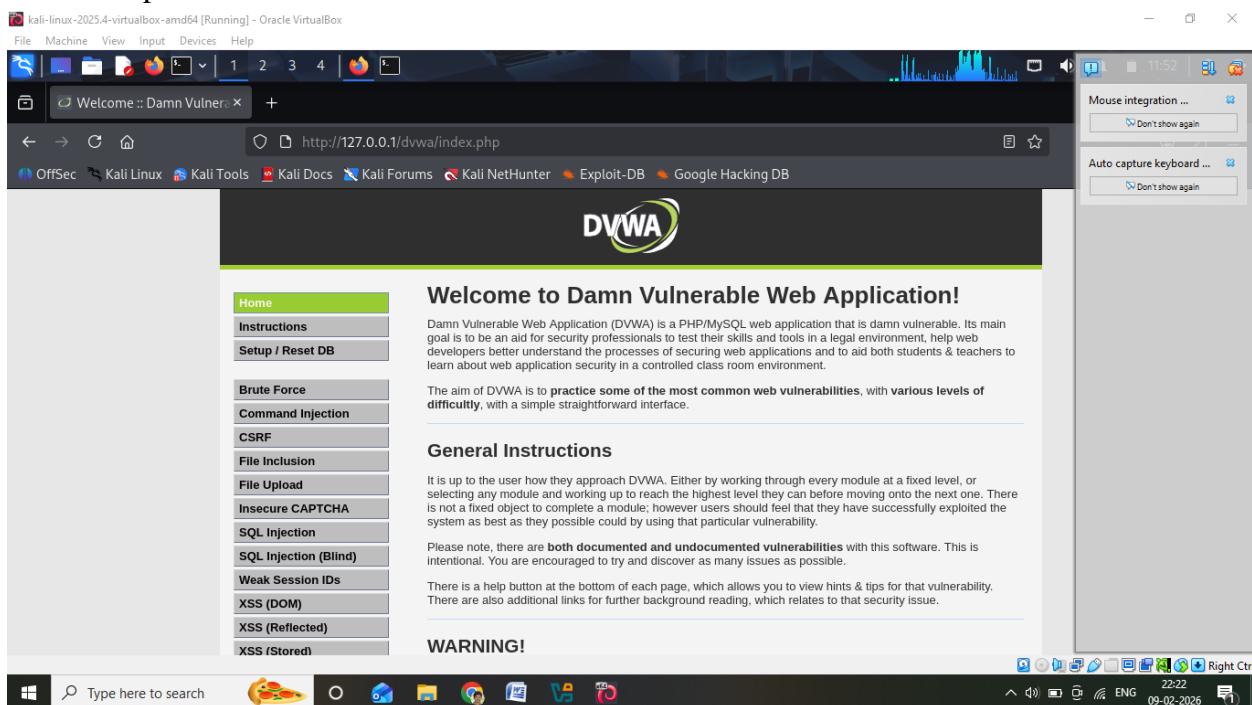


Step 3: Login to DVWA

Use default credentials:

Username: admin

Password: password



Step 4: Set Security Level

- Go to **DVWA Security**
- Select **LOW**

- Click Submit

DVWA Security :: Damn Vulnerable Web Application

<http://127.0.0.1/dvwa/security.php>

DVWA

DVWA Security 🔒

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and has **no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code. Prior to DVWA v1.9, this level was known as 'high'.

Low ▾ **Submit**

PART B: Testing Authentication Weaknesses

Experiment 1: Weak Password Authentication

Step 1: Open Brute Force Module

Navigate to:

DVWA → Vulnerabilities → Brute Force

Vulnerability: Brute Force

Login

Username:

Password:

More Information

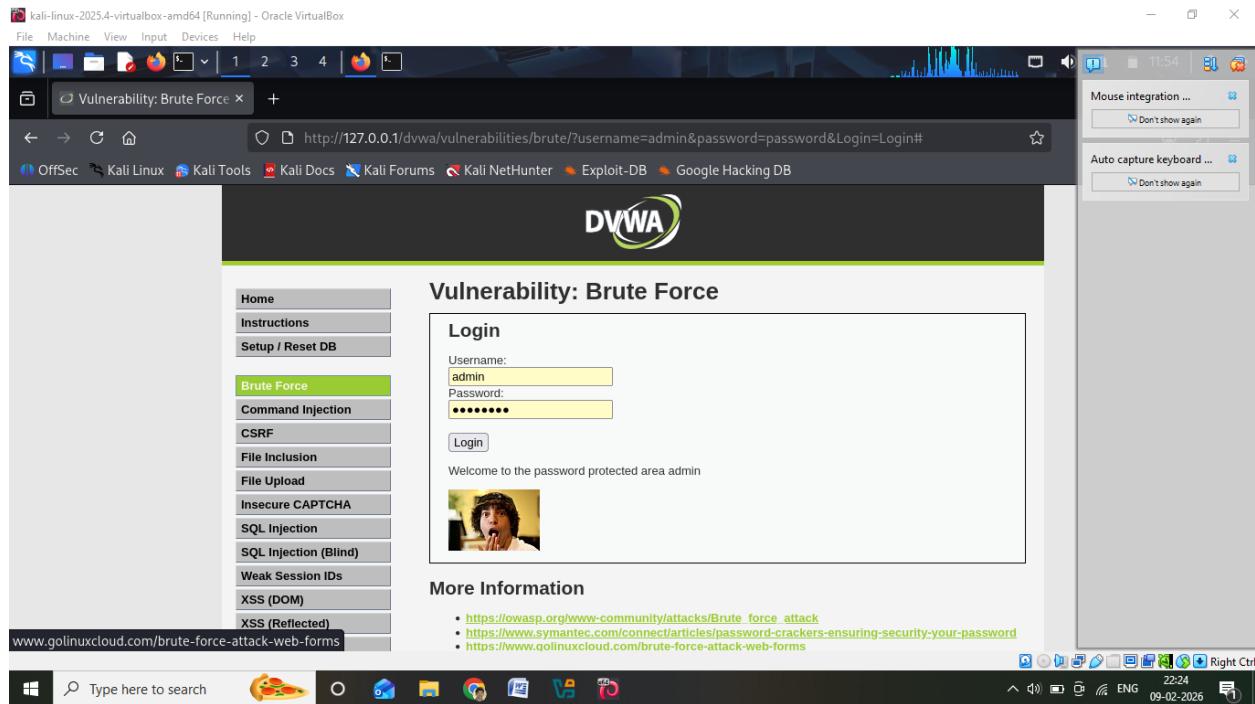
- https://owasp.org/www-community/attacks/Brute_force_attack
- <https://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <https://www.golinuxcloud.com/brute-force-attack-web-forms>

