AURA Stealer: A Crude Clone of LummaC2 – Technical Analysis and Threat Breakdown

By Foresiet

Published: 2025-07-29 · Archived: 2026-04-05 22:50:23 UTC

Latest from the blog

Posted on: 29 July 2025 | Author: Foresiet

Executive Summary

AURA Stealer is a newly emerging information-stealing malware that presents itself as a streamlined alternative to more established stealer families such as LummaC2. Marketed as a carefully engineered solution, AURA is positioned by its developers as purpose-built for efficiency and results—eschewing unnecessary complexity in favor of a focused and modular design.

The stealer claims support for the extraction of credentials, session data, and autofill information from over 110 browsers and 70 applications, including cryptocurrency wallets and 2FA tools. Additionally, it is configured to target data from more than 250 browser extensions, with the flexibility to expand its coverage through customizable configuration updates—allowing threat actors to add new applications or extensions with minimal effort.

A notable technical feature includes cookie harvesting from Chromium-based browsers without terminating the browser process, reducing user disruption and avoiding cookie invalidation. The malware leverages custom shellcode for decrypting App-Bound data, with all sensitive decryption handled server-side, limiting suspicious behavior on the infected host.

AURA also includes a built-in loader for delivering secondary payloads, and the overall binary size remains lightweight at approximately 500–700 KB, thanks to a from-scratch development approach fortified by a custom morpher to evade static detection.

While AURA attempts to mirror the operational blueprint of more advanced stealers, technical analysis reveals its limited evasive capabilities and flawed implementation. Despite its marketed strengths, it remains a low-quality clone with questionable reliability in live environments.

Background and Initial Discovery

The sample was retrieved from underground forum. Tagged as Lumma, it caught our attention due to a newly surfaced blog on Telegra.ph which openly mocked it as a “low-quality LummaC2 parody.”

- **SHA256 Hash**: bac52ffc8072893ff26cdbf1df1ecbcbb1762ded80249d3c9d420f62ed0dc202

- **File Type**: PE32 executable
- **Packers**: LLVM Morphing Engine
- **Execution Method**: Process hollowing via rundll32.exe

Introduction to AURA Stealer

AURA Stealer positions itself as a modern info-stealer offering:

- Browser password harvesting
- Cryptocurrency wallet stealing
- Fingerprinting (OS, hardware, user)
- Basic evasion (anti-debug, region bypass)
- Subscription-based C2 infrastructure

However, upon deeper reverse engineering, we found numerous signs of amateur development and reused modules from LummaC2 without proper implementation.

Subscription & Panel Overview

AURA Stealer is offered under a subscription-based model, making it accessible to a wide range of threat actors with varying technical and financial capabilities. According to promotional material and screenshots of the threat actor's control panel, two subscription tiers are available. The Basic tier, priced at $2.95 per month, provides access to harvested logs, enabling actors to view and potentially resell stolen credentials and data. The Advanced tier, offered at $5.85 per month, unlocks the full functionality of the stealer's panel.

This includes capabilities such as a build generator, comprehensive statistics view, and full control over configurations allowing operators to customize payloads and monitor infection metrics with ease. The low pricing model suggests a deliberate attempt to attract a broader base of low-skilled cybercriminals looking for ready-to-use tools.

Infection Chain & Execution Flow

Upon execution, AURA Stealer initiates its infection chain by spawning a hollowed rundll32.exe process, a commonly abused Windows binary used to masquerade malicious activity under legitimate system processes. Within this process, the malware allocates a read-write-execute (RWX) memory region, where it injects custom shellcode. This shellcode is then executed to begin the core data harvesting routine.

Behavioral analysis reveals a typical execution pattern seen in commodity stealers, lacking advanced techniques such as memory encryption, API unhooking, or multi-stage loaders. Dynamic sandbox testing indicates that AURA does not implement sophisticated memory protection mechanisms or deep obfuscation layers, making it detectable by behavioral and heuristic-based solutions.

Despite using process hollowing a well-known stealth technique AURA's implementation isrudimentary, indicating minimal effort to bypass modern endpoint defenses.

Static Analysis

Static analysis of the AURA Stealer payload dumped using Scylla and disassembled in IDA uncovers several notable characteristics that reflect both the tool's simplicity and its targeted design.

The malware leverages the nlohmann::json C++ library, commonly used for lightweight JSON parsing, to handle configuration data and manage structured communication with its command-and-control (C2) infrastructure. String obfuscation is handled through classic XOR-based encryption routines, a low-effort technique that adds minimal complexity to reverse engineering but is often sufficient to evade basic static signatures.

One of the more technical evasion tactics includes resolving all API imports dynamically via LdrGetProcedureAddress, instead of the conventional GetProcAddress. This subtle deviation is aimed at bypassing import-based detection heuristics and static analysis tools that rely on import tables.

Hardcoded string artifacts discovered within the binary include process and module names such as Ollydbg.exe, Wookx.dll, and avghookx.dll, suggesting rudimentary anti-debugging and anti-analysis checks. These are likely used to detect and avoid execution in monitored or sandboxed environments.

