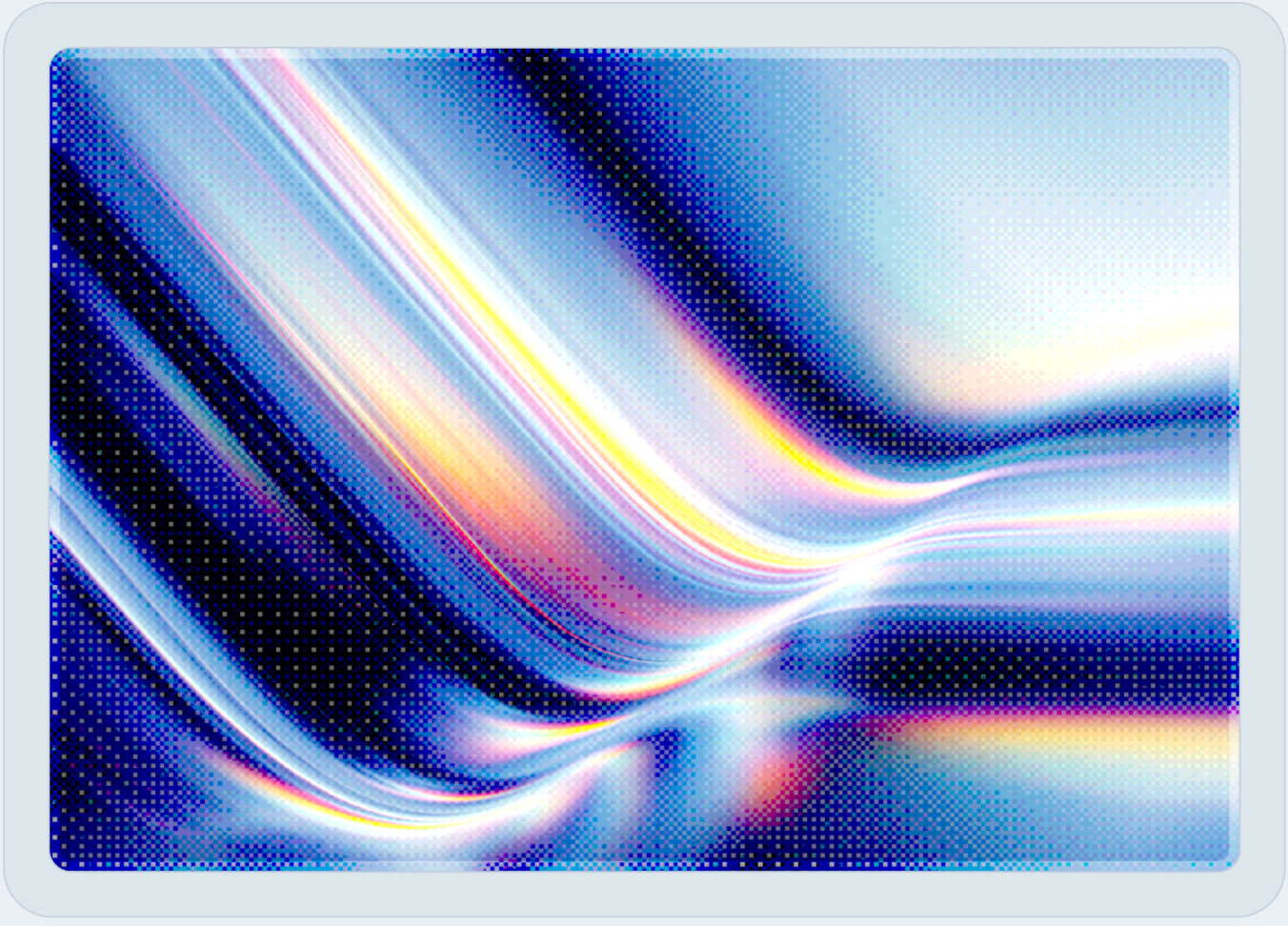
CSAI Foundation | Cloud Security Alliance

CLOSEDQUORUM: Malware That Lets AI Models Vote on Attacks

Security Implications and Guidance

2026-09-27

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

On September 22, 2026, Cisco Talos disclosed CLOSEDQUORUM, a Windows implant that queries a panel of up to four commercial large language models — DeepSeek, Qwen, Mistral, and Google Gemini — and executes whichever action wins a plurality vote among them, rather than waiting for a human operator to issue instructions [1]. Talos describes it as the first publicly documented case of malware whose tactical decision-making has been handed to a committee of AI models instead of a person, and it released the disclosure alongside CAIRN, an open-source toolkit purpose-built to hunt this emerging class of AI-integrated malware without executing suspect binaries [2].

