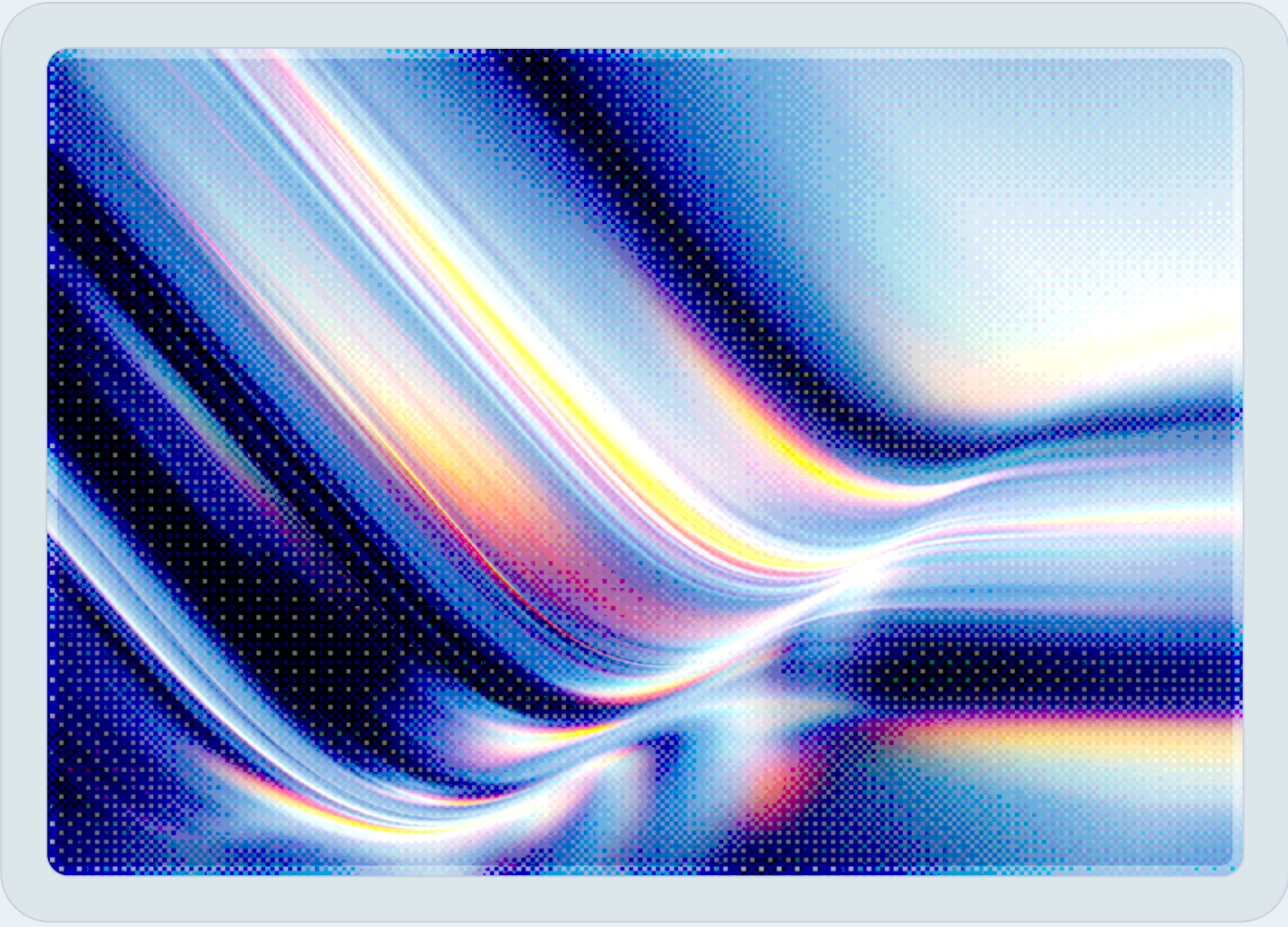
CSAI Foundation | Cloud Security Alliance

