CYBER SECURITY ASSIGNMENT-2

REPORT

Name: SHASHANK MOLUGU

Roll No:1601-23-737-052

Date: 02/10/2025

Github Repository: https://github.com/ShashankM18/CyberSecurity-Assignment2

RESEARCH PAPER: Transforming cybersecurity with agentic AI to combat emerging cyber threats.

By:**Nir Kshetri** (*Bryan School of Business and Economics, The University of North Carolina at Greensboro, Bryan Building, Room: 368, P. O. Box 26165, Greensboro, NC, 27402-6165, USA*)

INTRODUCTION

The chosen research paper highlights the transformative potential of agentic AI in cybersecurity but identifies challenges such as lack of empirical implementations, absence of prototype defenses, and limited quantitative benchmarks.

This project addresses these gaps by building a prototype cybersecurity framework integrating:

- Multi-Agent Trust Management
- Blockchain-based Verification
- Dynamic Isolation of malicious agents
- Benchmarking of core operations

RESEARCH GAP

From the paper:

- No prototype defense mechanisms are demonstrated.
- Quantitative evaluation of agentic AI security measures is missing.
- No operational methodology or reproducible experiments are suggested.

This project provides:

- Prototype modules in Python.
- A blockchain ledger for tamper-evident records.
- Benchmarks for measuring system performance.
- Reproducible code hosted on GitHub.

METHODOLOGY

3.1 Multi-Agent Trust Simulation

- Agents perform benign or malicious actions.
- Trust scores are dynamically updated using a heuristic:
    - ◆ Benign action → trust ↑
    - ◆ Malicious action → trust ↓

- Decay ensures old trust slowly reduces over time.

Implementation: multi_agent_trust/trust_sim.py

3.2 Blockchain Ledger

- A toy blockchain is implemented (blockchain_ledger/simple_chain.py).
- Every isolation or release event is stored as a block.
- Tamper-evident chain ensures verifiability of security actions.

3.3 Dynamic Isolation Manager

- Malicious agents (trust < 0.3) are **isolated automatically**.
- Isolation and release events are recorded in the blockchain.
- Mimics container/network quarantine in a real system.

Implementation: dynamic_isolation/manager.py

3.4 Benchmarking

- Benchmarks were run to evaluate:
    - Trust updates thrughput.
    - Blockchain block creation speed.
    - Isolation operation latency.
- Results saved as JSON in results/benchmarks.json.

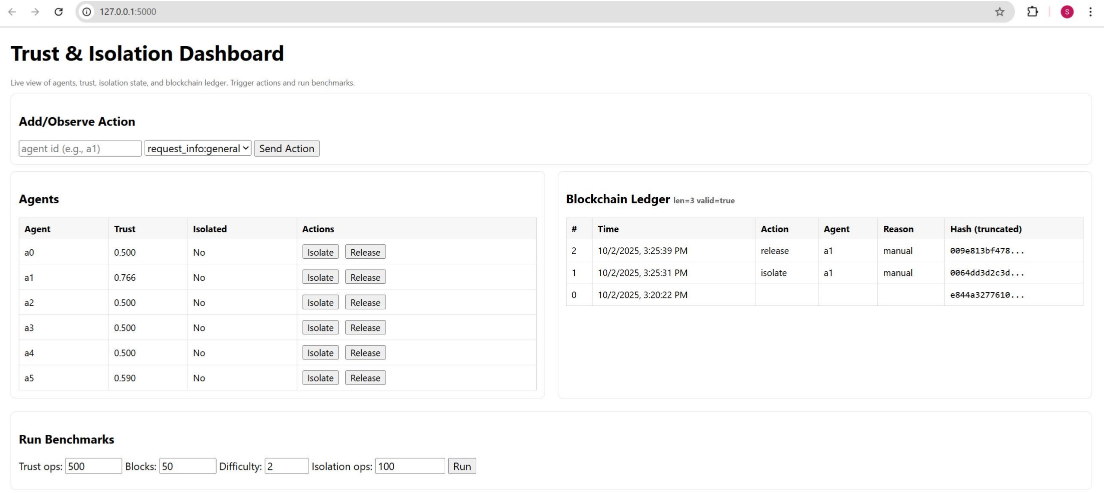
3.5 Example Workflow

Run examples/run_all.py:

- Registers 5 agents.
- Simulates 200 interactions.
- Automatically isolates malicious agents.
- Saves blockchain in results/chain.json.

