

REVSTEALER ramps up: analysis of up-and-coming infostealer

elastic.co/security-labs/threat-command/revstealer-credential-harvesting-infostealer

September 2, 2026



[Threat Command](#)

Elastic Security Labs deep dives into REVSTEALER, an emerging infostealer targeting browsers, wallets, and gaming accounts.

Elastic Security Labs is tracking an emerging infostealer, REVSTEALER, under REF2859. This family has gained momentum in recent months, with higher distribution volume, and has emerged as a formidable threat, featuring a comprehensive credential harvester, an embedded sandbox scoring system, and a Polygon blockchain-based dead drop for resilience. Beyond credential theft, REVSTEALER targets gaming platforms for additional monetization; the developer clearly understands that victim accounts on these platforms hold real monetary value in resale markets.

Gen Threat Labs first [covered](#) REVSTEALER earlier this summer, detailing its core features. In this research publication, we will walk through how it's distributed, cover its main features, and share observed infrastructure. For readers who prefer more in-depth technical details on the malware, we are also releasing a whitepaper covering the family. We are also releasing an IDAPython string decryption script for anyone working with samples directly.

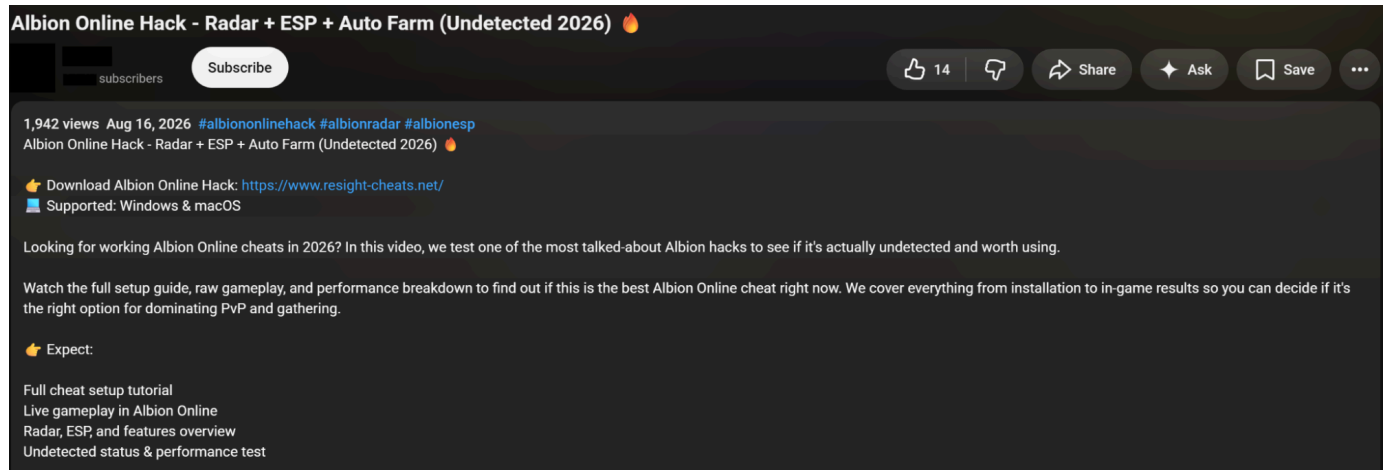
The white paper is located [here](#) and contains an in-depth analysis of REVSTEALER and REF2859.

Key takeaways

- Large volume of samples (~4700) on VirusTotal over the past year, with varied submission context
- Comprehensive stealer component targeting browsers, VPN/chat applications, password managers, gaming platforms. Includes ABE debugger-based bypass.
- Weighted anti-analysis checks (sandbox scoring system)
- EtherHiding - Polygon-based dead drop for resilient infra swapping
- Tasking delivers four modules, not previously documented publicly
- Widespread use of VMProtect packer in most REVSTEALER samples; unpacked samples display a verification prompt similar to LummaStealer and AuraStealer