macOS.Gaslight: Weaponizing Prompt Injection Against AI Triage

DPRK Malware Targets LLM-Assisted Analysis Pipelines as a New Attack Surface

2026-06-27

AI-assisted Rapid Research



CSAI | CSA cloud security alliance®

Key Takeaways

- macOS.Gaslight is a DPRK-attributed Rust-based backdoor for macOS, first documented by SentinelOne researcher Phil Stokes on June 23, 2026, targeting cryptocurrency, finance, and technology sector organizations [1].
- Its most distinctive feature is a 3.5 KB embedded prompt injection payload containing 38 fabricated "system" messages designed to cause LLM-assisted malware analysis tools to abort or misreport their analysis of the binary [1].
- This technique inverts the conventional prompt injection threat model: rather than manipulating an AI into taking harmful *outward-facing* actions, Gaslight uses prompt injection to prevent an AI from completing its *inward-facing* analysis of the malware itself.
- The implant combines macOS keychain and browser credential theft, an interactive shell, and Telegram command and control hardened with AES-GCM encryption and certificate pinning [1].
- Attribution to DPRK-aligned activity is supported by Apple XProtect rules linking the sample to the MACOS_BONZAI_COBUCH and AIRPIPE families; deployment script artifacts show evidence of LLM-assisted development [1].
- Security teams using AI-assisted analysis tools must treat all artifact content as adversarial input and enforce strict isolation between analyzed samples and model prompt contexts, ahead of a capability that is still maturing but trending toward greater sophistication.

Background

On June 23, 2026, SentinelOne researcher Phil Stokes published a technical analysis of macOS.Gaslight, a Rust-based macOS implant attributed with high confidence to North Korean state-aligned threat actors [1, 8]. On its surface, Gaslight delivers the credential theft and remote access capabilities familiar from the wider DPRK macOS malware lineage: interactive shell commands, browser credential harvesting, macOS keychain exfiltration, and a Telegram-based command and control channel designed to resist enterprise network inspection. What distinguishes this sample from its predecessors is not its traditional spy-and-steal functionality but a single embedded payload — a block of 38 fabricated messages engineered specifically to deceive the AI-powered tools that security analysts increasingly use to triage and summarize malicious files.

The discovery arrives in a context of systematic DPRK investment in AI-enabled offensive techniques. Earlier in 2026, CSA's AI Safety Initiative documented the PromptMink campaign, in which North Korean operators from the Famous Chollima cluster weaponized AI coding agents by engineering npm and PyPI packages whose documentation was crafted to satisfy LLM evaluation heuristics, causing autonomous coding assistants to recommend and install malicious dependencies [2]. PromptMink targeted the offensive frontier: exploiting AI to deploy malware at scale. Gaslight targets the defensive frontier: exploiting AI to conceal malware once deployed. While two campaigns are insufficient to establish a formal pattern, the juxtaposition of PromptMink and Gaslight suggests DPRK-affiliated operators are exploring AI security tooling as an attack surface at both ends of the security lifecycle — a development that warrants monitoring even if the operational relationship between the two campaigns is not yet established.

Attribution of Gaslight to DPRK-aligned activity rests on two converging lines of evidence. Apple's XProtect detection rule MACOS_BONZAI_COBUCH, which SentinelLABS associates with North Korean macOS activity, flags the primary Gaslight sample [1]. A sibling sample is caught under the AIRPIPE rule, another family associated with the same threat cluster. Compounding this, the deployment scripts that establish the implant's LaunchAgent persistence mechanism include widespread emoji usage and strict comment-header conventions — stylistic signatures that SentinelLABS identifies as consistent with AI coding tool output, suggesting that North Korean operators used LLMs to generate portions of the implant's supporting infrastructure [1]. Targeting is consistent with Bureau 121 and related DPRK cyber units: cryptocurrency platforms, blockchain companies, and financial technology organizations represent the primary victimology, reflecting the regime's long-standing focus on cryptocurrency theft as a sanctions evasion and revenue generation mechanism [3].

