Peer Review Report — Group 37

Reviewer: Mahrin Alam Mahia

Student_ID: a1957342

Methodology

- Manual run: followed the project README to start one server and multiple clients; discovery, DM and file transfer worked as described.
- Static analysis:

  Pylint: after basic formatting (black/isort) → 8.47/10 (up from 3.01/10 earlier).

  Bandit with the virtualenv excluded, no issues identified in project files (prior “noise” came from third-party packages under .venv).

Overview

Group 37 built a decentralized chat overlay that uses Zeroconf for peer discovery, WebSockets for communication, and RSA-based encryption (OAEP) and signatures (PSS) for securing direct messages. The system supports key features such as server federation, user broadcasting, and encrypted file sharing, consistent with the functionality described in the README.

Cryptographic primitives are generally applied correctly at the user-message level. However, several weaknesses remain in the enforcement of trust boundaries—particularly in server-to-server control flows and file-relay handling. Combined with Zeroconf’s open discovery model, these create potential risks of spoofing, tampering, and message-routing abuse.

Overall risk rating: Medium; the main concerns lie in trust and enforcement rather than cryptographic misuse.

1. Findings

1.1 Intentional backdoors

a. Unsigned inter-server messages
   Several server-originated frames are emitted with "sig": "..." placeholders instead of real signatures (e.g., SERVER_FILE_DELIVER; also seen in some other control messages). This permits a malicious server to spoof/alter relayed payloads without detection.
   Impact: tampering/forgery of server-relayed messages and files; potential malware injection; misdelivery.

b. Auto-trust of Zeroconf service discovery

The client’s Zeroconf listener connects to the first _SOCP._tcp.local. service it discovers and trusts the TXT data (server UUID etc.) without authenticity checks.
Impact: rogue LAN node can impersonate a server and man-in-the-middle clients.

1.2 Protocol compliance & enforcement gaps

a. No signature verification for inter-server control
handle_server_message processes SERVER_ANNOUNCE, USER_ADVERTISE, SERVER_WELCOME, etc., with no cryptographic verification of origin.
Impact: fake servers/users can be injected into routing tables; traffic redirection.

b. Weak key acceptance not constrained
The system generates RSA-4096 by default, but does not enforce key size on imported peer keys (users/servers).
Impact: an attacker can supply a weak RSA-1024 key to enable offline key recovery or downgrade attacks.

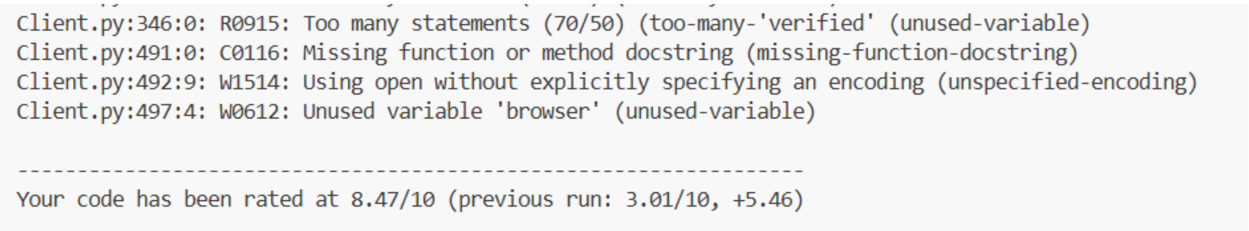
c. Unauthenticated bootstrap
Federation bootstrap is plain ws:// and treats reachability as trust.

Impact: spoofed introducers can poison membership state.