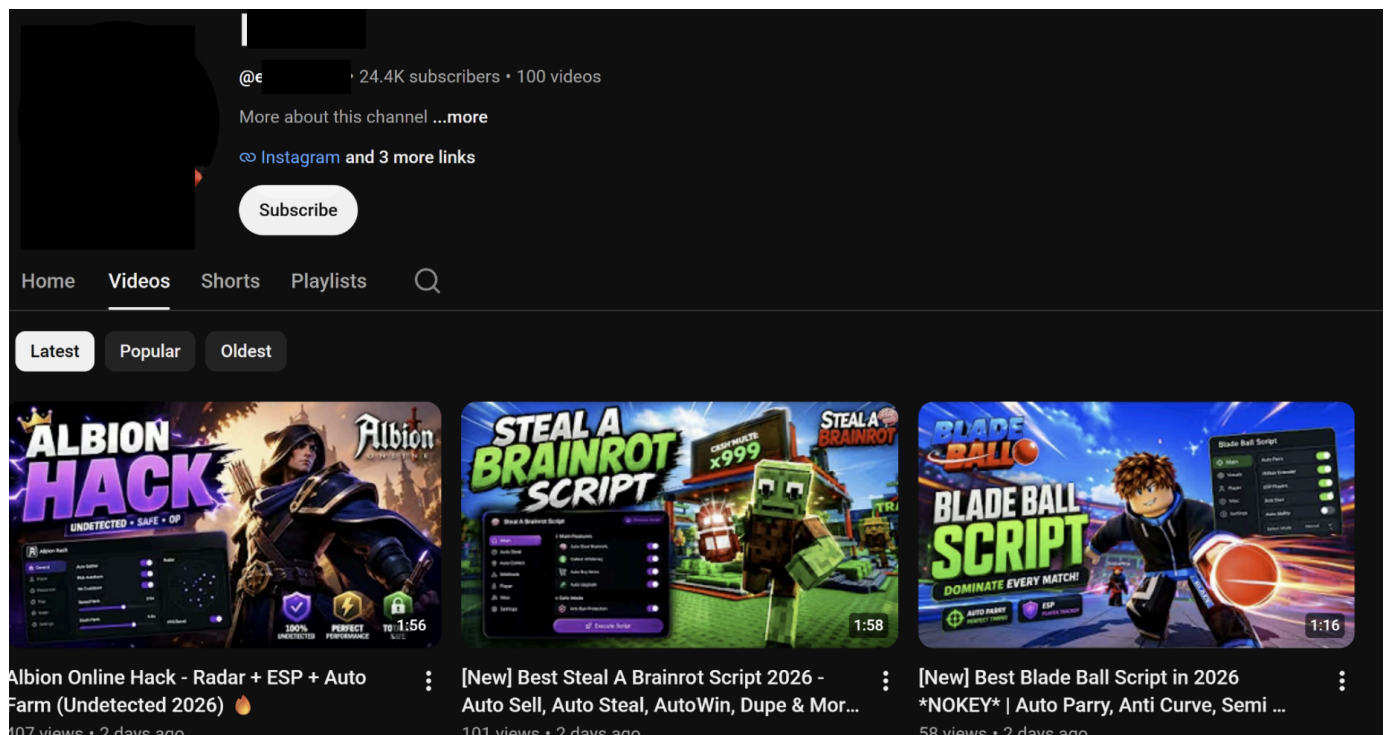
Attack vector

Gen Threat Labs documented REVSTEALER reaching victims through a social-engineering path targeting gamers by advertising free cheats and mod menus. Building on this reporting, we identified at least 17 YouTube channels promoting the two related domains elitecheatsx[.]live and resight-cheats[.]net.



Malicious link in a YouTube video description

In the campaigns we observed, this took the form of YouTube account takeovers. Existing channels with established subscriber bases were hijacked and used to post short (1–2 minute) AI-generated videos demonstrating the "cheats" and linking viewers to the malicious sites.