Additionally, the binary contains regional geofiltering logic, with hardcoded ISO country codes for several CIS-region countries, including Russia (RU), Kazakhstan (KZ), and Uzbekistan (UZ). This implies a deliberate choice to avoid infecting systems within those jurisdictions, a common practice among threat actors operating from or catering to post-Soviet regions, possibly to avoid local law enforcement scrutiny.

Anti-Debugging & Anti-Analysis Techniques

AURA Stealer employs a limited but deliberate set of anti-debugging and anti-analysis techniques aimed at evading detection and frustrating manual analysis. Among the first mechanisms observed is debugger detection, specifically targeting tools such as OllyDbg, identified by checking for the presence of associated processes or window titles.

The malware also performs regional avoidance checks, skipping execution on systems located in CIS countries such as Russia, Kazakhstan, and Uzbekistan—reinforcing the hypothesis that the developers aim to avoid drawing attention from local authorities.

Another technique involves the use of the Windows API function MapFileAndCheckSumW, typically used for validating file integrity. In this context, it's repurposed as a lightweight integrity check to detect tampering or runtime manipulation, such as code injection or binary modification.

Additional anti-analysis methods include window title string checks (to detect analysis tools) and registry key lookups associated with sandbox environments or virtual machines.

While these mechanisms demonstrate basic awareness of reverse engineering environments, they lack depth and sophistication. Most of the checks can be bypassed with simple binary patching, sandbox evasion scripts, or debugger cloaking tools—underscoring AURA's relatively low barrier to analysis.

This screenshot confirms successful binary unpacking. It shows restored imports from KERNEL32.dll, including functions like ExitProcess, CreateFileW, CreateThread, and others—indicating typical stealer capabilities (file access, thread management, process control).

C2 Infrastructure & Network Activity

AURA Stealer communicates with its command-and-control (C2) infrastructure primarily over HTTP, using plaintext POST requests to exfiltrate victim data. Upon execution, the malware sends a system fingerprint to its C2 domain—**http://glossmagazine.shop/**—which typically includes OS version, browser count, installed applications, and other environment details. The C2 responds with instructions in JSON format, enabling real-time tasking such as additional payload delivery or configuration updates.

Analysis reveals that this infrastructure is reused across multiple campaigns, some of which appear to overlap with LummaC2 operations, indicating either shared infrastructure, kit reselling, or actor crossover. The consistent reuse of infrastructure suggests operational laziness or an attempt to rapidly deploy campaigns without concern for attribution risks.

Several additional domains have been observed in association with AURA-related stealer activity and panel logins:

- **softytoys[.]shop**
- **auracorp[.]cc**
- **secondhandcloth[.]shop**
- **armydevice[.]shop**
- **opencamping[.]shop**
- **glossmagazine[.]shop**

These domains serve either as C2 endpoints or as web panels for actor access to logs, build generation, and campaign management. The use of low-cost, disposable .shop and .cc domains aligns with trends seen in other stealer-as-a-service operations, reflecting a focus on low overhead and ease of replacement upon takedown or blacklisting.

AURA vs. LummaC2: Comparison

AURA attempts to mimic Lumma but fails in reliability and stealth.

Conclusion

AURA Stealer positions itself as a competitor to LummaC2, mimicking its branding, architecture, and subscription model but falls significantly short in terms of quality and sophistication. The malware suffers from poor operational security (OPSEC), basic anti-analysis measures, and weak implementation of core stealer functionalities. These flaws make it easier to reverse-engineer, detect, and mitigate, especially for organizations with mature threat detection capabilities.

Despite its shortcomings, AURA still presents a real threat to unprotected systems, particularly those lacking endpoint detection and response (EDR) solutions or relying solely on signature-based antivirus. Its low cost, ease of use, and active promotion make it appealing to low-skill threat actors looking to harvest credentials, cookies, and other sensitive information at scale.

Security teams are advised to track associated domains, monitor for typical process injection patterns (e.g., hollowed rundll32.exe), and apply behavioral detection rules tailored to info-stealer activity. Early identification and proactive blocking can prevent compromise, even from unsophisticated threats like AURA.

About us!

Foresiet is the pioneering force in digital security solutions, offering the first integrated Digital Risk Protection SaaS platform. With 24x7x365 dark web monitoring and proactive threat intelligence, Foresiet safeguards against data breaches and intellectual property theft. Our robust suite includes brand protection, takedown services, and supply chain assessment, enhancing your organization’s defense mechanisms. Attack surface management is a key component of our approach, ensuring comprehensive protection across all vulnerable points. Compliance is assured through adherence to ISO27001, NIST, GDPR, PCI, SOX, HIPAA, SAMA, CITC, and Third Party regulations. Additionally, our advanced antiphishing shield provides unparalleled protection against malicious emails. Trust Foresiet to empower your organization to navigate the digital landscape securely and confidently.
