Dynamic Analysis Report - Impl08

1. Execution Context

The updated dynamic analysis was executed using the same sandboxed container configuration as before. The command run inside the container was `python client.py`. The program displayed its usage guide:
Usage: python client.py <user_uuid_v4> <ws://host:port>
As no arguments were provided, the application exited immediately without performing its normal client-server operations.

2. Network Activity

Network logs (ss.txt) confirm there were no active TCP or UDP sockets, validating that the container's network isolation (--network=none) functioned correctly. The application did not attempt to connect to any external host or open listening ports.

3. System Call Trace (strace_log.16)

The newly captured strace log (strace_log.16) revealed similar safe behavior to the previous run. System calls primarily consisted of Python runtime initialization and library imports, such as openat(), read(), and mmap(). No socket(), connect(), or bind() calls were made, confirming zero network activity. No attempts to write to system directories (e.g., /etc, /root) or execute external binaries were found. Process termination was graceful with exit code 0.

4. Program Behavior Summary

The SOCP client again performed only parameter validation and usage printing. No data files were created or modified, and no sub-processes or background threads were spawned. This indicates deterministic and predictable execution behavior with no external side effects.

5. Security Observations

Category	Observation	Severity
Execution control	Graceful exit on missing arguments	None
Network I/O	No sockets, connects, or binds observed	None
File I/O	Only library reads, no writes to disk	None
Process creation	No subprocesses spawned	None

6. Overall Assessment

Behavior: Benign — the program printed usage instructions and exited safely.

Security posture: Secure — no network, file, or process-level risks detected.

Reliability: Consistent — results matched prior run with deterministic output.

7. Recommendations

1. To observe real communication behavior, rerun the client with valid arguments, such as:

   ```
   python client.py 123e4567-e89b-12d3-a456-426614174000 ws://127.0.0.1:9000
   ```

2. Maintain --network=none for untrusted submissions during dynamic testing.

3. No malicious indicators or unauthorized activities were detected in this version.