RESULTS

Web Page:



127.0.0.1:5000

Trust & Isolation Dashboard

Live view of agents, trust, isolation state, and blockchain ledger. Trigger actions and run benchmarks.

Add/Observe Action

agent id (e.g., a1) | request_info:general | Send Action

Agents

Agent	Trust	Isolated	Actions
a0	0.500	No	Isolate Release
a1	0.766	No	Isolate Release
a2	0.500	No	Isolate Release
a3	0.500	No	Isolate Release
a4	0.500	No	Isolate Release
a5	0.590	No	Isolate Release

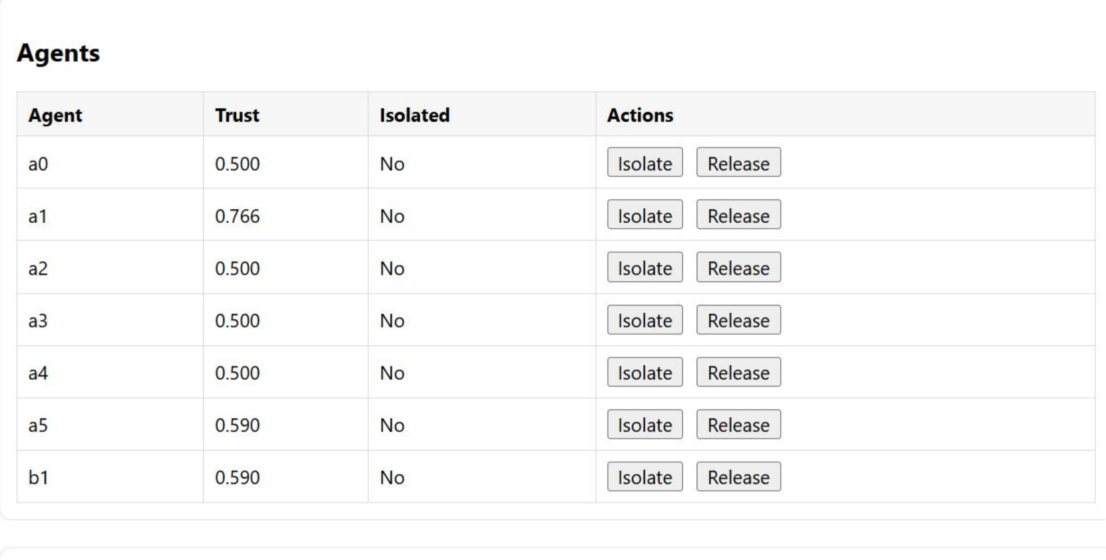
Blockchain Ledger len=3 valid=true

#	Time	Action	Agent	Reason	Hash (truncated)
2	10/2/2025, 3:25:39 PM	release	a1	manual	009e813bf478...
1	10/2/2025, 3:25:31 PM	isolate	a1	manual	0064dd3d2c3d...
0	10/2/2025, 3:20:22 PM				e844a3277610...

