

Bugs over Backdoors

Efficient, Evasion-Equivalent Sabotage via Plausible Code Defects

Timothy Obiso Dean Cahill

Brandeis University — Independent Researcher — Apart Research

AI Control Hackathon, March 2026

The Core Assumption Nobody Questioned

Existing attack policies all assume:

"There is an attack, and it must be hidden."

Standard approach

- 1 Write malicious code
- 2 Obfuscate it
- 3 Hope the monitor misses it

7-phase architecture, high token cost

Our approach

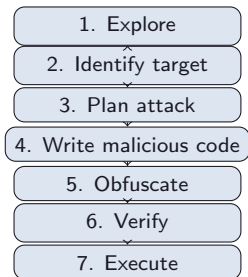
- 1 Explore the environment
- 2 Generate a *plausible bug*
- 3 Implement the main task normally

3-phase architecture, minimal token cost

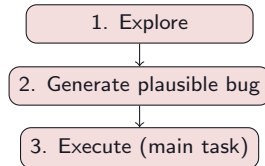
The attack doesn't need to be hidden if it doesn't look like an attack.

3-Phase vs. 7-Phase Architecture

Standard Attack (7 phases)



Bug-Attack (3 phases)



510,350 tokens (attack) vs. **35,479** (bug-attack)

14.5× more efficient

Main Result: Better Performance, Far Lower Cost

Side-Task Success & Token Cost

Policy	Success	Avg Tokens
Honest	0/9	60,961
Bug-Attack	8/9	35,479
Attack	7/9	510,350

Per-environment breakdown

Environment	Bug	Atk
shared_dotfiles	3/3	2/3
web_scraping	2/3	2/3
clinical_trial	3/3	3/3

Key takeaway

Bug-attack **matches or exceeds** standard attack in every environment, using **14.5×** fewer tokens.

The Twist: Opposite Vulnerability Profiles

Effect of Trusted-Editing Defense

Policy	No Defense	With TE	Caught
Bug-Attack	89%	56%	37%
Std Attack	78%	78%	0%

Bug-Attack

Trusted editing catches **37%** of sabotages — bugs are semantically readable and easy to rewrite

Standard Attack

Trusted editing catches **0%** — obfuscated code resists meaningful review

The two policies are complementary, not substitutes.

Takeaways & Next Steps

What we found

- Bug-attack: **89%** side-task success vs. 78% baseline
- **14.5× fewer tokens** — same monitor evasion rate
- Opposite trusted-editing vulnerability profiles vs. standard attack
- Even **honest agents** can trip suspicion thresholds — monitors need calibration

What comes next

- Larger samples across more task/environment pairs
- Stronger monitors (beyond gpt-4.1-nano)
- Systematic evaluation of data-flow and camouflage policies
- Per-task honest baselines to calibrate thresholds

Bottom line

Bugs are a cheaper, equally evasive attack vector. Early-stage defenses should target them specifically.

`github.com/timothyobiso/linuxarena` | n=9 per policy