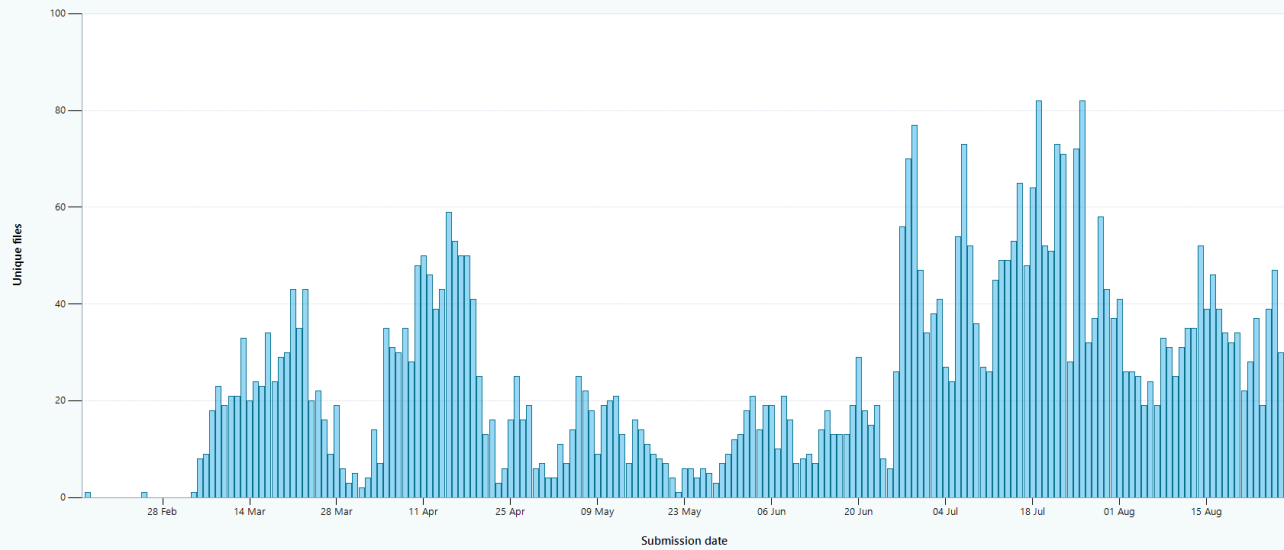


Compromised YouTube account

The game-cheat campaign is only one observed presentation. Related VirusTotal submissions used names and file metadata that impersonated unrelated, recognizable software, including Slack, qBittorrent, Steel Series GG, Blender, and many more. This breadth suggests the builds are packaged for multiple lure themes, not just gaming-focused campaigns. We also ran a Retrohunt with our YARA rule, which returned about 4,700 matching samples across VirusTotal's database over the past year.

Retrohunt Matches by Submission Date

16 Feb 2026 to 27 Aug 2026 | 4,733 unique files | UTC



Retrohunt matches visualized by submission first seen date

Many of the samples we analyzed were packed with VMProtect, but the malware code was mostly recoverable through memory dumping.

