CSAI Foundation

Cloud Security Alliance AI Safety Initiative

AI Inference Under Siege: The ComfyUI Cryptomining Botnet

GPU-Rich AI Workloads as High-Value Cryptojacking Targets

Unofficial AI-assisted Research

2026-04-08

Key Takeaways

- Censys ARC researchers discovered an active campaign in March 2026 targeting over 1,000 publicly accessible ComfyUI instances to mine cryptocurrency and construct a proxy botnet.
- Attackers exploit ComfyUI's custom node ecosystem to achieve unauthenticated remote code execution, using a purpose-built scanner that sweeps cloud infrastructure on a continuous cycle.
- CVE-2025-67303, an unprotected alternate channel vulnerability in ComfyUI-Manager (CVSS 7.5 HIGH), enables unauthenticated remote manipulation of configuration data on instances running versions prior to 3.38.
- Compromised hosts mine Monero via XMRig, which uses the CPU-optimized and ASIC-resistant RandomX algorithm exploiting the server's CPU resources, and mine Conflux via lolMiner using the server's GPU hardware, while simultaneously being enlisted in a Hysteria V2 proxy botnet, with all activity orchestrated through a Flask-based command-and-control dashboard.
- The campaign is part of a broader pattern in which AI inference tools—including OpenWebUI and Ollama—are systematically targeted due to their combination of internet exposure, GPU-rich hardware, and minimal default authentication.
- Organizations should immediately patch to ComfyUI-Manager v3.38 or later, restrict public internet access to all AI inference endpoints, and audit installed custom nodes for unauthorized or vulnerable packages.

Background

ComfyUI is a widely used open-source visual workflow interface for Stable Diffusion AI image generation—with over 107,000 GitHub stars—enabling users to construct complex image synthesis pipelines by connecting modular "nodes" that represent discrete processing steps [12]. Since its release, the platform has accumulated over 1,300 community-developed custom node packages that extend its capabilities across upscaling, face restoration, video processing, and model management [13]. These extensions are one of ComfyUI's primary attractions: they allow researchers, artists, and developers to rapidly expand

the platform's functionality without modifying the core codebase. ComfyUI-Manager, included by default in the official ComfyUI release, provides a browser-accessible interface for discovering, installing, and managing these nodes from community repositories and the official Comfy Registry.

However, this extensibility creates a substantial security surface. Each custom node introduces executable Python code that runs with the same privileges as the ComfyUI server process, and some community nodes expose additional HTTP endpoints that accept and execute arbitrary Python or shell commands. Researchers at Snyk documented in December 2024 how certain custom nodes provide attack paths from network access to full server compromise, demonstrating that even minor vulnerabilities in the node ecosystem could escalate to remote code execution (RCE) [4]. That research identified the problem as architectural: ComfyUI's custom node model grants third-party code broad execution authority with no enforcement boundary between nodes and the underlying server.

The platform's attack surface has grown as AI generative workloads migrated from developer laptops to cloud-hosted GPU instances and shared inference servers. Many organizations deploy ComfyUI on publicly reachable addresses to support collaborative use across distributed teams, API integration with downstream workflows, or demonstration environments—often without adapting the tool's default configuration, which was designed for trusted local use. This gap between deployment practice and the tool's security assumptions underpins the campaign examined in this research note.

Security Analysis

The Campaign

Censys ARC researchers discovered this active exploitation campaign in March 2026 after identifying an open directory on IP address 77.110.96[.]200—infrastructure associated with Aeza Group, a bulletproof hosting provider known for resisting law enforcement takedown requests [1]. The directory contained a previously undocumented toolset assembled specifically to compromise internet-exposed ComfyUI instances at scale, leading researchers to characterize this as an ongoing, operationally active campaign rather than isolated opportunistic exploitation.

The campaign was publicly reported on April 7, 2026, at which point Censys data identified more than 1,000 publicly accessible ComfyUI instances reachable without authentication across major cloud providers [1][2]. This figure was collected after filtering out known honeypots. The concentration of high-value GPU hardware makes each compromised host substantially more profitable to an attacker than a typical general-purpose server.