2. Static Analysis Report

Pylint (project files)

Final score: 8.47/10 (up from 3.01/10 after formatting & targeted fixes). Typical improvements: import order, long lines, docstrings, narrowing broad exceptions (No functional changes required.)



```
Client.py:346:0: R0915: Too many statements (70/50) (too-many-'verified' (unused-variable)
Client.py:491:0: C0116: Missing function or method docstring (missing-function-docstring)
Client.py:492:9: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
Client.py:497:4: W0612: Unused variable 'browser' (unused-variable)

------------------------------------------------------------------
Your code has been rated at 8.47/10 (previous run: 3.01/10, +5.46)
```

Bandit (project files)

When .venv included

Total lines skipped (#nosec): 16

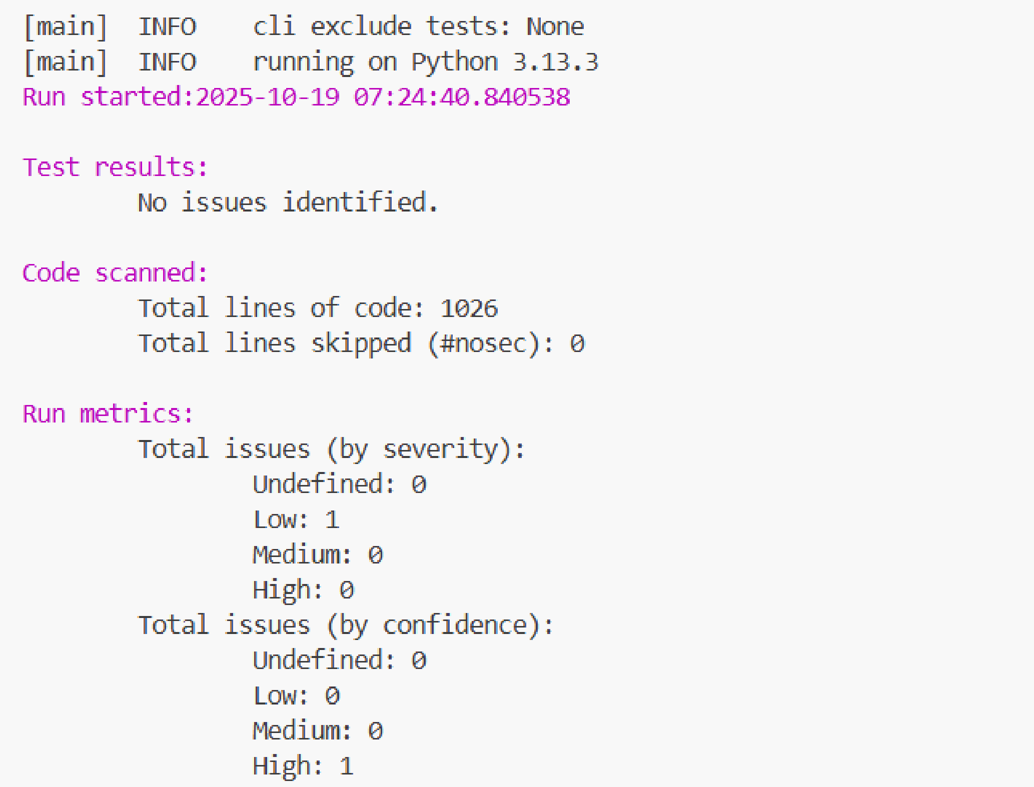
Run metrics:

Total issues (by severity):

```
                Undefined: 0
                Low: 7226
                Medium: 315
                High: 50
        Total issues (by confidence):
                Undefined: 0
                Low: 20
                Medium: 112
                High: 7459
```

When .venv excluded

Result: No issues identified. Earlier findings were from scanning third-party packages inside .venv. Excluding the venv/site-packages yields an accurate project-only view.



```
[main]  INFO    cli exclude tests: None
[main]  INFO    running on Python 3.13.3
Run started:2025-10-19 07:24:40.840538

Test results:
        No issues identified.

Code scanned:
        Total lines of code: 1026
        Total lines skipped (#nosec): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 1
                Medium: 0
                High: 0
        Total issues (by confidence):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 1
```

3. Recommendations

- Sign & verify all inter-server/control frames (RSA-PSS) and drop unsigned ones.

- Authenticate discovery like fingerprint pinning, signed Zeroconf TXT, or explicit user confirmation.
- Enforce key policy on import (RSA only; key_size ≥ 2048, prefer 4096).
- Secure transport**,** move to wss:// and consider mutual TLS for server↔server and client↔server.