Features

Development

Across its design, REVSTEALER reflects a developer building for long-term operational use. The malware incorporates features such as self-deletion, indirect syscalls, API hashing, string encryption, mechanisms to avoid user-mode hooks, and custom exception handling.

```

.text:0000000140042E20 invoke_indirect_syscall proc near
.text:0000000140042E20
.text:0000000140042E20 var_78= byte ptr -78h
.text:0000000140042E20 arg_20= qword ptr 30h
.text:0000000140042E20 arg_28= qword ptr 38h
.text:0000000140042E20 arg_30= qword ptr 40h
.text:0000000140042E20 arg_38= qword ptr 48h
.text:0000000140042E20 arg_40= qword ptr 50h
.text:0000000140042E20 arg_48= qword ptr 58h
.text:0000000140042E20 arg_50= qword ptr 60h
.text:0000000140042E20 arg_58= qword ptr 68h
.text:0000000140042E20 arg_60= qword ptr 70h
.text:0000000140042E20 arg_68= byte ptr 78h
.text:0000000140042E20
.text:0000000140042E20 push rbp ; Central indirect-syscall dispatcher. Re-marshals
.text:0000000140042E21 mov rbp, rsp
.text:0000000140042E24 push rbx
.text:0000000140042E25 push rdi
.text:0000000140042E26 push rsi
.text:0000000140042E27 sub rsp, 80h ; Integer Subtraction
.text:0000000140042E2E mov rbx, rcx
.text:0000000140042E31 mov r10, rdx
.text:0000000140042E34 mov rdx, r8
.text:0000000140042E37 mov r8, r9
.text:0000000140042E3A mov r9, [rbp+arg_20]
.text:0000000140042E3E lea rsi, [rbp+arg_28] ; Load Effective Address
.text:0000000140042E42 lea rdi, [rsp+98h+var_78] ; Load Effective Address
.text:0000000140042E47 mov rcx, 8
.text:0000000140042E4E rep movsq ; Move Byte(s) from String to String
.text:0000000140042E51 movzx eax, word ptr [rbx+0Ch] ; Move with Zero-Extend
.text:0000000140042E55 xor ax, cs:g_syscall_index_xor_key ; Logical Exclusive OR
.text:0000000140042E5C mov r11, [rbx]
.text:0000000140042E5F xor r11, cs:g_stub_ptr_xor_key ; Logical Exclusive OR
.text:0000000140042E66 call r11 ; Indirect Call Near Procedure
.text:0000000140042E69 add rsp, 80h ; Add
.text:0000000140042E70 pop rsi
.text:0000000140042E71 pop rdi
.text:0000000140042E72 pop rbx
.text:0000000140042E73 pop rbp
.text:0000000140042E74 ret ; Return Near from Procedure
.text:0000000140042E74 invoke_indirect_syscall endp
.text:0000000140042E74

```

An indirect syscall dispatcher

Unlike most commodity stealers, REVSTEALER seems to put more effort into validating its targets; this is evident in its cryptocurrency wallet harvester, which uses a multi-layer architecture with a discovery engine followed by a collector that applies wallet-specific extension filters to extract only relevant files.

CIS exclusion checks

Before REVSTEALER qualifies the victim machine, it first checks the machine's default/system languages and keyboard layout using a custom FNV-1a hash lookup. It retrieves each value from the machine, hashes it, and compares it against an embedded table of precomputed values representing the Commonwealth of Independent States (CIS) locales. If any of these three checks match, the malware terminates.

```

UserDefaultUILanguage = wrap_GetUserDefaultUILanguage();
if ( util_short_hash_lookup(langid: UserDefaultUILanguage) != 0 )
    wrap_ExitProcess(exit_code: 0);
SystemDefaultUILanguage = wrap_GetSystemDefaultUILanguage();
if ( util_short_hash_lookup(langid: SystemDefaultUILanguage) != 0 )
    wrap_ExitProcess(exit_code: 0);
keyboard_layout = get_keyboard_layout(a1: 0);
if ( util_short_hash_lookup(langid: keyboard_layout) != 0 )
    wrap_ExitProcess(exit_code: 0);

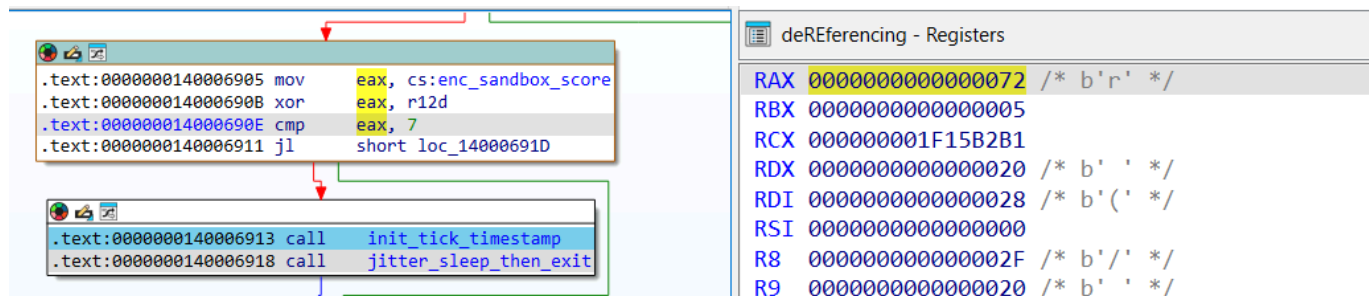
```

Exclusion checks based on language and keyboard layout

Sandbox scoring system

To determine whether the malware is running in a genuine environment, REVSTEALER integrates a custom sandbox scoring system using ten checks. Each check assigns a specific number of points based on sandbox characteristics such as low total physical RAM or CPU core counts.