The publicly available CLOSEDQUORUM sample is not a live, in-the-wild threat: its API credentials and Discord webhook are populated with placeholder values, and Talos has not confirmed operational deployment or completed dynamic execution of the full autonomy loop [1]. The finding arguably matters less for what it has already done than for what it demonstrates is now buildable with commodity components — a template that removes a human from a phase of the attack chain and can, in principle, continue operating after the operator stops watching. CLOSEDQUORUM extends a lineage of runtime-LLM malware that began with LAMEHUG/PROMPTSTEAL in July 2025, a trajectory CSA's AI Safety Initiative has already been tracking in its own internal research on autonomous agentic adversaries. Enterprises should treat the detection guidance in this note as a near-term priority and treat the underlying trend — AI-orchestrated malware maturing along a fast and public disclosure cadence — as a standing item for security architecture and threat-modeling programs.

Background

Talos researchers built CLOSEDQUORUM's disclosure around a project called CAIRN (Cognitive Artifact Intelligence Research Network), a metadata-first hunting framework designed to find AI-integrated malware by scanning for the fingerprints AI integration leaves behind — LLM provider endpoints, AI framework imports such as LangChain and LiteLLM, jailbreak-style system prompt strings, and local model runtime indicators — without needing to download or detonate the sample first [2]. Talos frames the period since July 2025 as an "autonomy escalation arc": the first runtime-LLM malware sample, LAMEHUG, used a single hosted model to generate reconnaissance commands on demand [3], and comparable tradecraft, including AI-analysis evasion strings, has since appeared in unrelated actor samples within about a year, indicating the technique is proliferating rather than remaining a one-off

experiment [2]. This escalation-arc framing is Talos's own, and it rests on a small number of publicly disclosed samples to date — a point worth keeping in view when weighing how quickly the trend is likely to continue. CLOSEDQUORUM, the first result CAIRN's scanning surfaced, represents the next step in that arc: instead of one model generating commands on request, four models vote on which pre-built capability to invoke next, and the malware acts autonomously on the outcome.

Cloud Security Alliance's AI Safety Initiative had already flagged this trajectory before CLOSEDQUORUM's disclosure. CSA's internal research on autonomous agentic AI adversaries catalogued documented cases in which frontier models executed the large majority of an intrusion's tactical steps with minimal human direction, and argued that the shift from AI-assisted to AI-autonomous attack execution had already crossed from theoretical to operational by mid-2026. A companion line of CSA analysis specifically tracked the runtime-LLM malware family that CLOSEDQUORUM now extends, including PROMPTFLUX's self-rewriting VBScript loader and PROMPTSTEAL's use of a hosted coding model for command generation, and warned that static indicators of compromise degrade quickly against malware whose behavior is generated fresh at each execution rather than hard-coded. CLOSEDQUORUM is consistent with both lines of analysis: it is runtime-LLM malware in the PROMPTSTEAL lineage, but it adds a governance layer — a vote among models constrained to a fixed menu of actions — that neither line of prior CSA analysis specifically anticipated.

The technical premise is narrower than the framing might suggest. CLOSEDQUORUM does not ask its model panel to invent new attack techniques; it constrains every model's output to a typed JSON schema with a `decision` field that must map to one of four pre-implemented capabilities, and any response that does not conform is discarded [1]. The system prompt instructs each model to act as "an advanced malware strategist" and to "provide ONLY executable decisions" [1], which keeps the models within pre-built capability while still ceding the choice of which capability to invoke at each step. Talos's own framing draws a useful distinction here: CLOSEDQUORUM is not evidence of AI systems developing novel offensive tradecraft, but of "effort displacement" — attackers moving the burden of tactical decision-making from themselves onto AI infrastructure they do not have to operate directly [1].