Step 2: Try Common Passwords

Enter:

Username: admin

Password: password



Observation

Successful login indicates weak authentication.

Experiment 2: Manual Brute Force Attack

Enter Username (Same Every Time)

In **Username** field, type:

admin

Do NOT change username.

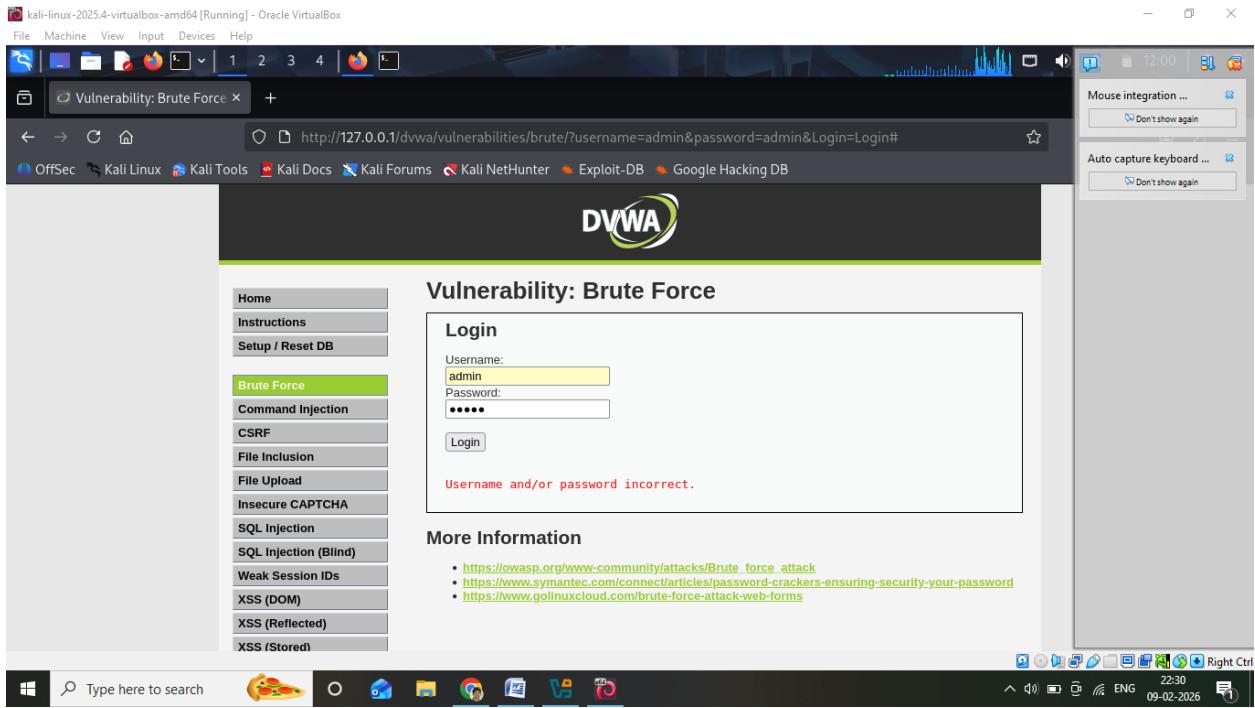
Step 3: Try Passwords ONE BY ONE

Now you will **manually try** passwords (this is the “manual brute force”).

