Got it 👍 Let’s prepare **precise notes for “Testing for Clickjacking” (WSTG-CLNT-09)**.

**📝 Testing for Clickjacking**

**🎯 Purpose**

To check if the application can be **framed inside another website**, tricking users into clicking hidden or disguised elements (buttons, links, confirmations).

* Used for **UI redress attacks** (e.g., force victim to transfer money, change settings).

**⚡ Common Attack Scenarios**

1. **Invisible Frame Attack**
   * Attacker loads target in <iframe> and places fake buttons on top.
   * Victim thinks they’re clicking attacker’s UI → actually clicking sensitive actions (e.g., “Delete Account”).
2. **Likejacking / Social Media Abuse**
   * Victim tricked into “liking” or “sharing” attacker content.
3. **File Upload / Settings Change**
   * Hidden buttons used to make victim enable attacker-controlled settings.

**🔍 How to Test**

**1. Frame Busting Test**

Create a malicious page with:

<html>

<body>

<h1>Click me!</h1>

<iframe src="https://target.com/sensitive-action"

style="opacity:0; position:absolute; top:0; left:0; width:100%; height:100%;">

</iframe>

</body>

</html>

* If the target loads → vulnerable.
* If it refuses to load (blank or blocked) → protections exist.

**2. Check Response Headers**

* Look for **anti-clickjacking headers**:
  + X-Frame-Options: DENY → cannot be framed.
  + X-Frame-Options: SAMEORIGIN → only framed by same domain.
  + Content-Security-Policy: frame-ancestors 'none' → modern & preferred.

**3. Dynamic Frame Busting**

* Some apps use JavaScript like:
* if (top !== self) top.location = self.location;

→ Test if this can be bypassed by sandboxed iframes or disabled JS.

**🛡️ Mitigation**

* Use **CSP frame-ancestors** (preferred):
* Content-Security-Policy: frame-ancestors 'none';
* Or fallback with:
* X-Frame-Options: DENY
* For UI-rich apps needing embedding (e.g., widgets):
  + Use **frame busting + CSRF tokens**.
  + Require **user interaction confirmation** for sensitive actions.

✅ **Key Takeaway:**  
Clickjacking is a **UI deception attack**.  
Prevent it with **X-Frame-Options or CSP frame-ancestors**, never rely solely on JavaScript.

👉 Do you want me to also give you a **ready-to-use clickjacking PoC HTML template** (with transparency overlay + button lure) so you can test apps instantly?