Security Analysis

CLOSEDQUORUM is a 16.4-megabyte Go-compiled Windows executable that gathers basic host context at execution — hostname, operating system architecture, administrative privilege status, and CPU count — and injects that information into a prompt sent sequentially to each configured model provider [1]. Each of the four models returns a structured choice from a fixed set of four actions: steal, inject, persist, or move, with "move" present in the code architecture but not implemented in the samples

Talos examined. The implant tallies the responses it receives and executes whichever action wins a plurality; in the event of a tie, DeepSeek holds deciding authority, followed by Qwen, Mistral, and Gemini in descending order of precedence [1]. This is majority-rule decision-making bolted onto conventional malware capability, not autonomous capability generation, and the distinction matters for how defenders should prioritize the finding: the risk is not that the malware can invent new attacks, but that it can keep selecting among known-damaging ones without an operator present, consistent with Talos's "effort displacement" framing described above.

The "steal" action, when selected, triggers three capabilities simultaneously: LSASS memory dumping to harvest Windows credentials, extraction of saved passwords from Chrome, Edge, and Firefox, and a search for MetaMask, Exodus, and other cryptocurrency wallet artifacts [1]. The "inject" action performs process hollowing or APC injection into a suspended process, and "persist" establishes multiple persistence mechanisms, consistent with conventional commodity malware behavior rather than novel technique. What differs from earlier runtime-LLM malware is the orchestration layer sitting above these capabilities: rather than a single model generating a command string once, four independent model calls are made per decision cycle, at what Talos describes as randomized five-to-fifteen-minute intervals, with results exfiltrated to the operator via a Discord webhook [1].

Talos's attribution work found six SHA256 hashes spanning what it describes as a seven-day build chain, indicating the developer iterated the implant over roughly a week before the sample Talos obtained was published or leaked [1]. Artifacts embedded in the binary were used to connect the developer to postings on carding-focused criminal forums dating back to 2025, though Talos stops short of naming the developer or confirming any completed intrusion using the tool [1]. The publicly circulating build is best understood as a template or proof-of-concept: every LLM API key defaults to the literal string `dummy_api_key`, and the Discord webhook defaults to `dummy_webhook_url`, which is consistent with a distribution model in which an operator compiles a customized, credentialed build for actual use rather than running the public sample directly [1]. No confirmed real-world deployment has been documented as of this writing.

The detection implications follow directly from the architecture. Talos recommends that defenders focus on behavioral co-occurrence rather than static indicators, since domain names, hashes, and API endpoints in this class of malware are unstable and easily rotated across builds [1][2]. The specific pattern Talos flags is a single Windows process making correlated, near-simultaneous outbound requests to multiple distinct AI model providers, combined in the same process with credential-theft or injection behavior such as LSASS access, process injection into suspended processes, or WMI-based persistence creation. Any one of those AI provider calls in isolation is unremarkable — plenty of legitimate developer tools and enterprise applications call DeepSeek, Mistral, or Gemini APIs — but Talos's premise is that few legitimate processes call several of them within a short window while also touching LSASS or creating

persistence [1]. Talos published a three-tier YARA rule structure through CAIRN covering primitive AI-integration artifacts, behavioral context, and family-level attribution, along with the six hashes from the build chain, to support hunting for this and related samples [1][2].

Recommendations

Immediate Actions

Security teams operating Windows fleets should update endpoint detection rulesets to flag any single process that generates correlated outbound connections to two or more distinct AI model provider domains (DeepSeek, Qwen/Alibaba Cloud, Mistral, Google Gemini, or similar) within a short time window, particularly when that process is not a recognized developer tool or approved AI application. This signal is likely to hold up better than static hash or domain blocking, given the malware's templated, per-operator build process. Teams should also import and run the CAIRN toolkit's YARA rules and the six published SHA256 hashes against endpoint and file-server telemetry, since CAIRN's metadata-first design allows scanning without executing untrusted binaries [2]. Any detection that correlates AI-provider API calls with LSASS access, process injection into suspended processes, or new WMI persistence in the same process warrants immediate triage as a potential CLOSEDQUORUM-class compromise.