Attempt 1

- Username: admin
- Password: admin
- Click **Login**

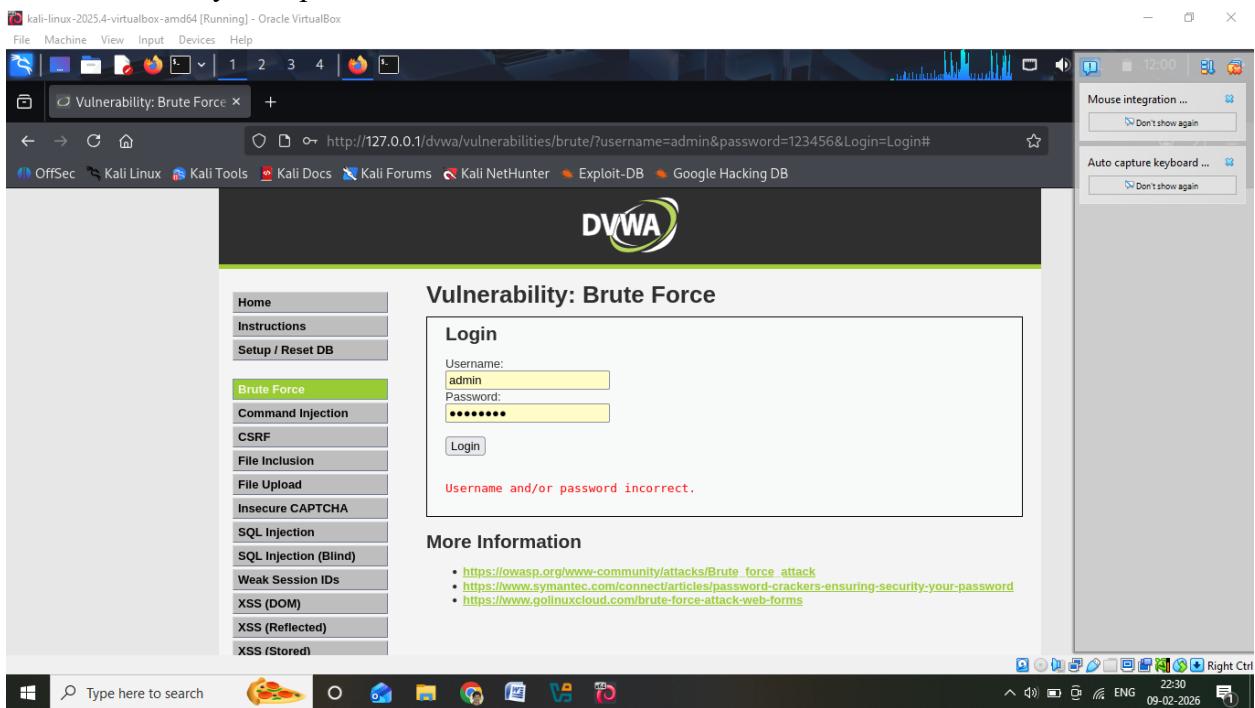
✖ If it fails → try next password



Attempt 2

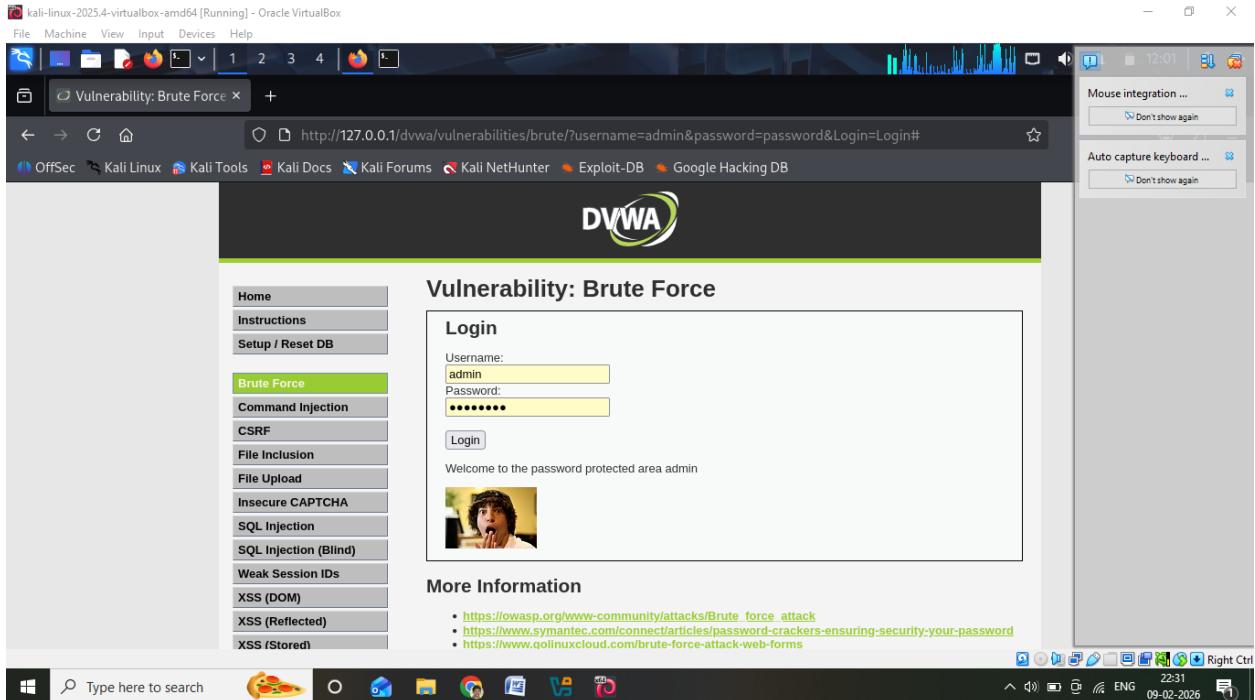
- Username: admin
- Password: 123456
- Click Login

