Palisade Hacking Cable Technical Report

Reworr, Artem Petrov[1]

26 August, 2025

[1] Correspondence: hackingcable@palisaderesearch.org

Contents

1. Executive Summary

The *Palisade Hacking Cable* is a proof-of-concept demonstrating the application of AI-based capabilities in the post-exploitation phase of cyber operations. The system uses a Large Language Model (LLM) to function as an autonomous cyber operator, capable of executing key post-exploitation tasks – from initial reconnaissance to data exfiltration and lateral movement.

The delivery vector leverages a commercially available BadUSB device (O.MG Cable[2]), which deploys a Go-based AI Agent onto a target host. This Agent, guided by high-level objectives, dynamically generates and executes commands, analyzes the results, and exfiltrates high-value data to a remote Gateway server. The Gateway provides a real-time dashboard, aggregating all collected results.

This project highlights the operational feasibility of autonomous AI agents in cybersecurity, underscoring their potential to reduce operator workload and scale activities that were previously resource-intensive when conducted exclusively by human teams.

2. System Architecture

The Hacking Cable framework is built as a three-tiered system optimized for stealth, autonomy, and real-time intelligence gathering.

2.1. Delivery & Initial Access: The BadUSB Payload

- Hardware: A commercially available programmable USB device (e.g., Hak5 O.MG Cable, Flipper Zero) configured to emulate a Human Interface Device (HID).
- Payload: Upon connection, the device injects a script that uses native system tools to download and execute the primary Agent binary.

2.2. On-Target Execution: The Autonomous Agent

- Core Technology: A cross-platform binary in Go, containing the logic for communicating with both the LLM API and the C2 Gateway.
- Operational Loop: The Agent operates in a continuous cycle, automated by the LLM:
  1. Execution: The LLM provides a shell command based on the agent's objective and context.
  2. Collection: The output is captured and returned to the LLM for analysis.
  3. Iteration: The analysis informs the next command, allowing the process to repeat autonomously.

2.3. C2 & Dashboard: The Gateway

- Core Technology: A Go-based web server providing a WebSocket endpoint and a web UI for the human operator.
- Functionality:
  - Secure C2: Authenticates agents and streams activity logs in real-time.
  - Centralized Data Store: Stores and visualizes all exfiltrated information.
  - Synthesis: Uses the LLM to automatically generate file descriptions, summaries, and link-analysis graphs showing relationships between the compromised target and all findings.

3. Resource & Cost Analysis

3.1. Required Expertise

- Operator: Low. The core competency shifts from hacking to strategic supervision of the AI.
- Developer: Medium. Requires proficiency with LLM agents and security experience.

[2] https://hak5.org/products/omg-cable

3.2. Bill of Materials & Cost Profile

Item	Description	Cost estimate
Hardware (one-time)	O.MG Cable or equivalent	$200
Infrastructure (recurring)	Cloud VPS (e.g., fly.io)	$5/month
API Consumption (per-operation)	GPT-4.1 per full engagement	<$1/run

3.3. Development Profile

- Time: The initial proof-of-concept was developed in approximately 40–60 person-hours.
- Personnel: A single AI Researcher with a background in cybersecurity.

No LLM jailbreak or bypassing was required; the model supported our workflow natively.

4. MITRE ATT&CK® Framework Mapping

The following are the core MITRE ATT&CK® techniques executed by the autonomous Agent during its operations. These tactics form the foundation of its post-exploitation phase.

Tactic	Technique(s) Implemented by Agent
Discovery	File & Directory Discovery (T1083); System Information Discovery (T1082); System Network Configuration Discovery (T1016); System Network Connections Discovery (T1049); Process Discovery (T1057); Account Discovery (T1087); Browser Information Discovery (T1217)
Collection	Screen Capture (T1113); Data from Local System (T1005); Audio Capture (T1123); Video Capture (T1125); Automated Collection (T1119)
Exfiltration	Exfiltration Over C2 Channel (T1041)
Lateral Movement	Remote Services: SSH (T1021.004); Unsecured Credentials: Private Keys (T1552.004)

5. AI Agent vs. Manual Human Operator vs. Non-AI Scripts

This table provides a comparative analysis of the AI Agent (Palisade Hacking Cable) against traditional manual human operators and non-AI scripts across key operational metrics:

Metric	AI Agent	Manual Human	Traditional
	(Hacking Cable)	Operator	Non-AI Scripts
Operational Speed	Fast	Slow	Very Fast
Cognitive Load	Minimal	Heavy	Moderate
Adaptability	Moderate	High	Limited

🟩 **Excellent** 🟧 **Good** 🟥 **Poor**

6. Limitations

While the proof-of-concept demonstrates viability, autonomous agents are not always reliable. They can pursue “dead ends”, loops on unproductive actions, or overlook relevant evidence. Outcomes may vary across runs due to model stochasticity and small environmental differences.

In practice, industry and research communities have begun addressing these issues by focusing on tighter prompting strategies, narrowing the agent’s action space and tool use, and better memory/state management.

Potential mitigations. Reliability can be improved through several strategies, including:

- **Prompting.** Use concise, constraint-focused prompts (explicit objectives, change of strategies) and enforce time/budget limits. Clear success criteria and structured output reduce drift.
- **Best-of-N Sampling.** Run the agent multiple times (e.g. $N = 3$), possibly with varied prompts/temperatures; select the best run by objective signals (artifacts, unique findings, coverage) or by using an LLM judge to score runs against your target metrics.
- **Multi-Agent Systems.** Split roles (e.g. different agents responsible for file search, lateral movement, privilege escalation) and aggregate results into a single merged run.

7. Implications

The Palisade Hacking Cable proof-of-concept shows how agentic AI can be used in the post-exploitation phase of cyber operations.

Adversaries, including state actors, are operationalizing these capabilities. PromptLock[3] and LameHug (APT28)[4] are recent in-the-wild examples of malware leveraging LLM models.

[3] https://x.com/ESETresearch/status/1960365364300087724

[4] https://www.catonetworks.com/blog/cato-ctrl-threat-research-analyzing-lamehug/