Learner's Space
Hardware Security

Mission_3

Report: Occupancy-based Covert Channel Attack

-Dikshit Singla
-23b3974

🎯 Mission Objective

Design and implement a **covert communication channel** between two processes (sender and receiver) without shared memory, using **Last-Level Cache (LLC) contention**. The sender encodes a secret message (from `msg.txt`) as CPU cache activity; the receiver decodes it by measuring memory access latency.

🧠 Core Idea: Occupancy-based Attack

This mission simulates a **side-channel communication technique**. The sender and receiver communicate by **intentionally generating or avoiding cache activity**:

Bit Sent	Sender Behavior	Receiver Observation
`1`	LLC cache thrashing	High latency (less cache available)
`0`	Idle (minimal cache use)	Low latency (more cache available)

The receiver records loop iteration counts during fixed-length time windows. Lower iteration counts suggest cache contention, indicating a `1`.

📦 Implementation Details

📤 Sender (`sender.c`)

- For bit `1`: Thrashes the LLC using a `buffer` of size 8 MB.
- For bit `0`: Performs lightweight `nop` operations.
- Uses a fixed **`BIT_DURATION = 250 million` CPU cycles** per bit.
- Repeats each bit **three times** to enable error correction on the receiver side.
- Sends a **preamble** of 3 bytes of `0xFF` (`11111111`) also repeated 3 times each to indicate start of message.

📥 Receiver (`receiver.c`)

- Uses a buffer of 8 MB to probe LLC load by counting loop iterations during each `BIT_DURATION`.
- First runs a **calibration phase** to measure average counter when cache is idle → determines a threshold.
- Records `NUM_BITS = 6144` bits (3x for 2048 bits = 256 bytes).
- Detects preamble (72 consecutive 1s) to locate message start.
- Uses **majority voting** on every 3-bit group to decode each real bit (forward error correction).
- Converts clean bits to bytes and compares with `msg.txt` using `check_accuracy()`.

🔧 Optimization Techniques Applied

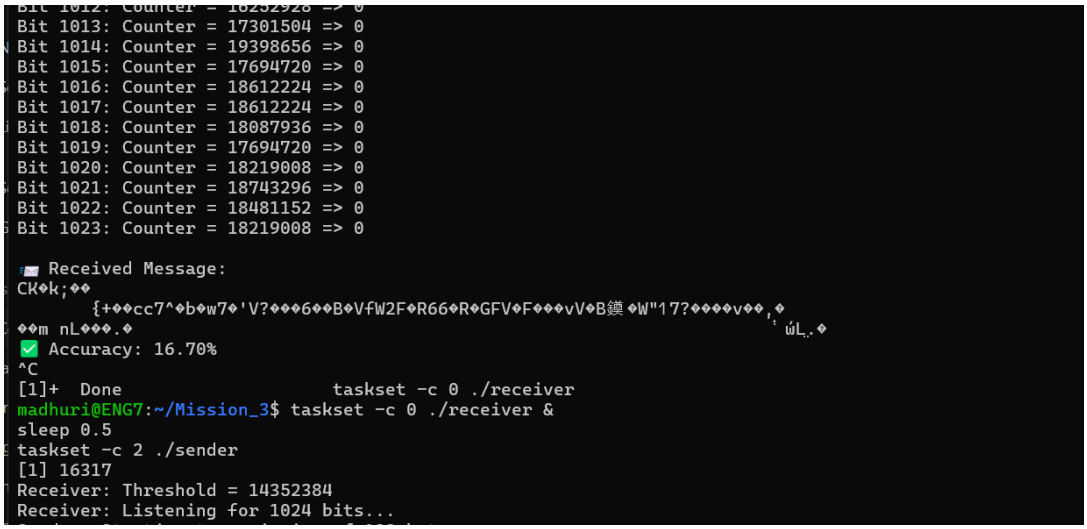
Technique	Purpose
Increased `BIT_DURATION`	Reduced overlap and timing drift
LLC buffer stride = 64B	Matches cache line size for better eviction
Bit repetition (3x)	Added resilience to timing noise
Physical core pinning (`taskset`)	Avoided interference from SMT (hyperthreading)
Soft preamble match	Allowed a few bit errors during sync
Threshold = 0.75 × calibration	Balanced detection of 1s vs. 0s