✗ If it fails → try next password



Attempt 3

- Username: admin
- Password: password
- Click **Login**



LOGIN SUCCESSFUL

Step 4: Observe What Happened

- DVWA did NOT block you
- DVWA did NOT lock account
- DVWA allowed unlimited attempts

This is called **Brute Force Vulnerability**

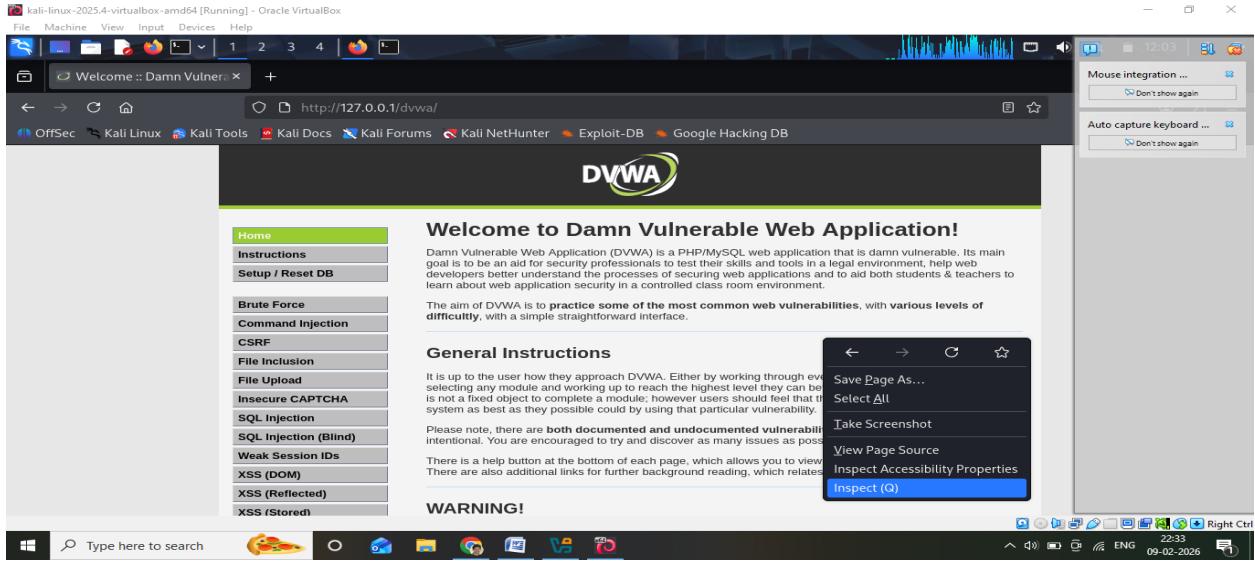
PART C: Testing Session Management Vulnerabilities

❖Experiment 3: Session ID Analysis

Step 1: Login to DVWA

Open browser developer tools:

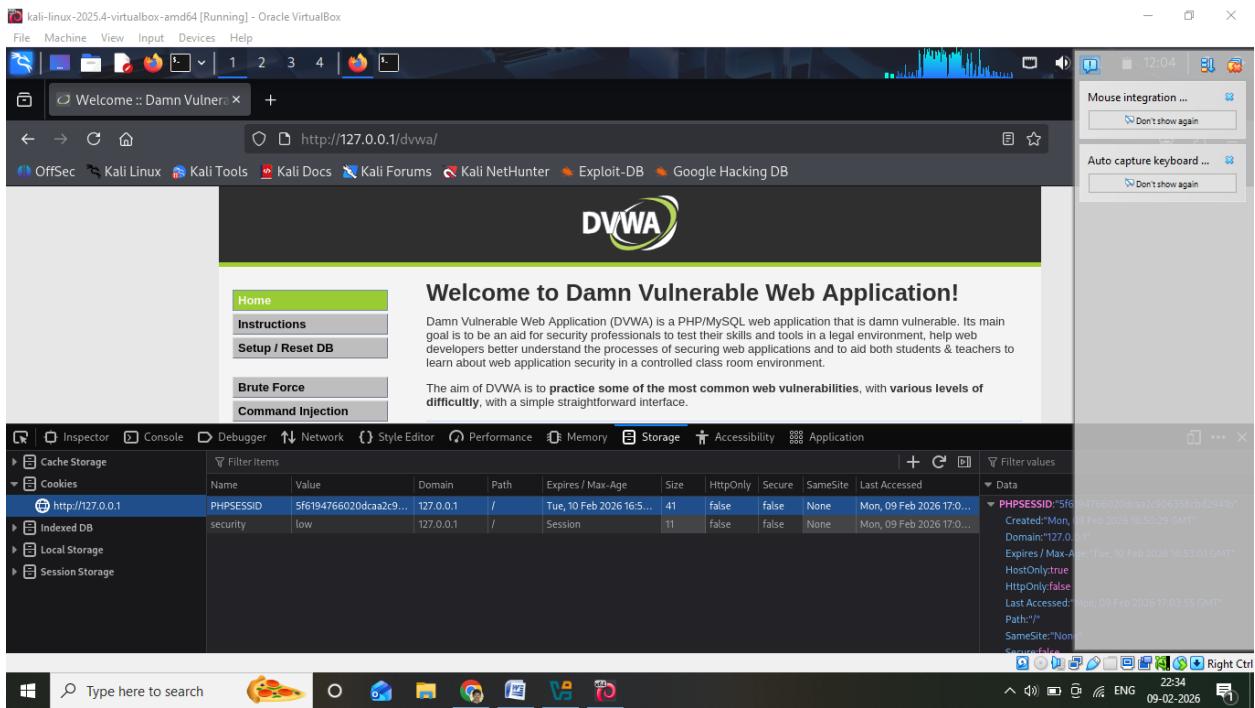
Right Click → Inspect → Storage → Cookies