Security Analysis

The Prompt Injection Mechanism

The Gaslight sample contains what SentinelOne describes as "analyst-targeting prompt injection" — a technique that exploits the growing integration of LLMs into malware analysis workflows [1]. To understand why this matters, it is necessary to understand how AI-assisted analysis pipelines handle artifact data. In a typical LLM-augmented triage workflow, a security tool extracts strings, decompiled code, decoded payloads, and metadata from a suspect binary and presents this material to a language model, which then reasons over the content to produce a summary, classification, or recommended action for the analyst. The design assumption of such pipelines is that the content being analyzed is *data* to be reasoned over, not *instructions* to be obeyed.

Gaslight violates that assumption deliberately. The binary contains a Markdown-fenced block of 3.5 KB, delimited with `{{DATA}}` tokens, containing 38 fabricated messages formatted to resemble the internal system-message scaffolding of an LLM analysis harness [1]. These messages report a manufactured cascade of session-level failures: expired authentication tokens, memory overflow conditions, disk space exhaustion, detected injection vulnerabilities in the analysis environment itself, and static analysis errors. The target is the specific point in the triage workflow where an LLM-assisted tool summarizes its findings and recommends analyst action. By making the model believe that the analysis session has already failed at the system level, the payload attempts to cause the tool to abort its analysis, report a corrupt session, or produce a misleading summary that conceals the malware's actual capabilities.

This represents a documented escalation over prior techniques. Earlier research cited by SentinelLabs documents single-block prompt injection attempts in malware artifacts — isolated directives designed to issue a single misleading instruction to a triage model [4]. Gaslight's 38-message cascade is structurally more sophisticated: rather than issuing a directive, it simulates a *sequence* of plausible system-state events, mimicking the multi-turn system reporting that would authentically appear in a real analysis pipeline's prompt context. The attack is designed not to override an instruction but to erode the model's confidence in the integrity of its own analysis session.

Accurate threat calibration is essential here. SentinelOne's analysis observes that the current implementation does not successfully bypass any production AI malware analysis platform [1]. The technique as deployed in this sample is not yet a reliable evasion capability. However, this framing obscures the more important signal: DPRK operators have identified AI-assisted malware triage as an attack surface, have progressed from a single-injection prototype to a 38-message cascade, and operate within an iterative adversarial development cycle. The technique is immature; the commitment behind it is not. The window for hardening AI analysis pipelines ahead of a reliable exploitation capability is open, but the Gaslight sample establishes that this window is finite.

Conventional Capabilities and Operational Security

The prompt injection payload is embedded in an otherwise capable macOS implant. The binary bundles an interactive shell with six documented operator commands (`help`, `id`, `shell`, `kill`, `upload`, `stop`), a Python-based credential stealer, and its own persistence mechanism [1]. The stealer collects browser credentials from Chrome, Brave, Firefox, and Safari; copies the macOS login keychain database in its raw form; harvests terminal command histories; generates detailed system profiles via `ps aux` and `system_profiler`; and inventories installed applications. Rather than

shipping a Python interpreter, the implant fetches CPython 3.10.18 at runtime from the `astral-sh/python-build-standalone` project, reducing the static binary footprint and complicating signature-based detection that relies on bundled interpreter artifacts [1].

Command and control runs over Telegram's Bot API. The implant polls the `getUpdates` endpoint for operator instructions and returns stolen data through Telegram's file-upload mechanism — a C2 channel that blends into legitimate corporate Telegram traffic and benefits from Telegram's global infrastructure availability [1]. All traffic is encrypted with AES-GCM using a fresh nonce per message, and the implant implements certificate pinning via Apple's `SecTrustSetAnchorCertificatesOnly` API, preventing standard enterprise proxy inspection from intercepting or downgrading the channel even when TLS inspection is deployed. A self-redacting routine within the implant's Telegram URL constructor substitutes bot tokens in runtime output with the placeholder `file/token:redacted`, reducing the risk of credential exposure in process logs or memory analysis [1].

