Hive Pro

Threat Level

Amber

HiveForce Labs

THREAT ADVISORY

ATTACK REPORT

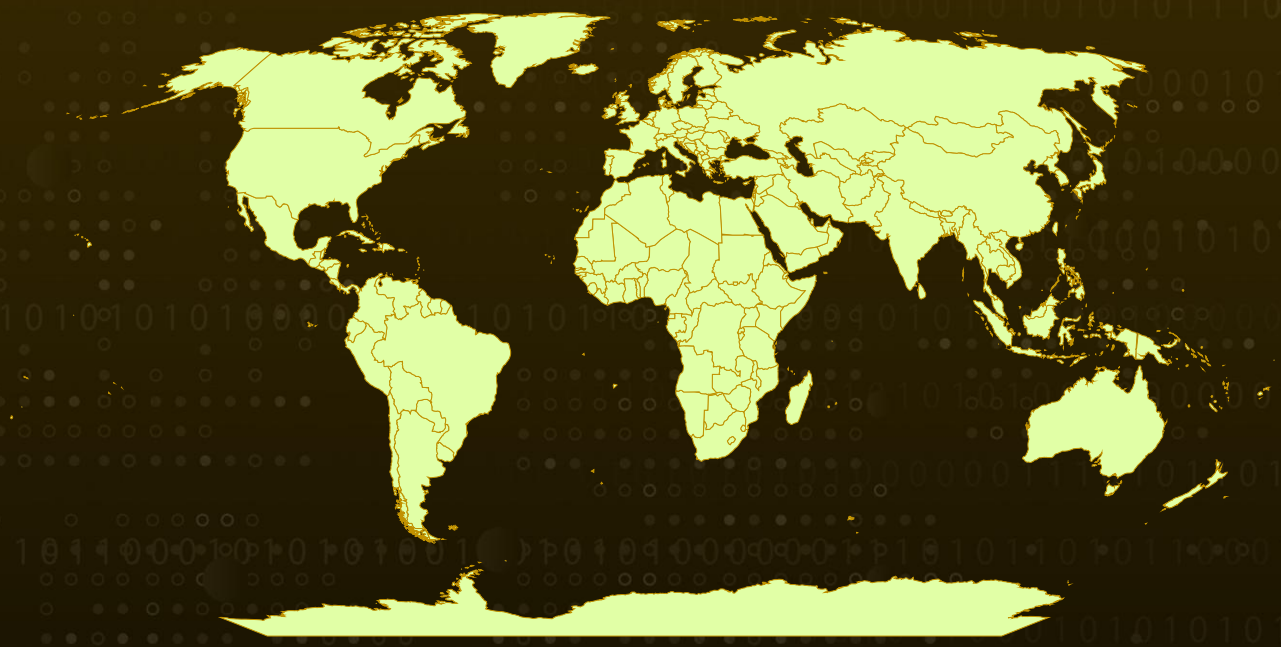
Gaslight: The Rust-Powered macOS Implant Designed to Mislead AI Tools

Date of Publication	Admiralty Code	TA Number
June 26, 2026	A1	TA2026179

Summary

First Seen: May 22, 2026
Targeted Regions: Worldwide
Targeted Platforms: macOS
Targeted Browsers: Chrome, Brave, Firefox, and Safari browser data
Malware: Gaslight (macOS.Gaslight)
Attack: Gaslight is a Rust-based macOS implant and information stealer attributed with high confidence to DPRK-aligned activity. Its defining feature is an embedded 3.5 KB Markdown-fenced payload of 38 fabricated "system" messages designed as a prompt-injection cascade that targets LLM-assisted malware triage pipelines rather than conventional sandboxes, attempting to make the analyst's AI tooling abort, truncate, or refuse analysis. The implant combines an interactive Telegram Bot API command-and-control channel hardened with AES-GCM payload encryption and certificate pinning, a self-staged Python information stealer that harvests browser data and the login keychain, a LaunchAgent masquerading as an Apple system service, and an OPSEC routine that self-redacts the operator's Telegram bot token from runtime output.

Attack Regions



Targeted Non-Targeted

Attack Details

#1

A newly identified macOS malware strain dubbed Gaslight has been engineered to mislead AI-powered malware analysis tools by embedding prompt-injection strings and fabricated debugging information directly in its executable. The sample surfaced after an Apple XProtect update flagged a Mach-O binary uploaded to VirusTotal in May 2026, with detection based solely on its file hash rather than embedded code signatures. The malware is ad hoc signed under a misleading identifier and, at the time of discovery, evaded traditional static detection engines. Apple currently detects it under the MACOS_BONZAI_COBUCH signature family, with related samples also linked to AIRPIPE, both of which have previously been associated with North Korean threat activity.

#2

The implant itself is written in Rust and employs several evasion techniques to minimize its static footprint. It dynamically resolves API calls using dlsym instead of exposing them through the symbol table and determines its own executable path at runtime rather than relying on hardcoded locations. For persistence, it deploys a LaunchAgent disguised under the label com.apple.system.services.activity, blending into Apple's legitimate namespace. It also creates power-management assertions to prevent the system from sleeping, ensuring uninterrupted command-and-control communications and long-running operations even during periods of user inactivity.