The Attack Chain

The campaign's operational backbone is a purpose-built Python scanner that continuously sweeps curated IP address ranges across AWS, GCP, Oracle Cloud, and other major cloud providers, handling hundreds of concurrent checks on a cycle of approximately every 3–4 hours [1]. When the scanner identifies an exposed ComfyUI instance, it queries which custom nodes are installed and evaluates whether any of those nodes expose unsafe functionality. Several nodes found in legitimate community repositories were identified as providing direct attack paths, including nodes from repositories attributed to Vova75Rus, filliptm, seanlynch, and ruiqutech—nodes that, while not designed for malicious purposes, expose endpoints capable of executing shell commands or arbitrary Python code without authentication [1].

When none of these exploitable nodes are present, attackers take a second pathway: leveraging CVE-2025-67303 in ComfyUI-Manager to remotely install a malicious node of the operator's choosing. CVE-2025-67303 is an unprotected alternate channel vulnerability (CWE-420) affecting ComfyUI-Manager versions prior to 3.38, with a CVSS 3.1 score of 7.5 HIGH and an exploit profile requiring no authentication, no user interaction, and only network access [3]. The flaw arises because ComfyUI-Manager's data and configuration directories were accessible through ComfyUI's web API without the access controls applied to the main application paths, allowing an unauthenticated remote attacker to read and write configuration data—including the node installation configuration used to pull and execute custom node packages [3]. The vulnerability was published by NIST on January 5, 2026, and patched in ComfyUI-Manager v3.38, with the fix also requiring ComfyUI core v0.3.76 or later to enable the System User Protection API that secures the configuration paths [3].

Once remote code execution is achieved through either pathway, the campaign deploys a dual-purpose payload. Compromised hosts are enlisted in a cryptomining operation: XMRig mines Monero (XMR) using the RandomX algorithm, which is CPU-optimized and ASIC-resistant, exploiting the server's CPU resources, while lolMiner mines Conflux (CFX) using the server's GPU hardware [1][2]. Simultaneously, a Hysteria V2 component configures the compromised server as a proxy node whose TLS traffic is disguised as connections to bing.com; each node is registered with the C2 server and its connection URI exported in bulk, indicating the operator may also monetize proxy access through resale to third parties [1]. All infected hosts are managed centrally through a Flask-based command-and-control dashboard, enabling the operator to push instructions and additional payloads to the full fleet.

The malware resists removal through several persistence mechanisms. Manipulation of the LD_PRELOAD environment variable intercepts system calls, the `chattr +i` command is applied to make critical files immutable against standard deletion, and binary copies are distributed across multiple

filesystem locations [1]. These techniques are familiar from Linux-targeted cryptomining campaigns broadly but are notable in the AI inference context, where system administrators may not anticipate rootkit-level persistence on what they perceive as a workflow tool.

The Supply Chain Dimension

This campaign's use of ComfyUI-Manager to deliver malicious nodes connects to a parallel threat that the ComfyUI community has been actively managing: the introduction of malicious packages into the custom node ecosystem itself. In early 2025, Comfy Org's security team responded to multiple supply chain incidents affecting the custom node ecosystem, including compromise of the ultralytics model management library and vulnerabilities discovered in the ComfyUI_LLMVISION node, prompting a series of security hardening measures across the platform [5]. These incidents established that threat actors were actively targeting the extension ecosystem as a delivery vehicle, motivating the security improvements that followed. A separate incident confirmed the presence of malicious content in the official Comfy Registry itself: packages published by user "lonemilk" were identified as delivery mechanisms for Akira Stealer—a modular Golang-based information-stealing malware that harvests browser credentials and cryptocurrency wallet data [6].

ComfyUI's maintainers have responded with a series of security enhancements, including restrictions on `eval` and `exec` usage within custom nodes, blocking runtime pip installation for registry packages, and introducing a remote disable mechanism in ComfyUI-Manager that allows administrators to deactivate compromised nodes without restarting the server [5][6]. These measures reflect the ongoing tension in maintaining an extensible open-source platform: the same architectural openness that enables the community ecosystem also enables malicious actors to use that ecosystem as a delivery vehicle.