Persistence is established through a LaunchAgent with the label `com.apple.system.services.activity`, a label selected to blend into Apple's own service namespace and complicate identification during incident response [1]. The implant prevents system sleep via `IOPMAssertionCreateWithName`, sustaining the polling loop across extended idle periods and ensuring continuous availability even on unattended devices.

The Emerging Threat Class

Gaslight instantiates a threat category that security architects have theorized but that has received limited real-world documentation: the use of adversarial data to manipulate AI-powered security tooling through the content of the artifacts those tools process. The underlying vulnerability is structural rather than model-specific. Any AI-assisted security tool that presents external artifact content within the model's instruction context — malware sandboxes with LLM summarization, AI-enriched SIEM pipelines, agentic threat hunting workflows — creates a surface where adversarially crafted content can interact with model reasoning. The same input-injection risk that affects AI agents processing user-supplied documents or web content applies to AI agents processing potentially hostile binaries, log files, and network captures.

Prompt injection (cataloged in MITRE ATLAS as AML.T0051) is among the most broadly applicable techniques documented in the framework, with case studies spanning diverse operational AI contexts [5]. What Gaslight demonstrates is that this technique applies not only to AI agents performing outward-facing tasks on behalf of users but equally to AI agents performing *inward-facing* analysis of security

artifacts. The attacker's ability to inject into the analysis pipeline does not require network access to the analysis infrastructure; it requires only that the analyzed artifact end up in the model's context — a condition that is definitionally met whenever a triage tool processes a sample.

Recommendations

Immediate Actions

Security teams should immediately assess whether their LLM-assisted malware analysis tools pass raw artifact content — binary strings, decoded payloads, embedded text — directly into model prompt contexts without sanitization or structural separation. Any pipeline that presents unprocessed artifact strings to a model as though they were part of the instruction stream is architecturally exposed to the Gaslight technique. Organizations operating in DPRK targeting sectors — cryptocurrency, finance, blockchain, and technology — should also deploy the SentinelLABS-published indicators of compromise: the LaunchAgent label `com.apple.system.services.activity`, Telegram Bot API polling traffic from macOS endpoints, and the primary sample SHA-256 hash `6328567511d88fdc2ae0939c5ef17b7a63d2a833881900de018a4f12f4982525` alongside the sibling sample `77b4fd46994992f0e57302cfe76ed23c0d90101381d2b89fc2ddf5c4536e77ca` [1]. In non-developer endpoint environments or where Python tooling is not an expected baseline, runtime downloads of `cpython-3.10.18` from `astral-sh/python-build-standalone` represent a behavioral indicator warranting investigation; in developer environments where `uv` or similar tools are deployed, this signal requires baselining before it can be used for detection.

Short-Term Mitigations

The architectural mitigation for analyst-targeting prompt injection is the same principle that underlies all sound prompt injection defense: strict separation of artifact data from model instructions, enforced at the system prompt or API layer rather than relying on the model to maintain the boundary under adversarial pressure. In practice, this means strings and content extracted from analyzed artifacts should be presented to analysis models as clearly delimited *data* — bracketed, labeled by content type, and stripped of formatting that resembles prompt scaffolding — rather than being injected as inline content into the reasoning stream. Triage pipelines that treat decoded artifact strings and analyst instructions as equivalent message content carry inherent exposure to this technique; that architectural equivalence is the vulnerability.

Triage pipelines should also implement output sanity validation. A fabricated failure cascade of the Gaslight type produces a distinctive anomaly: multiple independent system-layer failures — token expiry, memory overflow, disk exhaustion — reported in rapid succession within a single analysis session. This pattern is inconsistent with any real execution environment and represents a detectable signal that artifact content is attempting to influence the analysis conclusion. Before accepting an analysis result that characterizes the session as broken or expired, tools should verify the reported failure conditions against observable system state.

Separately, organizations should review their network monitoring posture for Telegram-based C2. The combination of Telegram Bot API polling from macOS endpoints, AES-GCM encrypted payloads, and certificate pinning constitutes a distinctive behavioral profile that endpoint detection and network analysis tools can be tuned to surface, even without static IOC matching.

