Vulnerability Report: Weak Session ID in DVWA

Overview

This report covers the analysis and testing of the *Weak Session ID* vulnerability in the Damn Vulnerable Web Application (DVWA). The vulnerability was explored using the DVWA interface under different security levels. Screenshots and Developer Tools were used to examine and interpret session behaviour.

Steps Performed

Step 1: Open DVWA and Set Security Level

- Launched DVWA through localhost/dvwa.
- Logged in using default credentials.
- Navigated to the "DVWA Security" tab.
- Set the security level to **Low**, **Medium**, **High**, and **Impossible** to test behaviour under each condition.

Step 2: Navigate to "Weak Session IDs" Module

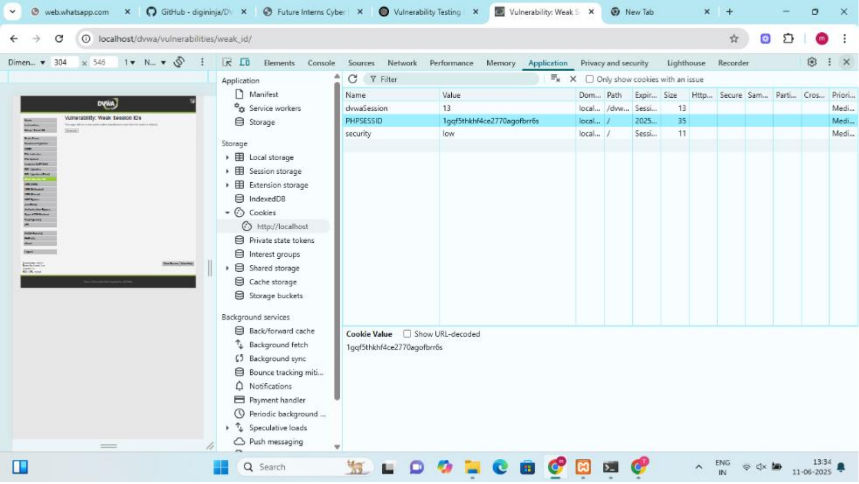
- Clicked on **"Weak Session IDs"** from the left navigation panel.
- This module allows observation of how session identifiers (IDs) are handled.

Step 3: Open Developer Tools

- Pressed F12 or right-clicked anywhere on the browser page and selected **"Inspect"**.
- Switched to the **Application** tab inside Developer Tools.
- Under **Storage**, expanded **Cookies**.

Step 4: Check for Session Cookie

- Looked for a cookie entry matching the DVWA URL (e.g., localhost).
- Observed session identifier name: typically PHPSESSID.
- In the shared screenshot, no DVWA-specific cookie (like localhost) was listed under Cookies, which indicates the application page might not have been active in that specific tab.



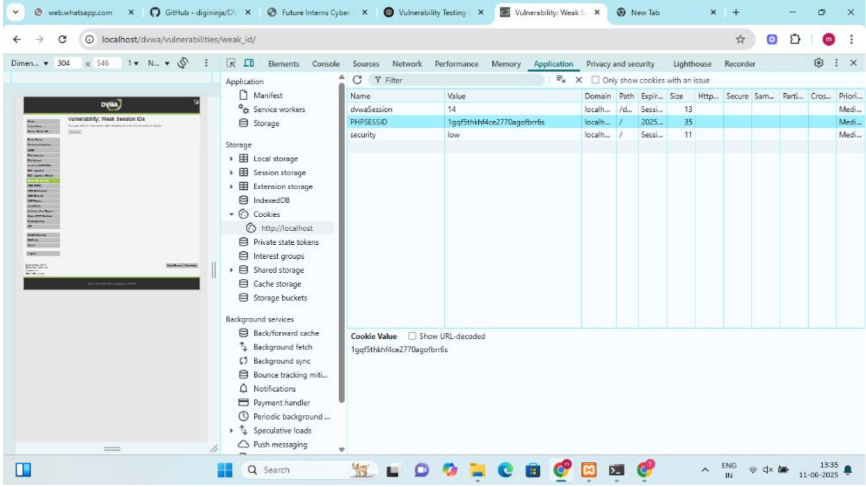
Name	Value	Dom...	Path	Expir...	Size	Http...	Secure	Sam...	Parti...	Cros...	Priori...
dvwaSession	13	local...	/dvw...	Sessi...	13						Medi...
PHPSESSID	1gqf5thkhf4ce2770agofbrr6s	local...	/	2025...	35						Medi...
security	low	local...	/	Sessi...	11						Medi...

Cookie Value ☐ Show URL-decoded
1gqf5thkhf4ce2770agofbrr6s

Note: You must first visit the DVWA application in the **same tab** where Developer Tools are opened in order to inspect its cookies.

Step 5: Check if Session ID Changes

- Refreshed the page and rechecked the session ID.
- Logged out and logged back in to see if the session ID changes.
- If the session ID remains **short**, **predictable**, or does **not** change after login/logout cycles, the session management is considered weak.



Name	Value	Domain	Path	Expir...	Size	Http...	Secure	Sam...	Parti...	Cros...	Priori...
dvwaSession	14	localh...	/d...	Sessi...	13						Medi...
PHPSESSID	1gqf5thkhf4ce2770agofbrr6s	localh...	/	2025...	35						Medi...
security	low	localh...	/	Sessi...	11						Medi...

Cookie Value □ Show URL-decoded

1gqf5thkhf4ce2770agofbrr6s

Note: The Session ID has not changed it remains the same, therefore the session management is considered as weak.

Security Level	**Observation**
Low	Session IDs are short and predictable (e.g., numeric or incremental).
Medium	Session IDs become slightly more complex but are still not cryptographically secure.
High	More randomized session IDs; still needs validation against secure standards.
Impossible	Strong and unpredictable session IDs; follows best security practices.

Security Concern

Weak Session IDs make it easier for attackers to:

- **Guess session tokens** through brute force or prediction.
- **Hijack sessions** if they obtain a valid session ID.
- Exploit poor session management, especially if HTTPS is not enforced.

Recommendations

- Use **cryptographically strong**, **random** session identifiers.
- Enforce **HTTPS** to secure session transmission.
- Implement **session expiration** and **logout mechanisms**.
- Rotate session IDs upon login and privilege escalation.