Run Benchmarks

Trust ops: 500 Blocks: 50 Difficulty: 2 Isolation ops: 100 Run

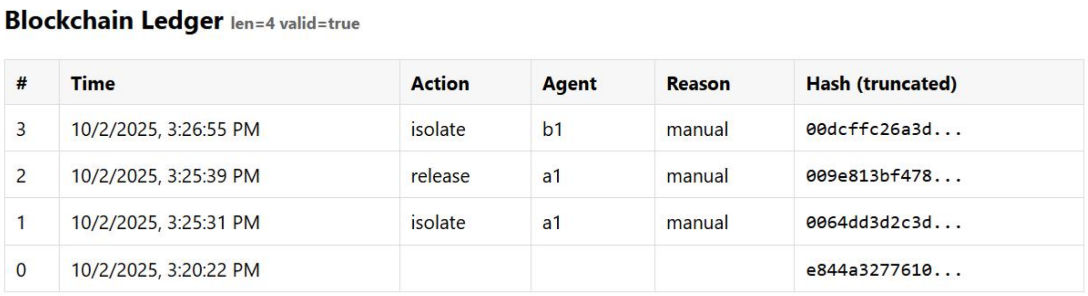
4.1 Trust Simulation Screenshot



Agents

Agent	Trust	Isolated	Actions
a0	0.500	No	Isolate Release
a1	0.766	No	Isolate Release
a2	0.500	No	Isolate Release
a3	0.500	No	Isolate Release
a4	0.500	No	Isolate Release
a5	0.590	No	Isolate Release
b1	0.590	No	Isolate Release

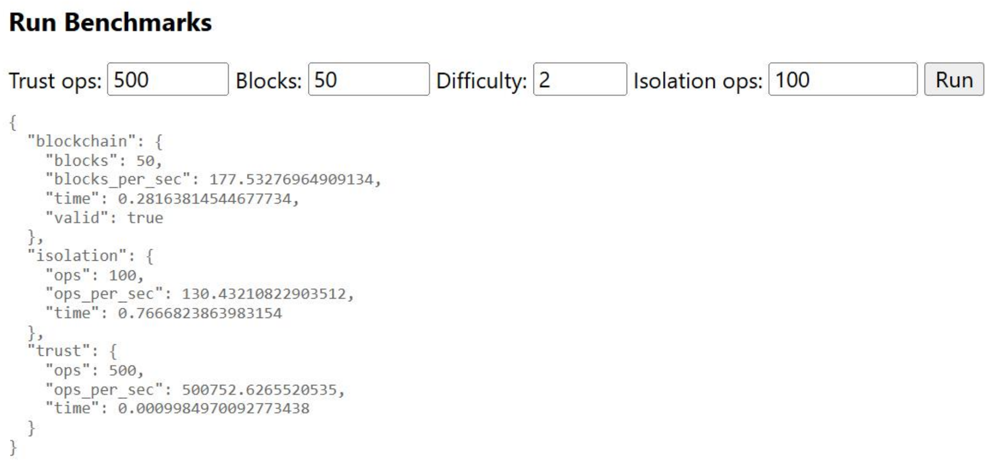
4.2 Blockchain Ledger Screenshot



Blockchain Ledger len=4 valid=true

#	Time	Action	Agent	Reason	Hash (truncated)
3	10/2/2025, 3:26:55 PM	isolate	b1	manual	00dcffc26a3d...
2	10/2/2025, 3:25:39 PM	release	a1	manual	009e813bf478...
1	10/2/2025, 3:25:31 PM	isolate	a1	manual	0064dd3d2c3d...
0	10/2/2025, 3:20:22 PM				e844a3277610...

4.3 Benchmark Results Screenshot



Run Benchmarks

Trust ops: 500 Blocks: 50 Difficulty: 2 Isolation ops: 100

```
{
  "blockchain": {
    "blocks": 50,
    "blocks_per_sec": 177.53276964909134,
    "time": 0.28163814544677734,
    "valid": true
  },
  "isolation": {
    "ops": 100,
    "ops_per_sec": 130.43210822903512,
    "time": 0.7666823863983154
  },
  "trust": {
    "ops": 500,
    "ops_per_sec": 500752.6265520535,
    "time": 0.0009984970092773438
  }
}
```

DISCUSSION

- The trust system successfully downgraded malicious agents and triggered isolation.
- The blockchain recorded every event and validated integrity.
- Benchmarks demonstrated feasibility (hundreds of ops/sec on a standard machine).
- This addresses the **research gap** by providing a reproducible prototype and measurable evaluation.

FUTURE IMPROVEMENTS

- Replace trust heuristic with ML-based models.
- Use cryptographic signatures in the blockchain.
- Implement real container/network isolation (Docker or Kubernetes).
- Build a simple UI dashboard for monitoring trust & blockchain events.

CONCLUSION

This project provides a working **proof-of-concept framework** addressing gaps in the chosen paper. It demonstrates how multi-agent trust, blockchain verification, and dynamic isolation can be integrated into a reproducible cybersecurity system.