Hive Pro

Threat Level

Amber

Hiveforce Labs

# THREAT ADVISORY

## ATTACK REPORT

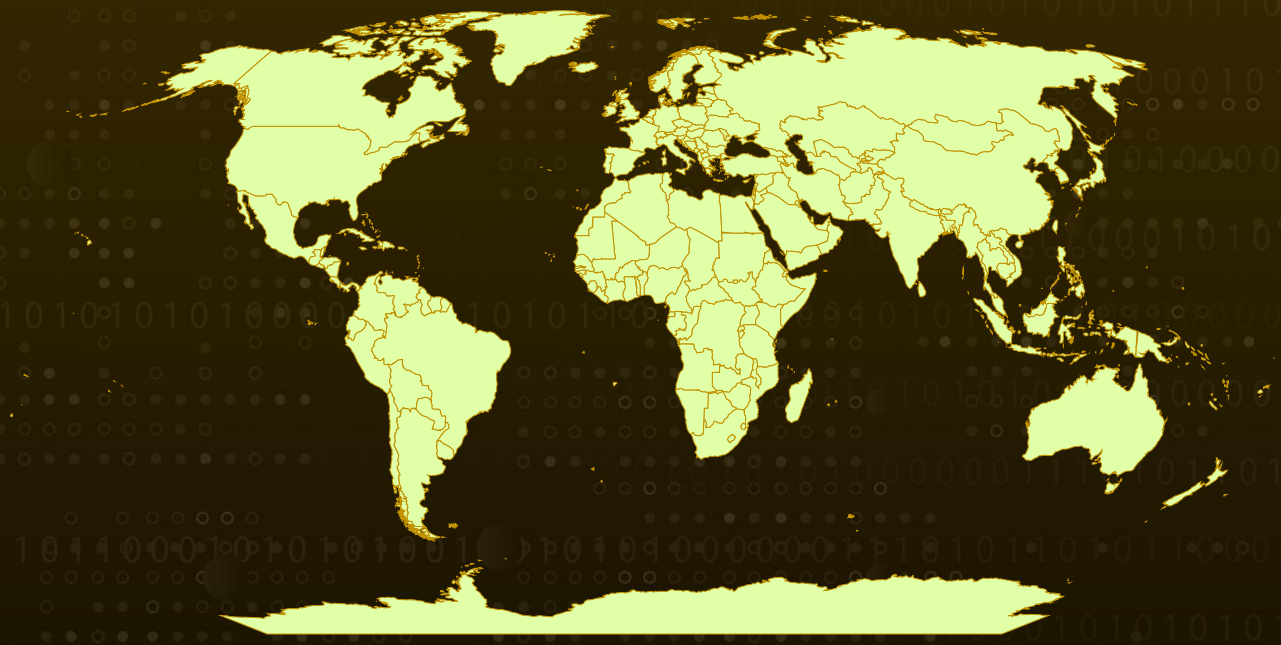
# SANSAR Abuses Exposed Ray Job Agents to Deploy Linux Cryptominers


| Date of Publication | Admiralty Code | TA Number |
| --- | --- | --- |
| November 18, 2026 | A1 | TA2026344 |

# Summary

**First Seen:** October 22, 2026
**Targeted Regions:** Worldwide
**Targeted Platforms:** Linux
**Targeted Products:** Ray, NVIDIA GPU hosts, NVIDIA Management Library (NVML)
**Threat Actor:** REF3364 (unattributed, financially motivated activity cluster; Turkish-language comments found in the rootkit source)
**Malware:** SANSAR
**Campaign:** REF3364
**Attack:** An attacker reached three Linux GPU hosts through an exposed Ray job agent that accepted job submissions without authentication, gained root, and installed a cryptomining toolkit built around SANSAR, a Rust orchestrator that takes commands over Tor and supervises CPU and GPU miners. To keep the mining out of sight, the actor replaced the NVML library with a wrapper that reports GPU utilization as 0% with invented temperature and power figures, loaded a kernel rootkit that hides selected processes, files, and socket entries, and installed a second backdoor named netns that pulls its C2 address from an Ethereum smart contract.

