

KuinaExtractor: Six Months of a Rust Infostealer's Evolution

 threatray.com/blog/kuinaextractor-six-months-of-a-rust-infostealers-evolution



[Contact us](#)

Key findings

- KuinaExtractor is a previously undocumented infostealer written in Rust, in active development since at least December 2025.
- We followed it across six months and four main stages by comparing the builds at the function level, linking dozens of samples into a single family.
- It collects browser data, crypto wallets and credentials for services such as Roblox, Steam and Discord, and includes a Chrome App-Bound-Encryption bypass.
- Development follows a clear line: a capable first build, a near-complete rewrite in January, a hardened production version in March, and a June rebrand to "k0to" focused on concealment.
- Two separate experiments — KuinaCookieExtractor and a "Zenith" C2 — ran in parallel and were abandoned.

Introduction

Over the past six months we tracked an undocumented infostealer written in Rust that we refer to as KuinaExtractor. Across that period it developed from an early, rough build into a hardened, full-featured stealer, with several short-lived experiments appearing alongside the main line.

We identified and grouped the builds through focusing on code similarities, allowing us to cluster dozens of samples into one family and detail the lineage. The builds appear to be the work of a single operator. The same markers recur throughout — shared mutex names, build-host paths left in the binaries, and a consistent set of Telegram contact handles — with the alias "Kuina" later replaced by "k0to". Vietnamese-language text runs through the code, including debug output, console and panel messages, and a "Thông tin hệ thống" ("System information") header in the collected-data bundle, which points to a Vietnamese-speaking developer. A C2 panel hosted in Vietnam and the inclusion of the Vietnamese CocCoc browser among the targets are weaker, supporting signals rather than firm attribution.

The evolution

December 2025 — first build

The earliest builds, from December 2025, were already capable. Alongside Roblox cookies, Steam sessions, crypto wallets and Discord tokens, they included a full Chrome v20 App-Bound-Encryption bypass that impersonates LSASS to recover the master key. Exfiltration ran over a Discord webhook, and privilege escalation used a single fodhelper/ms-settings UAC bypass. GitHub served two roles: as a content-delivery host and, through GitHub Actions, as disposable VPS/RDP infrastructure. The delivery role faded after a few months; the infrastructure role is still in use.

Analysis report: <https://reports.threatray.com/8ffaf364-d71b-408b-8738-1c4ea6c45478>

January 2026 — rewrite

In early January 2026 the stealer was rebuilt over a few days. The rewrite added substantial reconnaissance: eight WMIC hardware queries, WiFi SSID enumeration, a Windows Credential Manager dump and a routine that terminates 17 browser processes, along with victim-IP geolocation and a loop that disables Microsoft Defender. Exfiltration moved from the Discord webhook to a Telegram bot, and the single UAC bypass was replaced by a function-pointer table offering seven methods.

Analysis report: <https://reports.threatray.com/1a2ac9f2-1653-472d-8128-3e014a709fe2>

March 2026 — production hardening

By March the family had settled into a production build that stayed in use for months. The cookie-theft routine was largely unchanged — the same LSASS/ABE chain, extended to ChaCha20-Poly1305 for newer Chrome versions — but the surrounding code hardened. The UAC bypass moved to the SilentCleanup technique, browser coverage grew to around 40 (including CocCoc), and the build gained broad VM and sandbox detection. This variant is still observed today.

Analysis report: <https://reports.threatray.com/228b04ad-eb49-4203-bfca-a0f4d27e5e0c>

On 17 June a build appeared under the new name "k0to". It removes the "Kuina" name from the binary and, for the first time, focuses on concealment rather than new capabilities. It uses a self-contained HTTP stack (reqwest over hyper and rustls) that ships its own CA roots and does not rely on the system TLS store, wraps its strings in 28-byte XOR (including the Telegram C2 URL), and adds a sandbox check that scans PowerShell window titles for analyst tools. Its Telegram channel is now one-way: file upload only, with no command polling.

Analysis report: <https://reports.threatray.com/e5c2265e-a583-4f3c-9114-5a8a7dd24285>

Parallel experiments

KuinaCookieExtractor

Alongside the main stealer, the developer maintained a separate, leaner Rust codebase we call KuinaCookieExtractor during January. It reaches beyond browser cookies to Roblox and Steam sessions, Minecraft and FileZilla logins, Telegram tdata and Discord tokens, and exfiltrates over a Discord webhook rather than Telegram. Several markers tie it to the same author: the kuina build user, a custom KUINA_UAC_BYPASS_ATTEMPTED sentinel, the kuina1999 handle, and a main-line sample that references CookieExtractor's scheduled task. Its anti-analysis is lighter — on detecting a VM it only logs a warning and continues. It was visible for about two weeks before disappearing.

Analysis report: <https://reports.threatray.com/ac43f312-3296-4f06-abed-c5903ffda445>

Zenith

A second, short-lived experiment concerned C2 work. On 28 April a debug build shipped with logging left enabled, writing verbose [DEBUG] traces to %USERPROFILE%\Desktop\zenith_debug.txt, including an explicit author self-attribution block; its mutex is disguised as a network adapter name (Kuina_Intel(R) 82574L Gigabit Network Connection). A few days later, on 1 May, a "Zenith Stealer" panel appeared at 103.229.53[.]18:3000 (Vietnamese AS135918, Viet Digital Technology), with combined Telegram and HTTP exfiltration that was already push-only. It was abandoned soon after.

Analysis report: <https://reports.threatray.com/0d910065-f469-4a95-a835-810c38836fbd>

Conclusion

After six months and several experiments, KuinaExtractor remains active and under steady development. At the time of writing it is mid-transition to the "k0to" name, with recent operator activity. Following the family through code similarity made it possible to track this development across rebrands and infrastructure changes, and to separate the main line from the parallel experiments.

IOCs and YARA rules: <https://github.com/threatray/threat-research/tree/main/2026-06-25-KuinaExtractor>

More to explore...



[Intel 471 and Threatray Announce Joint Research Collaboration](#)

[Intel 471, the premier provider of cyber threat intelligence for leading intelligence, security,](#)

[December 14, 2022](#)

The screenshot shows the Intel 471 AI Analysis interface. At the top, the file path is 'c:\windows\syswow64\dbghelp.dll' with a hash of '0x6f3d0000', size of '0x18f000', and '3871 functions'. The interface has tabs for 'VERDICT DETAILS', 'CODE INTELLIGENCE', 'FUNCTIONS', 'OSINT HUNT', 'SIMILAR SAMPLES', 'CAPABILITIES', and 'AI ANALYSIS'. The 'AI ANALYSIS' tab is active, showing a verdict of 'Malicious' with a warning icon. A dropdown menu shows '2026-04-13 08:23 (29 functions)' and a 'Generate AI Analysis' button. Below this is a 'SUMMARY' section stating the binary is a sophisticated loader or backdoor. The 'CAPABILITIES' section lists several functions: 'Decryption' (XOR loop), 'Evasion' (API resolution), 'File Operations' (delete files), 'Memory Manipulation' (wildcard search), and 'System Reconnaissance' (module enumeration).

[Leveraging AI and Code Intelligence for Rapid Identification of Trojanized DLLs](#)

[How Threatray combines function-level code attribution with AI Analysis to pinpoint trojanized DLLs.](#)

[April 16, 2026](#)



Uniting for Discovery: TeamT5 and Threatray Announce Joint Research Collaboration to Unlock New Frontiers

[TeamT5 and Threatray are teaming up in a joint research collaboration to level up threat actor tracking and malware analysis.](#)

[April 23, 2025](#)