Dynamic Analysis Report - Impl09

1. Execution Context

The system executed successfully under pytest, completing all unit tests.

2. File System Activity

Temporary directories were created under /tmp/pytest-of-auditor/, including multiple test folders (e.g., test_server_accepts_user0, test_client_commands0). Each test generated data directories with JSON files (users.json, sessions.json, public_channel.json, messages.ndjson) and PEM key files under keys/. File modification events indicate serialization of chat state to temporary JSON files.

Interpretation: The chat system's persistence layer uses temporary JSON stores and short-lived key pairs. No suspicious file writes (e.g., /etc or user directories) were detected.

3. System Call Trace (strace)

System call logs showed normal open/read/write/close activity within app directories. No outbound network connections (connect/sendto) were observed, confirming network isolation. No hidden process creation or unauthorized access detected.

Security note: All traced operations remained confined to /home/auditor/app and /tmp/pytest-of-auditor, demonstrating proper sandboxing.

4. Program Output (program.log)

All test modules executed successfully, validating user registration, session handling, encryption, signature verification, and cross-server messaging. The only log warning was permission-related, confirming clean runtime behavior.

5. Potential Vulnerabilities

Category	Observation	Severity
File permissions	.pytest_cache denied write access	Low
Temporary key generation	Ephemeral keys in /tmp	Low
Network I/O	No external connections observed	None

6. Overall Assessment

Behavior: Normal functional execution under isolated conditions.

Security posture: Secure — no unauthorized network access or privilege escalation attempts.

Reliability: High — all tests passed, indicating correct chat protocol behavior.

7. Recommendations

1. Maintain current sandbox permissions; pytest cache warnings are harmless.

2. Continue using temporary storage for keys during testing.

3. Include ss/netstat snapshots in future audits to confirm no socket activity.