## Attack Regions



Powered by Bing
© Australian Bureau of Statistics, GeoNames, Microsoft, Navinfo, Open Places, OpenStreetMap, Overture Maps Fundation, TomTom, Zenrin

Targeted  Non-Targeted

# Attack Details

**#1**

The way in was a service that should never have been reachable from the internet. The activity referred to as REF3364 found that the attacker reached a Linux GPU host through an exposed Ray job agent. Ray is an open-source framework used to spread Python workloads such as AI training across many machines, and its Jobs API lets users submit commands to the cluster. On this host, the Ray dashboard was bound to the local interface, but the job agent was still reachable from outside and accepted jobs without any login. Probes showed up in the Ray logs on October 22, 2026. By October 24, a job was running nvidia-smi and sending the GPU inventory to an external server, and on October 25, the actor submitted a Python reverse shell that called back to their own listener.

**#2**

The reverse shell ran as the local account that owned the Ray session, and the actor added an SSH key to that account so the access would survive a reboot. After gaining root, the toolkit was placed in/opt/vadi_. The main component is SANSAR, a Rust program that first checks whether it is being debugged, then decodes a hidden onion address and connects to it over Tor using Arti. SANSAR starts and manages the miners, kurt (XMRig) on the CPU and tilki (SRBMiner-MULTI) on the GPU. At the same time, the actor set up persistence and concealment: they built the acpi_thermal_mgr kernel module with DKMS and added it to the boot image so it loads again at startup, swapped the real NVIDIA NVML library for a modified copy, and installed a second backdoor called netns as a systemd service.

**#3**

The toolkit ended up on three Linux GPU hosts, with the GPU miner arriving later on two of them. What SANSAR gathers is aimed at mining, not at user data: GPU names, CPU count, hostname, uptime, a count of active sessions, load averages, and the host's public IP, which it looks up by trying external IP-discovery services in order until one gives a valid answer. The rootkit then hides the activity, using ftrace to intercept directory listings, file access, process signals, and socket lists so chosen processes, files, and connections do not show up. The modified NVML library handles the hardware view, reporting GPU usage as 0% and returning made-up temperature and power values, so a quick look at nvidia-smi suggests an idle card.

**#4**

This campaign is less about stealing files and more about keeping a quiet channel open. The GPU inventory was sent to an external server on October 24, before the toolkit was installed. After that, SANSAR registers with its onion C2 using a timestamp and an HMAC-SHA256 value calculated with a key built into the binary, then exchanges custom binary messages. Sixteen message types cover heartbeats, bridge updates, interactive shells with history turned off, miner start, stop, and reconfiguration, and a task that floods a chosen address with TCP SYN packets. The netns backdoor works separately: it asks a hardcoded Ethereum smart contract for an IP address, connects to it on port 8080, and on command forks Bash out to any host and port the operator names, with no authentication on that channel. The result is a mining operation with two ways back in and very little that the owners can see.

# Recommendations

**Lock Down the Ray Job Agent:** Make sure the Ray dashboard and job submission agent are not reachable from the internet. Bind them to internal interfaces, place them behind a VPN or authenticating reverse proxy, and enforce authentication on the Jobs API so job submission is never anonymous.

**Hunt for Unexpected Ray Jobs:** Review Ray job logs for submissions you cannot tie to a known user or pipeline, especially jobs that run nvidia-smi, spawn Python reverse shells, or make outbound connections. In this campaign, the probing showed up in those logs three days before the reverse-shell job was submitted.

**Verify the NVML Library:** Compare libnvidia-ml.so.1 on every GPU host against the vendor-supplied package and look for a hidden backup file such as /usr/lib/.libnvidia-ml.so.1.backup. Host-level readings cannot settle this on their own, because the wrapper also fakes temperature and power, returning 36-46 degrees Celsius and 100-200 watts. Compare what nvidia-smi reports against the draw measured outside the operating system, at the PDU, BMC, or IPMI.

**Audit SSH Authorized Keys:** Check authorized_keys for every service and workload account, not just administrator accounts. Remove keys that no one can account for, and manage the files through configuration management so unauthorized additions are reverted automatically.