A Broader Pattern

The ComfyUI campaign is one component of an accelerating trend in which AI inference tools deployed on internet-accessible infrastructure attract systematic adversarial attention. Cybernews researchers conducting a parallel investigation in January 2026 found more than 15,000 publicly accessible OpenWebUI instances, a meaningful portion running versions vulnerable to CVE-2025-63391—an authentication bypass affecting the platform's configuration endpoint [7][14]—with dozens of servers already confirmed infected with cryptomining malware and infostealers in a campaign that had persisted undetected for over a year [7]. A TechRadar report published the same month documented over 175,000 publicly exposed Ollama AI server instances susceptible to resource hijacking through unauthorized inference requests—a class of attack researchers termed "LLMjacking" [8].

These campaigns share a common logic: AI inference servers run on powerful hardware that is well-suited to both their intended AI workloads and GPU-compatible proof-of-work cryptocurrency mining such as Conflux's CFX algorithm. Monero's RandomX algorithm is specifically designed to resist ASIC hardware and runs efficiently on commodity CPUs, meaning a single compromised AI server can simultaneously direct its CPU capacity toward Monero mining via XMRig and its GPU capacity toward Conflux mining via lolMiner. The result is that adversaries who gain access to AI inference infrastructure can immediately monetize both its compute resources, achieving a higher per-host return than equivalent attacks on general-purpose servers.

Recommendations

Immediate Actions

Organizations running ComfyUI or comparable AI inference tools should act without delay on three fronts. First, every ComfyUI deployment should be updated to ComfyUI-Manager v3.38 or later, alongside ComfyUI core v0.3.76 or later, to close CVE-2025-67303 and enable the System User Protection API [3]. Second, any ComfyUI instance currently reachable from the public internet without authentication should be taken offline or placed behind a network access control immediately; the tool's default configuration assumes a trusted local network, and deploying it unmodified on a publicly routable address creates the exact exposure this campaign exploits. Third, administrators should audit all currently installed custom nodes, cross-referencing against the specific repositories identified in the Censys ARC research and removing any node that exposes shell execution or arbitrary Python evaluation endpoints that the deployment does not require [1].

Short-Term Mitigations

Within the next 30 days, organizations should implement network-level controls that restrict access to AI inference interfaces to known IP ranges or authenticated users, using a reverse proxy with TLS termination and credential enforcement as an intermediary layer. Security monitoring should be updated to alert on outbound connections to cryptocurrency mining pool endpoints, anomalous GPU utilization patterns, and network traffic consistent with Hysteria or other proxy relay protocols. Treating AI inference servers as equivalent to any other GPU-backed application server—subject to the same network segmentation, monitoring, and endpoint detection rules—is the appropriate baseline posture.

For multi-user and collaborative ComfyUI deployments, the authentication controls introduced in the v3.38 update should be enabled and node installation should be restricted to a designated administrator role. Organizations should establish a policy for vetting custom nodes before installation, including review of the node's source repository, commit history, and any community-reported anomalies. The custom node ecosystem should be treated as a third-party software supply chain requiring explicit intake evaluation, not as a trusted extension of the core tool.

Strategic Considerations

At a strategic level, the ComfyUI campaign illustrates that AI inference infrastructure must be governed as production infrastructure from a security operations perspective, regardless of whether the deployment originated as a developer or research tool. As organizations transition open-source AI tools from exploratory use to operational workloads, the security properties of those tools need to be assessed with the same rigor applied to any other internet-facing service. GPU-backed AI servers represent high-value targets whose compromise carries dual financial risk: the direct cost of unauthorized compute consumption and the potential loss of proprietary model weights, workflows, or training data that may be accessible to an attacker who achieves code execution on the inference server.

Security teams should inventory all AI inference endpoints across their environment—including ComfyUI deployments, model runners, and any other service that exposes GPU-backed compute over a network interface—and verify that each is covered by the organization's vulnerability management, access control, and anomaly detection programs. The breadth of AI infrastructure exposure documented across multiple platforms in early 2026 suggests that adversaries are systematically enumerating these services using the same search tools and scanning infrastructure they have long applied to other exposed services. Organizations that have not yet been targeted should not interpret that as evidence of an adequate security posture; the campaign's automated scanning cycle of approximately every 3–4 hours means exposure to the public internet without authentication substantially elevates targeting risk [1].

