VukZero: Zero-Trust for Autonomous LLM Agents

Vuk Pejić
v.pejic@student.tudelft.nl

Supervisors: Johan Pouwelse, Bulat Nasrulin
Thesis Committee: Johan Pouwelse, Bulat Nasrulin, Andy Zaidman

TU Delft

1 The Problem

- Autonomous LLM agents act on untrusted input while invoking real tools.
- One prompt injection → unauthorized actions, reputation abuse, host access.
- Prevention alone → no recourse after a compromise.

Key Gap:
Existing defenses stop at prevention; a decentralized setting also needs accountability and containment.

2 Research Question

How effectively can a zero-trust architecture secure autonomous LLM agents through an agent permission system, tamper-evident behavioral recording, and system-level containment?

3 Approach

A Zero-Trust Architecture (VukZero) that withholds trust at every stage of a compromise.

SQ1	How does VukZero's permission system affect the attack success rate when inserted into an external prompt-injection benchmark at the tool-execution boundary?
SQ2	How does tamper-evident behavioral recording affect reputation lag and fallout under reputation-manipulation attacks?
SQ3	How does VukZero's least-exposure and hardened container architecture affect the fallout radius of an already-compromised agent?

4 VukZero Architecture

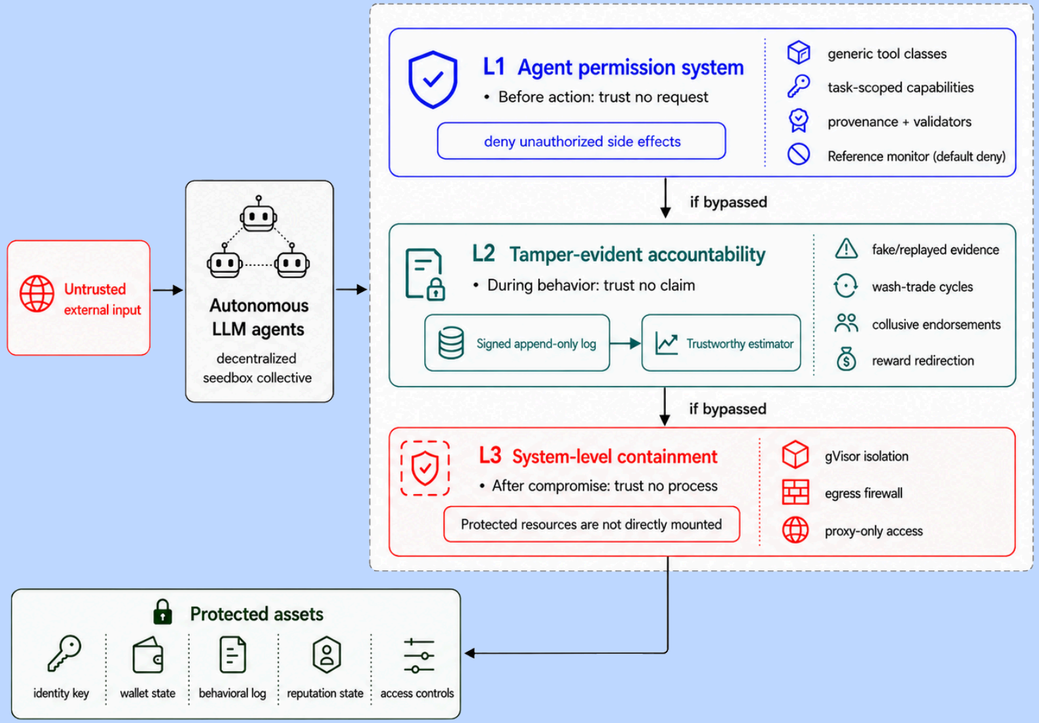

Figure 1: Three layers withhold trust across an agent compromise: L1 mediates privileged actions, L2 records behavioral evidence for expulsion, L3 containment keeps a compromised process off host assets.

5 Experimental Setup

Baselines
(Conditions VukZero is measured against)

- L1 → Undefended AgentDojo; AgentDojo with Progent enabled
- L2 → Logs missing integrity, cross-agent reconstruction, or both
- L3 → (runc, hardened runc, gVisor); least-exposure architecture on/off

L1
Does mediating every tool call cut attack success?
- Insert VukZero at the tool-execution boundary; 4 suites.

L2
Can we detect and expel reputation manipulator agents?
- 60 naive reputation-trap scenarios; 3 adaptive strategies × 40 scenarios each.

L3
Can a compromised process reach protected host assets?
- 24 hostile probes; 3×2 runtime sandboxing × architecture.

6 L1 Results

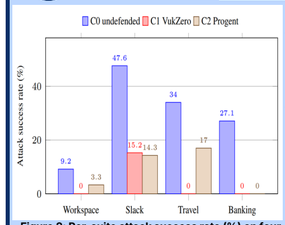

Figure 2: Per-suite attack success rate (%) on four AgentDojo suites with the tool_knowledge attack.

- VukZero cut macro-average attack success rate to 3.81% (vs. 8.66% Progent, 29.47% undefended)
- Blocked 977 unsafe tool calls in total: 635 clear violations + 342 unverified arguments (denied as a precaution)
- VukZero → Zero injection success in 3 of 4 suites

7 L2 Results

Naive Corpus

- VukZero expelled attacker in 60/60 scenarios; baseline 0/60
- VukZero reputation lag 2.67 events (baseline at 7.42+)
- Fallout broadcasts 8.42 → 3.67 (−56%)
- Fraudulent reputation gain 1.54 → 0.58 (−62%)

Adaptive Corpus

- Detection remains 100% for sybil-splitting and honest-dilution attackers.
- Optimal threshold = 6 → no honest agents wrongly expelled (13.6% false positives at threshold = 5)
- When adversary knows threshold, they evade expulsion at every threshold 1-10.

- Cross-agent reconstruction → catches collusion early (per-event rules alone miss it)
- Signed hash-chained log → tampering with evidence is always detected

8 L3 Results

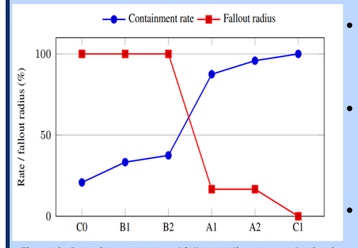

Figure 3: Containment rate and fallout radius across the 3 × 2 factorial.

- Containment climbs 31% → 94% when the least-exposure design is enabled
- gVisor and hardened containers alone contained 0 of 6 categories; strong isolation isn't enough
- Full stack closed the last gap: 24/24 probes blocked, zero fallout

9 Limitations & Conclusion

1) *Security costs utility → Suites with multi-step tasks reduce task success rate under attack (27% vs. 48%).*

2) *Threshold-aware attackers evade per-identity suspicion.*

3) *Full isolation costs latency (+228 ms for the last containment gap).*

Each layer holds against its threat; the contribution is integrating prevention, accountability, and containment so zero-trust holds throughout and after a compromise.