**Inspect Kernel Modules and the Boot Image:** Review loaded kernel modules, DKMS-built modules, and initramfs contents for entries like acpi_thermal_mgr that no vendor or internal team installed. Enabling kernel module signature enforcement and Secure Boot raises the bar for loading a rootkit in the first place.

**Block the Published Indicators:** Add the SHA-256 hashes, the onion address and the Ethereum contract from this advisory to endpoint and network blocklists, and hunt historically for the file paths under /opt/vadi_ and /usr/src/acpi_thermal_mgr-1.0/.

**Watch for Tor and Blockchain RPC Traffic:** GPU compute nodes rarely need to talk to Tor entry nodes, obfs4 or WebTunnel bridges, or public Ethereum RPC endpoints. Alert on such traffic from server subnets, since both C2 channels in this toolkit depend on it.

# Potential MITRE ATT&CK TTPs

| Tactic | Technique | Sub-technique |
|---|---|---|
| Initial Access | T1190: Exploit Public-Facing Application | |
| Execution | T1059: Command and Scripting Interpreter | T1059.006: Python |
| | | T1059.004: Unix Shell |
| Persistence | T1136: Create Account | T1136.001: Local Account |
| | T1098: Account Manipulation | T1098.004: SSH Authorized Keys |
| | T1543: Create or Modify System Process | T1543.002: Systemd Service |
| | T1547: Boot or Logon Autostart Execution | T1547.006: Kernel Modules and Extensions |
| Privilege Escalation | T1078: Valid Accounts | T1078.003: Local Accounts |
| | T1611: Escape to Host | |
| Lateral Movement | T1021: Remote Services | T1021.004: SSH |

| Tactic | Technique | Sub-technique |
|---|---|---|
| Stealth | T1014: Rootkit | |
| | T1622: Debugger Evasion | |
| | T1027: Obfuscated Files or Information | |
| | T1140: Deobfuscate/Decode Files or Information | |
| | T1690: Prevent Command History Logging | |
| Discovery | T1082: System Information Discovery | |
| | T1016: System Network Configuration Discovery | |
| Command and Control | T1090: Proxy | T1090.003: Multi-hop Proxy |
| | T1102: Web Service | T1102.001: Dead Drop Resolver |
| | T1095: Non-Application Layer Protocol | |
| | T1105: Ingress Tool Transfer | |
| Impact | T1496: Resource Hijacking | T1496.001: Compute Hijacking |
| | T1498: Network Denial of Service | T1498.001: Direct Network Flood |
| Defense Impairment | T1685: Disable or Modify Tools | |

# Indicators of Compromise (IOCs)

| TYPE | VALUE |
|---|---|
| SHA256 | 00c9ac44291f83c72d54302e99d155a3e26d02225a3e516764d5833a99657c9e, 96fc528ca5e7d1c2b3add5e31b8797cb126f704976c8fbeaecdbf0aa4309ad46, acc706afc9322a3eae1ef49049749e0a7c9f1a123895ac56dc8a289beb2fb5b7, cc1ad7f14777a098ccc91fd4e8f6143da14418c2b6fdf23b7f8f6b3d1b0b7f1d, 675c24728f87f7aa0fa07527027c95c24c246049c65eaef89270ab04ea50a180, 6ba1aa275e0e2826e8b48b36473539ca56cbdc41f429e539a45f8532b084d7a4, df8ab6fd699e6ae90e63e9696bae8f01342708bafc2ed88af414337e4f237194 |
| TOR Address | efxisnqldelyr5ra25oqfrsut54b34g26e7zrsiuupsoekjof2gho2id[.]onion[:]9051 |
| Ethereum Contract Address | 0x43eb291f56688c63c93e8a93c4591fc997cf6349 |
| Filename | sansar, kurt, tilki, lyrebird, libnvidia-ml.so.1, acpi_thermal_mgr.ko, netns |

# References

https://www.elastic.co/security-labs/threat-command/sansar-linux-cryptomining-malware

# What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a Demo of **HivePro**.

REPORT GENERATED ON

**November 18, 2026 • 08:30 AM**

© 2026 All Rights are Reserved by Hive Pro

More at www.hivepro.com