Step 2: Observe Session Cookie

Look for:

`PHPSESSID`



Observation

Session ID is visible and not encrypted.

`PHPSESSID : 5f6194766020dcaa2c906358cbd2941b`

Experiment 4: Session Hijacking

BEFORE YOU START (IMPORTANT)

DVWA security level = **LOW**

You are **logged in as admin** in DVWA

STEP-BY-STEP

Step 1: Open DVWA (Victim Session)

1. Open **Firefox**
2. Go to: <http://127.0.0.1/dvwa>
3. Login:

Username: admin

Password: password

4. Stay logged in (do NOT logout)

This browser is the **Victim**

Step 2: Copy the Session ID (PHPSESSID)

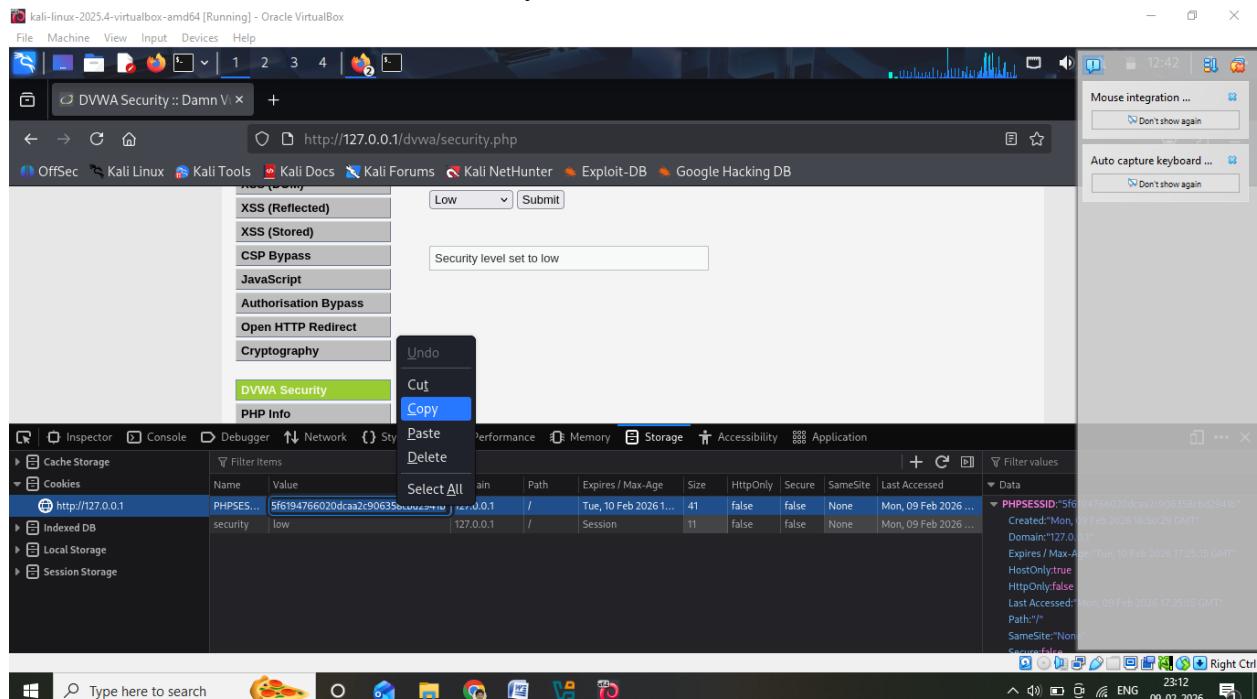
1. In the same Firefox window
2. **Right click → Inspect**
3. Click **Storage** tab
4. Click **Cookies**
5. Select: <http://127.0.0.1>

You will see something like:

PHPSESSID a8c9f7e3d4b1...

6. **Right-click on PHPSESSID value → Copy**

This value is the **session ID** (user identity).