CSA Resource Alignment

The risks documented in this research note map directly to multiple CSA frameworks and research initiatives. CSA's MAESTRO framework for agentic AI threat modeling provides a structured, layer-by-layer approach to identifying trust assumptions across AI system architectures [9]. Applied to ComfyUI, MAESTRO's analysis of the tool execution layer (custom nodes accepting arbitrary code input) and the

orchestration layer (ComfyUI-Manager's configuration interface) directly surfaces the attack surface this campaign exploits. Organizations building or deploying AI workflow infrastructure can use MAESTRO to evaluate where their systems make implicit trust assumptions that could be violated by external access.

The CSA AI Controls Matrix (AICM) v1.0 provides 243 control objectives across 18 security domains, including supply chain security controls applicable to the custom node vetting problem and infrastructure security controls covering access control, network segmentation, and workload monitoring for AI environments [10]. The AICM provides a structured basis for gap analysis in organizations seeking to assess whether their AI security controls are commensurate with their AI deployment footprint—a pressing concern given that 73% of respondents to CSA's 2026 State of AI Cybersecurity survey report that AI-powered threats are already having a significant impact on their organization [11]. The campaign documented in this note—where legitimate extension mechanisms were weaponized—is precisely the supply chain scenario that the AICM's third-party component controls are designed to address.

CSA's Zero Trust guidance applies directly to the deployment pattern this campaign exploits. The default-open network posture documented across the 1,000+ publicly accessible ComfyUI instances identified by Censys [1]—where any host on the internet can reach the management interface without authentication—is the antithesis of Zero Trust's core principle that no service should be trusted by virtue of network position alone. Enforcing authentication and authorization at every AI service endpoint, regardless of where requests originate, is the structural control that would neutralize the attack chain described in this note.

Finally, CSA's 2026 State of AI Cybersecurity report, drawing on responses from over 1,500 security leaders, found that sensitive data exposure was cited as the leading concern by 61% of respondents, with 92% expressing concern about AI agents operating across their workforce [11]. The ComfyUI campaign illustrates both dimensions of this risk within a single attack chain: unauthorized compute consumption represents a direct operational impact, while an attacker with code execution on an inference server may have access to models, workflows, or data stored on or reachable from that server, depending on the deployment's internal access controls.

References

[1] The Hacker News. "Over 1,000 Exposed ComfyUI Instances Targeted in Cryptomining Botnet Campaign." The Hacker News, April 7, 2026.

[2] GBHackers Security. "ComfyUI Servers Hijacked for Cryptomining, Proxy Botnet Ops." GBHackers, April 8, 2026.

[3] NIST National Vulnerability Database. "CVE-2025-67303 Detail." NIST NVD, January 5, 2026.

[4] Snyk Labs. "Don't Get Too Comfortable: Hacking ComfyUI Through Custom Nodes." Snyk Labs, December 2024.

[5] Yoland Yan. "ComfyUI 2025 Jan Security Update." Comfy Blog, January 2025.

[6] Comfy-Org. "Upscaler-4K Malicious Node Pack Post Mortem." Comfy Blog, 2025.

[7] Cybernews. "Malicious campaign targeting vulnerable OpenWebUI servers: technical analysis." Cybernews, January 2026.

[8] TechRadar. "Over 175,000 publicly exposed Ollama AI servers discovered worldwide — so fix now." TechRadar, January 30, 2026.

[9] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA, February 6, 2025.

[10] Cloud Security Alliance. "AI Controls Matrix." CSA, 2025.

[11] Cloud Security Alliance. "The State of AI Cybersecurity 2026: Insights from 1,500+ Leaders." CSA, April 2, 2026.

[12] comfyanonymous. "ComfyUI." GitHub, 2025.

[13] Comfy Org. "ComfyUI Node Registry." Comfy Registry, 2025.

[14] NIST National Vulnerability Database. "CVE-2025-63391 Detail." NIST NVD, December 2025.