If the total score is 7 or higher, the malware determines the machine is likely running in a sandbox environment and then self-terminates.



Sandbox threshold comparison

The following checks are used in this feature:

- Process Blocklist (Type 0)
- CPU Core Count (Type 1)
- RAM Threshold (Type 2)
- GPU / PCI Vendor Check (Type 3)
- Username / Computer Name Blocklist (Type 4)
- System Uptime (Type 5)
- Sleep / Timing Check (Type 6)
- Media Foundation Check (Type 7)
- CPUID Check (Type 8)
- Virtualization Check (Type 9)

Several of these checks hash values such as process names or usernames using a custom FNV-1a hash algorithm and compare the result against an embedded table. This method ensures no plaintext strings appear in the binary and prevents static analysis tools from seeing them.

Credential harvesting

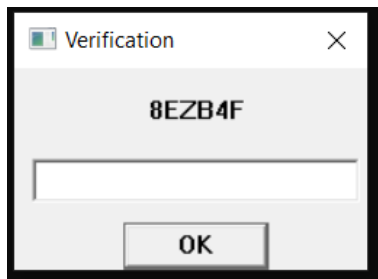
REVSTEALER targets browser credentials and cookies, cryptocurrency wallets, and browser extensions. This includes 225 Chromium extension identifiers spanning cryptocurrency wallets and password managers, alongside 51 standalone wallet applications. It also collects data from messaging clients, gaming platforms, VPNs, FTP clients, OBS Studio, Windows Sticky Notes, and 2FA applications, along with selected documents and configuration files from user directories and non-system drives.

REVSTEALER's Chromium support includes an implementation for bypassing App-Bound Encryption. To obtain the App-Bound key, it launches a browser process under debugger control, locates code associated with App-Bound decryption, and sets a nearby hardware breakpoint. When the breakpoint triggers, the malware reads the decrypted key directly from the browser's memory. The implementation was likely influenced by the public ElevationKatz project (under [ChromeKatz](#)).

Example of decrypted App-Bound key in memory

Payload watermarking

The developer places a 16-byte watermark at the end of raw payloads and validates it at runtime. Packed builds do not trigger the check because the watermark is no longer located at EOF. When a watermarked raw copy is launched, a verification window requires the user to enter a random six-character token in order to proceed with execution. This discourages distribution of unprotected payloads while also hindering sandbox analyses. Similar techniques have been observed in Lumma Stealer and AuraStealer.



Polygon dead drops as fallback

Each REVSTEALER build contains a primary C2 address and a Polygon smart-contract address. If the primary server is unavailable, REVSTEALER queries up to 5 public Polygon JSON-RPC endpoints, reads data from the contract, and decrypts a fallback C2 address using an embedded AES key.

Below are examples of extracted C2 hosts and Polygon dead drop configurations from three REVSTEALER samples with different themes.

Sample sha-256: 8b33e0f32c42a317e3d9cd67d5a6dc68e91a6cf9dd44742162858e7d83cf2073File Name: SteelSeriesGG.exeConfiguration(c2=b'polygon.iwmukj[.]xyz:443', polygon_contract_address=b'0x7e4126ADFE6679B3613F629CD49162Fb08fc53Bd', contract_function_selector=b'0xdbcfdad6', contract_return_aes_key='f07666ac22dafa1a361b8a8576baf69eadaf15431db9f3c34e93334d8d2f8d691')