Step 3: Open Attacker Browser (Private Window)

1. Press:

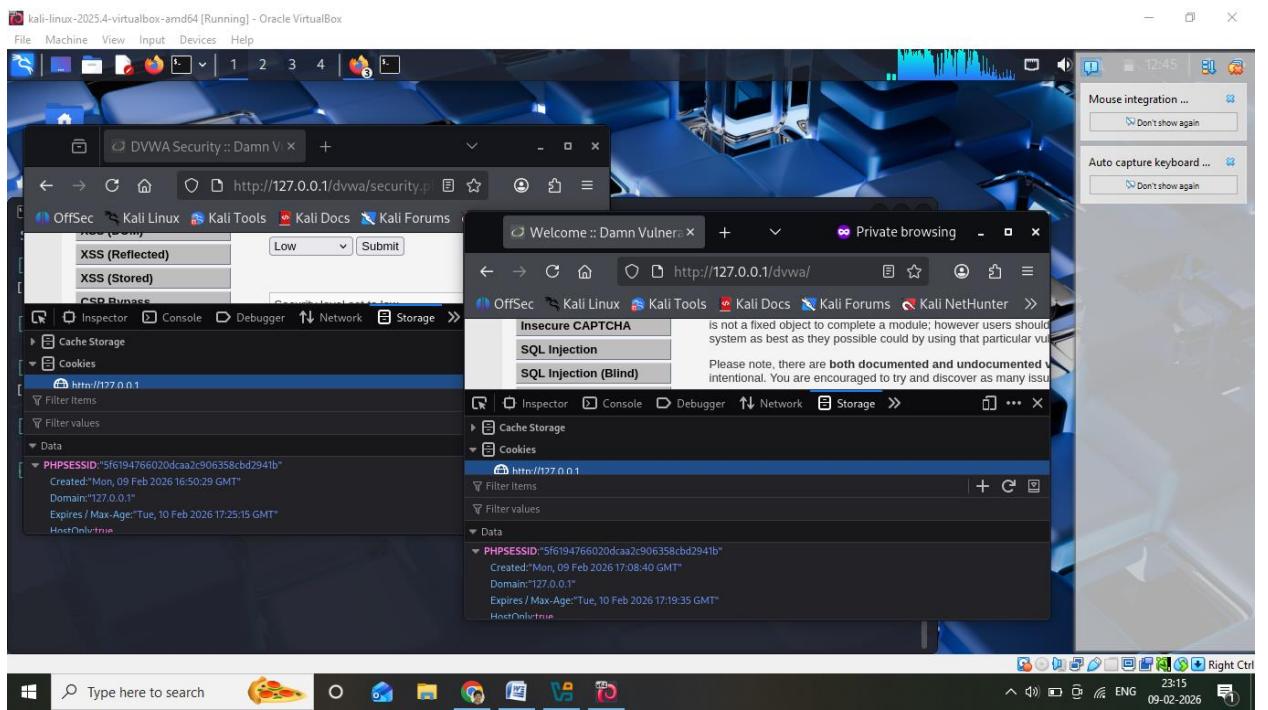
Ctrl + Shift + P

(Private Window opens)

Do NOT login here.

Step 4: Paste Session ID in Attacker Browser

1. In **Private Window**, go to: http://127.0.0.1/dvwa
2. Right click → **Inspect**
3. Go to **Storage** → **Cookies**
4. Click: http://127.0.0.1
5. Find **PHPSESSID**
6. **Replace its value** with the copied PHPSESSID (5f6194766020dcaa2c906358cbd2941b)
7. Press **Enter**



Step 5: Refresh Page

1. Refresh the page (F5)

You are logged in as admin without username or password!

Result

Attacker gains access without login → Session Hijacking.

Experiment 5: Session Fixation

IMPORTANT CONDITIONS (CHECK FIRST)

DVWA Security Level = **LOW**

Use **only ONE browser window** (normal window)

Do **NOT** use Private Window here

STEP-BY-STEP (DO EXACTLY THIS)

Step 1: Open DVWA WITHOUT Login (Attacker sets session)

