Perfect – this is a **core part of WSTG-SESS-02 (Testing for Cookies Attributes)** 🍪

**🧪 Testing for Cookies Attributes**

When a web app sets cookies, **security flags & attributes** determine how safe they are against theft, misuse, or manipulation. As a tester, you must inspect these in the HTTP response headers and browser storage.

**🔑 Important Cookie Attributes**

**1. Secure**

* Ensures cookie is **only sent over HTTPS**.
* Prevents leakage via plaintext HTTP traffic.
* Missing = risk of **session hijacking via MITM**.
* Example:
* Set-Cookie: SESSIONID=abc123; Secure

**2. HttpOnly**

* Prevents cookie from being accessed by **JavaScript (document.cookie)**.
* Mitigates **XSS-based session theft**.
* Missing = attacker can steal tokens if XSS exists.
* Example:
* Set-Cookie: SESSIONID=abc123; HttpOnly

**3. SameSite**

* Restricts sending cookies with **cross-site requests** (CSRF protection).
* Values:
  + Strict → Best, only first-party requests.
  + Lax → Allows top-level navigations (login forms).
  + None → Allows cross-site requests, **must** be Secure.
* Example:
* Set-Cookie: SESSIONID=abc123; SameSite=Strict

**4. Domain**

* Defines which hosts can receive the cookie.
* If set too broad (.example.com), subdomains can hijack.
* Example:
* Set-Cookie: SESSIONID=abc123; Domain=app.example.com

**5. Path**

* Defines which URL path can access the cookie.
* Should be as restrictive as possible (/app/).
* Example:
* Set-Cookie: SESSIONID=abc123; Path=/app/

**6. Expires / Max-Age**

* Controls cookie lifetime.
* **Session cookie** (default) → deleted when browser closes.
* Expires or Max-Age → persistent cookie.
* Risk: long-lived cookies increase exposure window.
* Example:
* Set-Cookie: SESSIONID=abc123; Max-Age=3600

**7. Priority (less common but useful)**

* Some browsers allow cookie eviction control.
* Priority=High → session cookies not cleared easily.

**8. Partitioned (CHIPS) (modern browsers)**

* Restricts cookies to the top-level site.
* Mitigates **tracking across origins**.
* Example:
* Set-Cookie: SESSIONID=abc123; Partitioned

**🔍 How to Test**

**Using Burp Suite**

1. Capture login response with Set-Cookie.
2. Inspect attributes under **HTTP history → Cookies tab**.
3. Check: Missing **Secure**, **HttpOnly**, **SameSite**? Overly broad **Domain**/ **Path**?

**Using Browser DevTools**

* Open **Application → Storage → Cookies**.
* Review each cookie’s flags.

**Using Command-line**

curl -I https://target.com | grep -i set-cookie

**✅ Best Practices**

* Always set **Secure + HttpOnly + SameSite=Strict (or Lax)**.
* Narrow **Domain** and **Path** scope.
* Avoid long-lived cookies unless required.
* Rotate/expire session cookies on logout & privilege change.

⚡ **Summary:**

Cookies are the backbone of session management. Missing attributes = easy targets for **XSS, CSRF, MITM, and session hijacking**.

👉 Do you want me to make a **quick Burp testing checklist for cookies** so you can use it directly in labs/exams?