Great topic 👍 Let’s walk through **Testing for Sensitive Information Sent via Unencrypted Channels** — this is a core part of **WSTG Information Transport Security** checks.

**🧾 Testing for Sensitive Information via Unencrypted Channels**

**1. Why It Matters**

If applications transmit sensitive data (logins, cookies, tokens, PII) over **unencrypted channels (HTTP, plain FTP, Telnet, etc.)**, attackers can sniff traffic and steal credentials.

* Tools like **Wireshark** or **tcpdump** make interception trivial.
* MitM attacks become possible on public WiFi or compromised routers.

**2. What to Test**

🔹 **Web Applications**

* Check if login, password reset, or profile pages use **HTTP instead of HTTPS**.
* Inspect if cookies (Set-Cookie) are marked with Secure and HttpOnly.
* Verify if APIs use HTTPS or still expose endpoints via plain HTTP.

🔹 **Services**

* Email: POP3/IMAP/SMTP running on ports **110, 143, 25** without TLS.
* FTP: transmitting creds in plain text.
* SNMPv1/v2: community strings sent unencrypted.

**3. Tools & Techniques**

**🔸 Using sslscan**

* sslscan checks if a service supports SSL/TLS properly.
* Can detect weak/invalid certificates, and vulnerabilities like **Heartbleed (CVE-2014-0160)**.

Example:

sslscan example.com

What to look for:

* Services still accepting **SSLv2/SSLv3, TLS 1.0** (obsolete).
* Weak ciphers (RC4, DES, NULL).
* Certificates that are self-signed or expired.
* Heartbleed test → reports if OpenSSL is vulnerable.

**🔸 Other Tools**

* **Wireshark**: capture traffic → look for cleartext creds.
* **nmap**:
* nmap --script ssl-enum-ciphers -p 443 example.com
* **testssl.sh**: more modern alternative to sslscan.

**4. How to Confirm Sensitive Data Leakage**

* Login with test credentials → capture traffic with Wireshark.
* If you see username/password in cleartext → **FAIL**.
* Check cookie flags in HTTP headers:
* Set-Cookie: sessionid=12345; Path=/; Secure; HttpOnly

❌ If Secure is missing, cookie can be stolen over HTTP.

**5. Preventive Controls**

✅ Enforce **HTTPS everywhere** (HSTS headers).  
✅ Redirect all HTTP traffic → HTTPS.  
✅ Use **TLS 1.2 or higher**, strong ciphers only.  
✅ Use valid certificates from trusted CAs.  
✅ Encrypt sensitive protocols (use FTPS/SFTP, SNMPv3, SMTPS).  
✅ Monitor with SIEM for downgrade attempts or sniffing activity.

✅ **Summary**:  
Testing unencrypted channels is about spotting weak SSL/TLS or plain-text traffic. sslscan (and modern tools like testssl.sh) help find misconfigurations, while Wireshark verifies if sensitive data is exposed.

👉 Do you want me to create a **Challenge Lab where you use Wireshark + sslscan to catch credentials over HTTP** so you can simulate this vulnerability?