#3

The malware's behavior is heavily driven by a runtime configuration schema containing parameters for Telegram communication, encryption keys, persistence settings, and optional Python-based components. Rather than spreading laterally or escalating privileges, Gaslight focuses on post-compromise collection and operator-controlled access. Once activated, it exposes an interactive shell that supports command execution, process termination, file uploads, and other management functions, enabling operators to maintain direct control over infected systems.

#4

Data collection capabilities are extensive and rely on an embedded Python script that targets browser data from Chrome, Brave, Firefox, and Safari, as well as Terminal histories, installed applications, running processes, hardware profiles, and copies of the macOS login keychain database. A separate Bash-based installer retrieves a standalone Python runtime to support these operations across both Apple Silicon and Intel systems. The collection workflow can be selectively enabled through configuration settings, allowing operators to tailor activity to specific targets.

#5

Stolen data is compressed into an archive and exfiltrated through the same Telegram Bot API infrastructure used for command-and-control. Communications are protected with AES-GCM encryption and reinforced with custom certificate trust validation to resist network interception. Gaslight's most notable feature, however, is a large block of fabricated Markdown-formatted "system messages" designed to resemble the internal prompts of an AI analysis framework. These fake warnings, error logs, memory failures, and security alerts are deliberately crafted to confuse or derail LLM-assisted malware triage systems, representing a novel form of anti-analysis tradecraft aimed specifically at modern AI-driven security workflows.

Recommendations

Treat Sample Content as Adversarial Input in LLM-Assisted Triage Pipelines: Anyone building LLM-assisted reverse-engineering or triage tooling should treat the contents of analyzed samples as adversarial input rather than as trusted instructions, and should be prepared to keep hostile content out of the model entirely through strict input sanitization, content boundary enforcement, and prompt-scaffold isolation that prevents sample-embedded text from being interpreted as system-level messages.

Hunt for the Apple System Service Masquerade LaunchAgent: Inventory LaunchAgents across macOS endpoints and alert on any plist whose Label value matches com.apple.system.services.activity, since this label is not a legitimate Apple system service and is used by Gaslight to persist within Apple's com.apple. namespace. Extend the hunt to LaunchAgent plists whose ProgramArguments resolve to binaries outside known Apple-signed paths.

Inspect Outbound Connections to the Telegram Bot API: Profile and scrutinize outbound HTTPS connections to api.telegram.org, particularly long-lived polling sessions consistent with getUpdates and the use of the multipart attach:// file-upload mechanism, since legitimate enterprise endpoints rarely exhibit Telegram Bot API traffic patterns and Gaslight relies exclusively on this channel for command-and-control and exfiltration.

Constrain Outbound TLS Inspection Bypasses: Audit and constrain endpoints whose outbound TLS can bypass enterprise inspection through certificate pinning, recognizing that Gaslight enforces its own trust anchor via SecTrustSetAnchorCertificatesOnly to defeat proxy-based interception, so detection must shift to host-side process telemetry, DNS, and connection metadata rather than payload inspection.

Treat Power-Management Assertions on Non-Media Workloads as Suspicious: Build detections for IOPMAssertionCreateWithName calls from non-media, non-presentation processes (especially short-lived or ad hoc signed binaries), since Gaslight uses this API to prevent system sleep and sustain its long-running polling loop on idle hosts.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Execution	T1059: Command and Scripting Interpreter	T1059.004: Unix Shell
		T1059.006: Python
	T1106: Native API	
Persistence	T1543: Create or Modify System Process	T1543.001: Launch Agent
Defense Evasion	T1036: Masquerading	T1036.005: Match Legitimate Name or Location
	T1140: Deobfuscate/Decode Files or Information	
Credential Access	T1555: Credentials from Password Stores	T1555.001: Keychain
		T1555.003: Credentials from Web Browsers

Tactic	Technique	Sub-technique
Discovery	T1057: Process Discovery	
	T1082: System Information Discovery	
	T1518: Software Discovery	
Collection	T1005: Data from Local System	
	T1560: Archive Collected Data	
Command and Control	T1102: Web Service	
	T1071: Application Layer Protocol	T1071.001: Web Protocols
	T1573: Encrypted Channel	T1573.001: Symmetric Cryptography
Exfiltration	T1041: Exfiltration Over C2 Channel	

Indicators of Compromise (IOCs)

TYPE	VALUE
SHA256	6328567511d88fdc2ae0939c5ef17b7a63d2a833881900de018a4f12f4982525, 77b4fd46994992f0e57302cfe76ed23c0d90101381d2b89fc2ddf5c4536e77ca, baabf249c77bc54c54ab0e66e15af798bd28aa5b4683554456a8b73ab8741239, b3c56d689414343589f38394d19ba2fe9a518133281200faa0556ba4e4136394
Signing Identifier	endpoint-macos-aarch64-5555494492fc075f441637fb9d894913dde3a2ea
LaunchAgent Label	com.apple.system.services.activity

References

https://www.sentinelone.com/labs/macos-gaslight-rust-backdoor-turns-prompt-injection-on-the-analyst-not-the-sandbox/

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

June 26, 2026 • 09:30 AM

More at www.hivepro.com