1. Open **Firefox**
2. Go to: <http://127.0.0.1/dvwa/>

You will see the **login page**

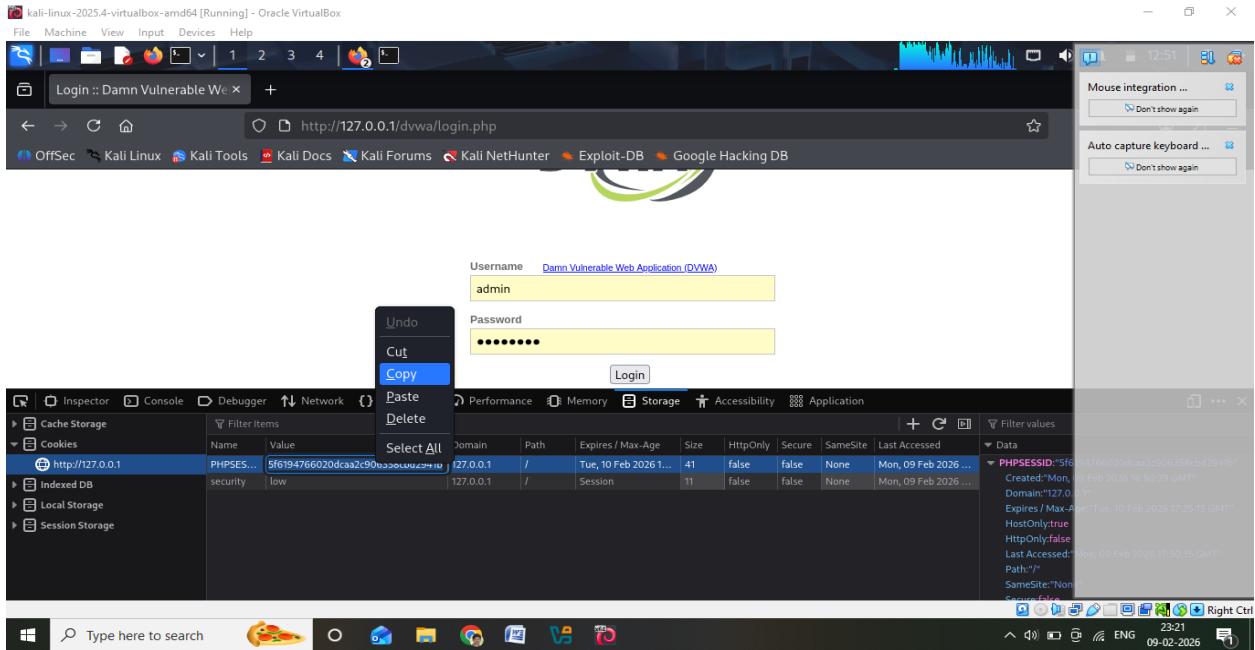
Do NOT login

Step 2: Note the Session ID (Before Login)

1. Right click → **Inspect**
2. Go to **Storage**
3. Click **Cookies**
4. Select: <http://127.0.0.1>

You will see:

PHPSESSID = 5f6194766020dcaa2c906358cbd2941b



Step 3: Login WITHOUT Closing Browser

Now, in the **same browser window**:

1. Enter:

Username: admin

Password: password

2. Click **Login**

Do NOT refresh, do NOT close browser

Step 4: Check Session ID AGAIN (After Login)

1. Again open:

Inspect → Storage → Cookies → http://127.0.0.1

2. Look at PHPSESSID

The screenshot shows a Kali Linux desktop environment with a Firefox browser window open to the DVWA Low application. The browser address bar shows `http://127.0.0.1/dvwa/index.php`. The DVWA logo is visible on the page. In the Firefox developer tools, the 'Storage' tab is selected, and the 'Cookies' section is expanded. A context menu is open over a cookie entry for 'PHPSESSID'. The menu options include Undo, Cut, Paste, Delete, Select All, and Copy, with 'Copy' being highlighted. The cookie table shows one row for 'PHPSESSID' with the value `5f6194766020dcaa2c906358cbd2941b`. The cookie details pane on the right shows the following properties:
Name: PHPSESSID
Value: 5f6194766020dcaa2c906358cbd2941b
Domain: 127.0.1
Expires / Max-Age: Tue, 10 Feb 2026 17:52:13 GMT
HttpOnly: false
HostOnly: true
Last Accessed: Mon, 09 Feb 2026 17:52:13 GMT
Path: /
SameSite: None

3.

OBSERVE CAREFULLY

Case 1 (VULNERABLE – DVWA LOW)

Before Login PHPSESSID = 5f6194766020dcaa2c906358cbd2941b

After Login PHPSESSID = 5f6194766020dcaa2c906358cbd2941b

Same value

Session Fixation exists

Case 2 (SECURE – DVWA HIGH / IMPOSSIBLE)

Before Login PHPSESSID = 5f6194766020dcaa2c906358cbd2941b

After Login PHPSESSID = be2d584526b42fef6742d5cf95ce008f

The screenshot shows a Kali Linux desktop environment with a Firefox browser window open to the DVWA Security application. The browser address bar shows `http://127.0.0.1/dvwa/security.php`. The DVWA logo is visible on the page. In the Firefox developer tools, the 'Storage' tab is selected, and the 'Cookies' section is expanded. A context menu is open over a cookie entry for 'PHPSESSID'. The menu options include Undo, Cut, Paste, Delete, Select All, and Copy, with 'Copy' being highlighted. The cookie table shows one row for 'PHPSESSID' with the value `be2d584526b42fef6742d5cf95ce008f`. The cookie details pane on the right shows the following properties:
Name: PHPSESSID
Value: be2d584526b42fef6742d5cf95ce008f
Domain: 127.0.1
Expires / Max-Age: Tue, 10 Feb 2026 17:56:36 GMT
HttpOnly: true
HostOnly: true
Last Accessed: Mon, 09 Feb 2026 17:56:36 GMT
Path: /
SameSite: Strict

Session regenerated

No session fixation

Experiment 6:

CONDITIONS (CHECK FIRST)

DVWA Security Level = **LOW**

You must know how to **view cookies**

STEP-BY-STEP (

Step 1: Login Normally (Victim Session)