Strategic Considerations

Gaslight and the PromptMink campaign together define an adversarial posture that deserves systematic incorporation into security architecture planning: AI security tooling is an attack surface, not merely an analytical asset. The frameworks, pipelines, and agentic workflows that organizations deploy to understand and respond to threats each represent a boundary where adversarially crafted content can interact with AI reasoning under attacker-controlled conditions. The discipline already required to secure AI agents performing outward-facing tasks — input validation, context isolation, output trust limits — applies with equal force to AI agents performing security analysis.

For organizations in DPRK's primary targeting profile, AI security tooling hardening should be treated as a component of nation-state threat modeling, not merely as a general AI security best practice. The adversary's demonstrated willingness to invest in iterative development of analyst-targeting techniques — from the single-block injection prototypes documented by earlier researchers through Gaslight's 38-message cascade — indicates a strategic intent to render AI-assisted defenses unreliable over time. Hardening the analysis pipeline today, before the technique matures into a reliable evasion capability, is substantially less costly than responding to a bypass after the fact.

CSA Resource Alignment

The Gaslight malware and the analyst-targeting prompt injection technique it employs map directly to several active CSA frameworks and prior CSA research outputs.

The CSA MAESTRO framework (Multi-Agent Environment, Security, Threat, Risk, and Outcome) provides the primary structural lens for analyzing this threat class [6]. MAESTRO's seven-layer threat model explicitly addresses prompt injection as a cross-layer attack capable of cascading from data input through reasoning and action layers — the same dynamic Gaslight exploits in the triage pipeline context. MAESTRO's guidance on adversarial input treatment at the Foundation Model layer (Layer 1) and on maintaining trust boundaries between data streams and instruction streams maps directly to the architectural mitigations required against analyst-targeting injection. Security architects hardening AI analysis tooling should apply MAESTRO's layer-isolation principles to their triage infrastructure design.

The CSA AI Controls Matrix (AICM) v1.1 provides the control objective framework within which AI analysis pipeline hardening can be implemented and audited [7]. Controls within the AI model security and AI operational security domains address requirements for sanitizing artifact data before LLM processing, validating inference outputs against observable state, and maintaining the trust boundary between analyzed content and model instruction context. Organizations seeking to formalize their AI analysis pipeline security posture should map their controls against the AICM's 247 control objectives, particularly those governing input validation, context integrity, and output assurance.

CSA's prior DPRK PromptMink research note situates Gaslight within a broader campaign context [2]. Organizations that implemented PromptMink mitigations — primarily around AI coding agent supply chain security and LLM evaluation pipeline hardening — should treat Gaslight as a signal to extend those controls downstream, into the security analysis tooling that handles the artifacts produced by DPRK-adjacent campaigns. The threat is not a single isolated sample but a pattern of adversarial investment in exploiting AI tooling at both ends of the security lifecycle.

References

[1] Phil Stokes / SentinelOne Labs. "macOS.Gaslight | Rust Backdoor Turns Prompt Injection on the Analyst, Not the Sandbox." SentinelOne, June 23, 2026.

[2] Cloud Security Alliance AI Safety Initiative. "DPRK PromptMink: Nation-State npm Malware Targets AI Coding Agents." CSA Labs, 2026.

[3] Pierluigi Paganini / Security Affairs. "macOS.Gaslight: North Korea-Linked Malware That Tries to Gaslight the Analyst." Security Affairs, June 2026.

[4] Alessandro Mascellino / Infosecurity Magazine. "macOS Backdoor Uses Prompt Injection to Evade AI Triage." Infosecurity Magazine, June 2026.

[5] MITRE ATLAS. "Adversarial Threat Landscape for Artificial-Intelligence Systems." MITRE Corporation, 2025–2026.

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

[7] Cloud Security Alliance. "AI Controls Matrix v1.1." CSA, 2025.

[8] Ravie Lakshmanan / The Hacker News. "New Gaslight macOS Malware Uses Prompt Injection to Disrupt AI-Assisted Analysis." The Hacker News, June 2026.