Sample sha-256: bd97d5cab2d09b001d1b9e08890bce1a2b2cf8542a4b31c86b2def59328cafb2File Name: slack.exeConfiguration(c2=b'polygon.mnyhgxda[.]xyz:443', polygon_contract_address=b'0xEC6a6D31b36271eBD06450EA98c84eBa8a191d5', contract_function_selector=b'0xd21368b4', contract_return_aes_key='5c30efd5b9ff75023c3813705f5c47feccc496f439b24b3e572db4c3b08dffe3')

Sample sha-256: 1617552169df805405b1bb70f742d6eac5af342f31c6e39ff6b9613bd2392354File Name: qBittorrent.exeConfiguration(c2=b'static4.livelab[.]jone:443', polygon_contract_address=b'0x49cE5712164755ed212209bc71539bBc6fCFF541', contract_function_selector=b'0x071258c6', contract_return_aes_key='83f5b4ca629681abab2b9680472a304572a8c91b4a8ccc1189bf58c6186db09')

Looking up one of the contract addresses on OKLink, there is an initial transaction to create a Polygon contract (bytecode prefix 0x60806040), initialized with an encrypted fallback C2 endpoint.


```

    wstr_decrypt_ror5_sub(battlestate_settings_path, 0x28u, 0x936F); // \Battlestate Games\BsgLauncher\settings
    _InterlockedExchange(&dword_14004F0B4, 2);
}
}
if ( wstr_ncat(full_path_battlestate, 0x104, battlestate_settings_path) != 0 )
{
    hFile = wrap_CreateFileW_0(
        full_path_battlestate,
        GENERIC_READ,
        FILE_SHARE_READ|FILE_SHARE_WRITE|FILE_SHARE_DELETE,
        0,
        OPEN_EXISTING,
        0,
        0);
}

```

Polygon transaction creating the REVSTEALER dead drop contract holding an encrypted fallback C2 address

The following table lists the gaming targets and their paths:

Platform/Game	File path
Battle.net (Blizzard)	%LOCALAPPDATA%\Battle.net\Battle.net.config
Battlestate Games	%APPDATA%\Battlestate Games\BsgLauncher\settings
Electronic Arts	%LOCALAPPDATA%\Electronic Arts\EA Desktop*.ini
Roblox	%LOCALAPPDATA%\Roblox\LocalStorage\RobloxCookies.dat
Steam	%LOCALAPPDATA%\Steam\local.vdf, %LOCALAPPDATA%\Steam\loginusers.vdf
Minecraft	%USERPROFILE%\intentlauncher\launcherconfig %USERPROFILE%\lunarclient\settings\game\accounts.json %APPDATA%\minecraft\TlauncherProfiles.json %APPDATA%\feather\accounts.json %APPDATA%\minecraft\meteor-client\accounts.nbt %APPDATA%\minecraft\Impact\alts.json %APPDATA%\Badlion Client\accounts.json %APPDATA%\minecraft\launcher_accounts.json %APPDATA%\minecraft\launcher_profiles_microsoft_store.json

Four follow-on modules delivered by C2 tasking

REVSTEALER can receive additional executable content through C2 tasks. We recovered four related, self-contained payloads that extend its capabilities beyond credential theft.

Module	Capability
ProManager	Wallet-file and browser-extension theft, phishing overlays, password-aware input capture, and payload delivery
WinUpdate	Cryptocurrency-address replacement and mnemonic-shaped clipboard theft
SoftManager	Reverse SOCKS5 proxy and backconnect access over an encrypted WebSocket protocol
LockAppHost	XMRig deployment, competitor suspension, and persistence

Despite their distinct roles, all four REVSTEALER modules follow a common design pattern: obfuscated configuration, VMProtect-style protection, and Polygon smart contracts used as dead drops for replaceable settings, including C2 endpoints and XMRig's command line.

REVSTEALER technical white paper

Elastic Security Labs presents a comprehensive technical analysis of REVSTEALER, covering its full capabilities, network communications, anti-analysis, sandbox scoring system, EtherHiding techniques, and follow-on modules.

You can find the technical white paper [here](#).

String decryption script for REVSTEALER in IDA Pro

Using the [nightMARE](#) framework, our team is releasing an IDAPython script available for download [here](#). This tool can place decrypted strings in the IDA Pro decompiler to help analysts pinpoint the malware's functionality.