1. Open Firefox
2. Go to: <http://127.0.0.1/dvwa/>
3. Login:

Username: admin

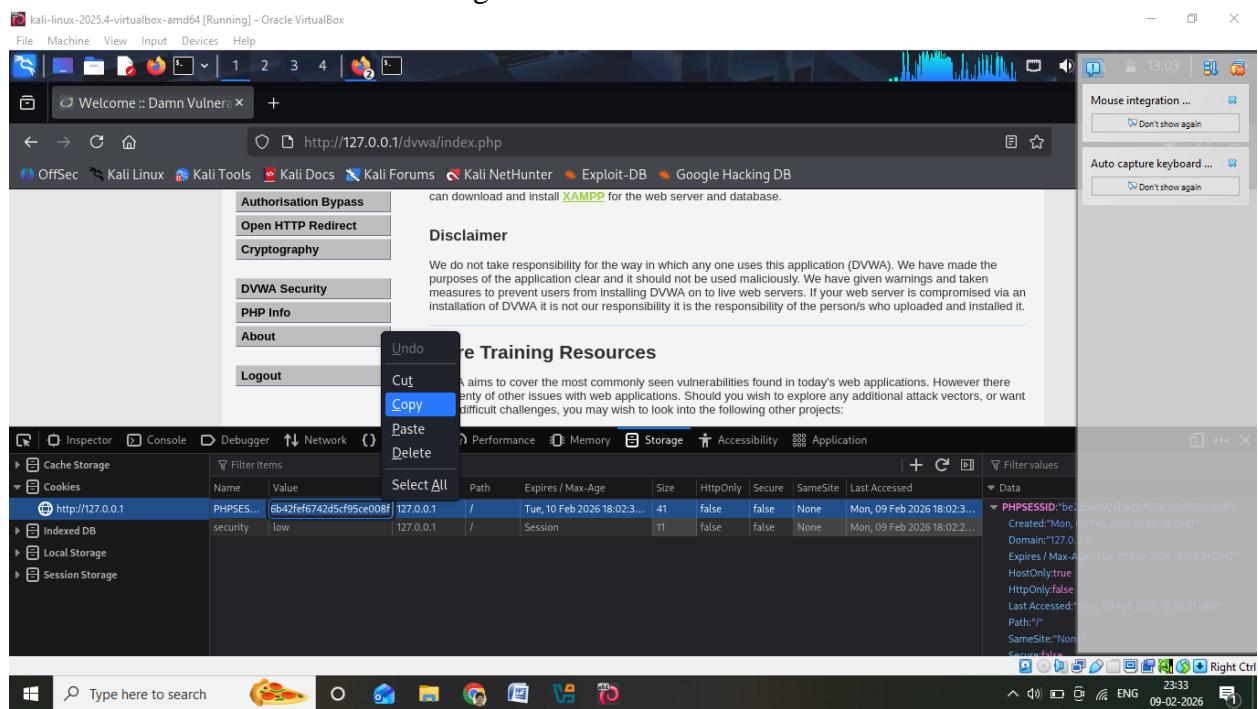
Password: password

Step 2: Copy Session ID (IMPORTANT)

1. Right click → **Inspect**
2. Storage → Cookies → <http://127.0.0.1>
3. Copy:

PHPSESSID = be2d584526b42fef6742d5cf95ce008f

Screenshot 1: PHPSESSID before logout



Step 3: Logout from DVWA

1. Click **Logout** (top right or menu)
2. You will see **login page**

Logout completed

Step 4: Reuse OLD Session ID (THIS IS THE TEST)

Option A (EASIEST & EXAM-SAFE)

1. Open Private Window

Ctrl + Shift + P

2. Go to:

http://127.0.0.1/dvwa/

3. Open Inspect → Storage → Cookies
4. Paste the **OLD PHPSESSID** (copied earlier)
5. Press Enter

Step 5: Open Internal Page (KEY STEP □)

In address bar, type:

http://127.0.0.1/dvwa/index.php

(or)

http://127.0.0.1/dvwa/vulnerabilities/brute/

- Do NOT press Login
- Do NOT enter username/password

The screenshot shows a Kali Linux desktop environment with several windows open. In the foreground, a Firefox browser window displays the "Welcome to Damn Vulnerable Web Application!" page. Below the browser, a "Storage" application window is open, specifically showing the "Cookies" tab. A single cookie is selected in this list, with its detailed properties displayed in a right-hand panel. The cookie's name is "PHPSESSID", its value is a long alphanumeric string, and it has various attributes like "Domain: 127.0.0.1", "Path: /", and "Expires / Max-Age: Tue, 10 Feb 2026 17:19:35 GMT". The "Secure" and "HttpOnly" flags are set to "false". The "SameSite" attribute is listed as "None". The "Last Accessed" timestamp is "Mon, 09 Feb 2026 18:10:31 GMT".

□ EXPECTED RESULT (DVWA LOW)

- ✓ You are logged in again
 - ✓ Without login
 - ✓ Using old session ID
- Logout did NOT destroy session**

OBSERVATIONS & RESULTS

Test Case	Result
Weak Password Login	Successful
Brute Force Attack	Allowed
Session ID Exposure	Found
Session Hijacking	Possible
Session Fixation	Observed
Improper Logout	Observed

Authentication and session management vulnerabilities were successfully identified in DVWA using Kali Linux. This experiment demonstrates the importance of secure authentication and proper session handling to prevent unauthorized access.

VIVA VOCE QUESTIONS

1. What is authentication?
2. What is brute force attack?
3. What is session hijacking?
4. What is session fixation?
5. How can session attacks be prevented?