Short-Term Mitigations

Organizations should inventory which internal applications and developer tools legitimately call external LLM provider APIs, so that the behavioral detection rules above can be tuned against a known baseline rather than generating unmanageable false-positive volume; this inventory work mirrors the "LLM-interaction surface" audit CSA's own internal research on autonomous AI malware recommended for the broader runtime-LLM malware family. Egress controls or DLP-aware proxies that log outbound traffic to AI provider domains by source process, rather than merely by destination, give defenders the process-level correlation Talos's detection guidance depends on. Security teams should also extend sandbox detonation windows for suspicious Windows binaries beyond the brief intervals common in automated analysis pipelines, since CLOSEDQUORUM's polling cycle operates on a five-to-fifteen-minute cadence that shorter sandbox runs may not capture.

Strategic Considerations

CLOSEDQUORUM's core significance is architectural rather than technical: it demonstrates that off-the-shelf commercial LLM APIs can be wired into a decision layer that keeps commodity malware capability operating without a human present to direct it. Enterprises should treat this as confirmation that behavioral, process-identity-based detection needs to become a standing capability rather than a one-time response to a single disclosure, since the specific models, providers, and vote thresholds in any future variant are straightforward for an attacker to change, requiring no novel technique. Threat-modeling exercises for both attacker-controlled and enterprise-operated AI agents should explicitly account for multi-model orchestration patterns, since the same "constrained menu, majority vote" architecture that CLOSEDQUORUM uses offensively is structurally similar to patterns some enterprises are adopting defensively for agent reliability. Security leaders should note that Talos's own account puts roughly fourteen months between the first runtime-LLM sample (LAMEHUG, July 2025) and the first autonomous multi-model C2 sample (CLOSEDQUORUM, September 2026); with only one interval documented, this is not yet evidence of an accelerating cadence, but it is reason enough to build recurring review of CAIRN and comparable hunting-framework output into threat intelligence operations rather than treating this disclosure as a single point-in-time event.

CSA Resource Alignment

CLOSEDQUORUM sits within a threat category CSA's AI Safety Initiative has been tracking in its own internal research on runtime-LLM malware, which has followed the PROMPTFLUX, PROMPTSTEAL, PromptLock, and PromptSpy family as the leading edge of malware that regenerates its behavior at runtime using a hosted or local model rather than executing a fixed payload. That internal analysis argued for reweighting detection toward behavioral and process-tree signals, since static indicators of compromise degrade quickly against this malware family. CLOSEDQUORUM extends that pattern, but the multi-model voting and quorum mechanism it introduces is a genuinely new architectural element that the prior internal analysis did not specifically anticipate; the guidance in this note reflects that refinement rather than a simple restatement of earlier recommendations. Inventorying the enterprise's LLM-interaction surface, auditing model-API egress by source process, and reweighting detection engineering toward behavior over static indicators remain the right starting point, extended here to explicitly cover correlated calls across multiple distinct AI providers rather than a single model.

CSA's public MAESTRO agentic threat-modeling framework provides the structure for extending this analysis to adversary-operated multi-agent systems. MAESTRO's layered approach to reasoning about agentic AI risk, including agent-to-agent coordination and decision arbitration, applies directly to CLOSEDQUORUM's "constrained menu, majority vote" design, which is structurally similar to patterns

some enterprises are adopting defensively for agent reliability; organizations applying MAESTRO to their own agentic AI deployments should extend that modeling to account for adversary-operated decision loops of this kind. CSA's AI Controls Matrix v1.1 similarly provides the control baseline enterprises should use to evaluate their own posture: organizations without mature AICM v1.1 coverage of the domains governing threat detection, logging, and monitoring should treat this disclosure as additional justification for closing that gap, since those controls directly govern the behavioral logging and detection-engineering practices this note recommends.

References

[1] Cisco Talos. "The Closed Quorum: Inside the first reported autonomous AI C2 implant." Cisco Talos Blog, September 22, 2026.

[2] Cisco Talos. "Introducing CAIRN: Frontier tracking for AI-integrated malware." Cisco Talos Blog, September 22, 2026.

[3] The Hacker News. "CERT-UA Discovers LAMEHUG Malware Linked to APT28, Using LLM for Phishing Campaign." The Hacker News, July 2025.