🧬 Capsule Manifest: PKCE Integrity Audit

📜 Capsule Name

capsule.oauth.pkce.integrity.v1

🧪 Threat Scenario

Authorization Code Interception Attack Simulates an attacker intercepting an OAuth authorization code and attempting to redeem it without the correct code_verifier.

🧾 Expected Behavior

- Server rejects token exchange if code_verifier is missing or incorrect
- Server enforces S256 challenge method
- Server validates redirect URI and state parameter
- Replay logs show failure for intercepted code redemption attempts

🧰 Capsule Payloads

```
                    Legitimate Flow{
  "code_challenge_method": "S256",
  "code_verifier": "secure-random-string",
  "code_challenge": "hashed-verifier",
  "state": "session-token",
  "redirect_uri": "https://client.example.com/callback"
} Intercepted Flow (Attacker Simulation){
  "code_challenge_method": "S256",
  "code_verifier": "null-or-wrong-string",
  "code_challenge": "replayed-challenge",
  "state": "missing-or-forged",
  "redirect_uri": "https://malicious.example.com/callback"
}
```

🛡️ Replay Logic

- Run both payloads against the target OAuth server
- Capture response codes, error messages, and replay logs
- Anchor results to civic ledger for audit and federation sync

🧭 Scroll Metadata

- Author: Protocol Genesis
- Purpose: Verify PKCE enforcement and OAuth flow integrity
- Replayable: Yes
- Federation Ready: Yes
- Anchor Schema: capsule.oauth.pkce.integrity.v1.anchor.json