Peer Review Report - G12 - Secure-Programming.zip

Introduction

Reviewed by running the app and examining the source code for security paths and protocol compliance. Environmental: WSL Ubuntu 22.04, Python 3.13, ripgrep, and manual inspection of server.py, keys.py, and datavault.py.
PoC validation performed using the provided scripts under quarantine/. Ethical note: All testing performed in an isolated VM environment; no data exfiltration or out-of-scope actions conducted.

Strengths

- Protocol coverage matches SOCP 1.3 core flows.
- Separation: crypto / DB / server logic is clear.

Vulnerabilities Identified

[Vulnerability #1 – Weak RSA Keys Acceptance (High)]

- **Where**: keys.py : load_or_create_keys()

```
if os.getenv("BACKDOOR_WEAK_KEYS") == "1":
    _generate_rsa(1024)
```

- **Exploit**: Running with BACKDOOR_WEAK_KEYS=1 makes the server or client generate 1024-bit RSA keys advertised as RSA-4096.
- **Impact**: Attacker can factor keys or replace them, breaking encryption and signatures.
- **Fix**: Remove this path; enforce RSA-4096 as per SOCP §4; validate and reject keys smaller than 4096 bits.

[Vulnerability #2 – Unsigned USER_ADVERTISE Acceptance (High)]

- **Where**: server.py : 948-969

```
if BACKDOOR_TRUST_GOSSIP:
    print("[BACKDOOR] accepting unsigned USER_ADVERTISE …")
```

- **Exploit**: Setting BACKDOOR_TRUST_GOSSIP=1 allows unsigned or invalidly signed adverts to be trusted.

- **Impact**: Server can be tricked into trusting fake peers, poisoning routing and impersonating users
- (Intentional backdoors per SOCP §16 – for demonstration, not accidental flaws.)
- **Fix**: Always require valid signatures; delete the bypass and add tests rejecting unsigned adverts.

Security Observations

- **Protocol:** Bootstrap + gossip implemented; advert verify only bypassed under backdoor.
- **Crypto:** OAEP/PSS used correctly by default; weakened by the gates.
- **AuthN/Z:** Directory model present; no sessions beyond key identity.
- **Integrity:** Transport sigs present; bypass only under backdoor flag.

Code Quality Feedback

- **Readability:** Functional, but comments sparse on the “why”.
- **Errors/Logging:** Adequate; consider structured logs for security events.
- **Testing:** No visible unit/CI tests. Add coverage for key-import checks, sig verify, replay.

Usability Feedback

- **Docs:** Multiple READMEs create confusion; consolidate into one README.txt as the authoritative document (SOCP requires ASCII format).
- **Run:** Starts as documented; SQLite auto-inits.
- **Workflow:** PoC scripts are clear; note that backdoors are intentionally documented (OK for assignment).

Recommendations

- Remove `BACKDOOR_*` gates for non-vuln build; add CI test that unsigned adverts are rejected.
- Encrypt `privkey_store` at rest (AES-GCM + KDF-derived key).
- Add unit tests (pytest) for: key size checks, advert sig verify, replay/drop logic.
- Merge READMEs; include a “Security posture” section and test matrix.