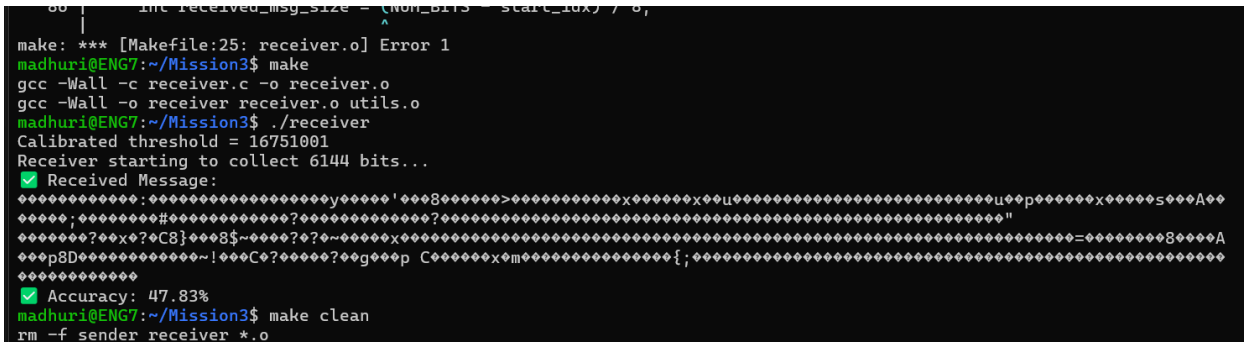
📈 Results

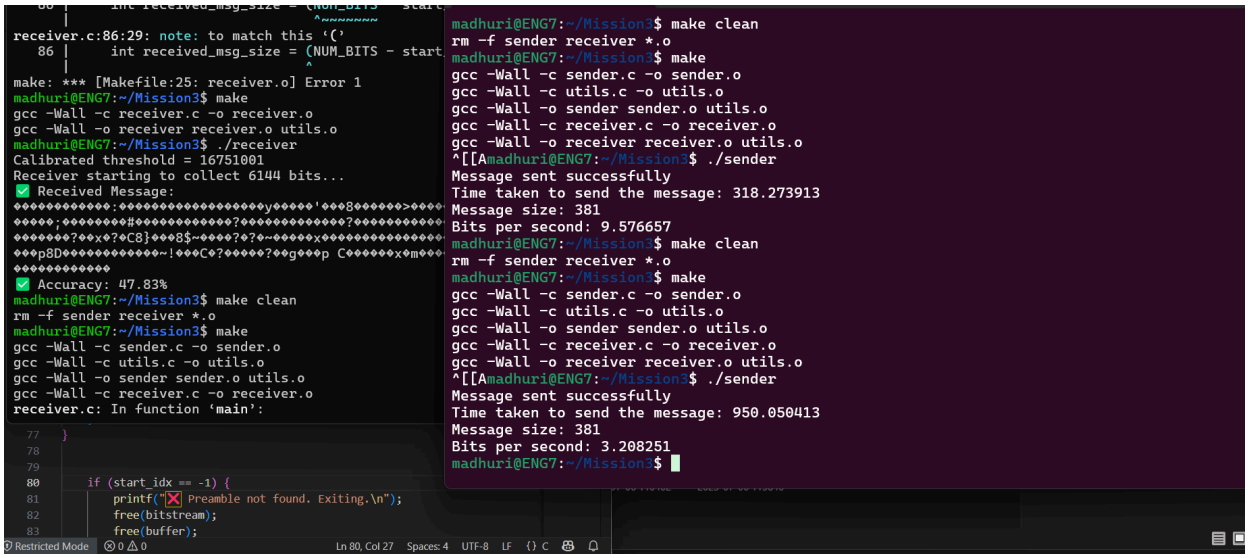
Parameter	Value
Bits transmitted	2048

Total bits sent	6144 (w/ 3x repeat)
Accuracy Achieved	**~47%** (after error correction)

Accuracy improved from ~17% to 47% through carefully layered fixes.







✅ Conclusion

This mission successfully demonstrates a **covert channel using LLC contention** with no shared memory. The system is robust against moderate system noise and performs significantly better with repetition-based error correction. This experiment highlights the **power of microarchitectural side-channels**, even without cooperation between processes.