REVSTEALER and MITRE ATT&CK

Elastic uses the [MITRE ATT&CK](#) framework to document common tactics, techniques, and procedures that threats use against enterprise networks.

Tactics

Tactics represent the why of a technique or sub-technique. It is the adversary's tactical goal: the reason for performing an action.

- [Collection](#)
- [Command and Control](#)
- [Credential Access](#)
- [Discovery](#)
- [Execution](#)
- [Exfiltration](#)
- [Stealth](#)

Techniques

Techniques represent how an adversary achieves a tactical goal by performing an action.

- [Automated Exfiltration](#)
- [Clipboard Data](#)
- [Credentials from Password Stores: Credentials from Web Browsers](#)
- [File and Directory Discovery](#)
- [Indicator Removal: File Deletion](#)
- [Native API](#)
- [Screen Capture](#)
- [System Information Discovery](#)
- [User Execution: Malicious File](#)
- [Virtualization/Sandbox Evasion: Time Based Checks](#)
- [Web Service: Dead Drop Resolver](#)

Detecting and preventing REVSTEALER

Detection rules for REVSTEALER

[Network Activity to a Suspicious Top Level Domain](#)

Prevention rules for REVSTEALER

- [Shellcode Execution from Low Reputation Module](#)
- [VirtualAlloc API Call from an Unsigned DLL](#)
- [Network Library Load via LdrLoadDLL](#)
- [DNS Query to Suspicious Top Level Domain](#)

- [Potential Evasion with Hardware Breakpoints](#)
- [Remote Thread Context Manipulation](#)
- [Potential Browser Information Discovery](#)
- [Suspicious Remote Process Suspend Activity](#)
- [Browser Process Spawned from an Unusual Parent](#)

YARA rules for REVSTEALER

Elastic Security has created YARA rules to identify this activity.

[Windows.Trojan.RevStealer.yar](#)

Observables and indicators of compromise

This research discusses the following observables.

All observables are also available for [download](#).

Observable	Type	Name	Reference
4c897108e8e793d6904110928c996815c302d6975c9bc61162149e855a963d50	SHA-256	resightloader.exe	REVSTEALER Loader
adc4aa652965396b52e79435ca54987ae9eb21bf5e67de5e9461b09655165ee4	SHA-256		REVSTEALER
14b2ac356ed75d10ef40bbaaa48e7dd9fff7de9719c2a43ad123fe843dd4e4e2	SHA-256	SoftManager72fb40.exe	SoftManager
7c08cf409194056a8517865e5d3433d1499bb8262263b55b49b8b07d9d182fcb	SHA-256	WinUpdate60e3a3.exe	WinUpdate
13d7237d7289e67c2d806a65d52580b453ce4987acbe2c4c4d04833f55ebccfa	SHA-256	ProManagerServicedc894.exe	ProManager
c66d2b77b9e85c53391891212413ad9a99eb66f4b11c6a431e78884a5b2651e5	SHA-256	LockAppHost14a02b.exe	LockAppHost
monitor5.roast-core85[.]click	domain-name		REVSTEALER C2 server
config.hubdisplay[.]lol	domain-name		ProManager C2 domain
health.journal-metric[.]lol	domain-name		WinUpdate C2 domain
metric.gardenpark[.]click	domain-name		SoftManager C2 domain
0x98FF8e7cdC13AE46b83B7590B986F25f1560DF03	Wallet address		ProManager Polygon contract address

Observable	Type	Name	Reference
0x0cF1Ec8B9551103de729c3b02D77221Da9d81Acc	Wallet address		WinUpdate Polygon contract address
0x0E04c59f31E382D2B8A1637f4B9A5f04165EC48d	Wallet address		SoftManager Polygon contract address
0xC4eC9B7be1c2A0B39Eca678673DcB9164CA5df53	Wallet address		LockAppHost Polygon contract address

References

The following were referenced throughout the above research:

[Gen Threat Labs REVSTEALER coverage on X](#)

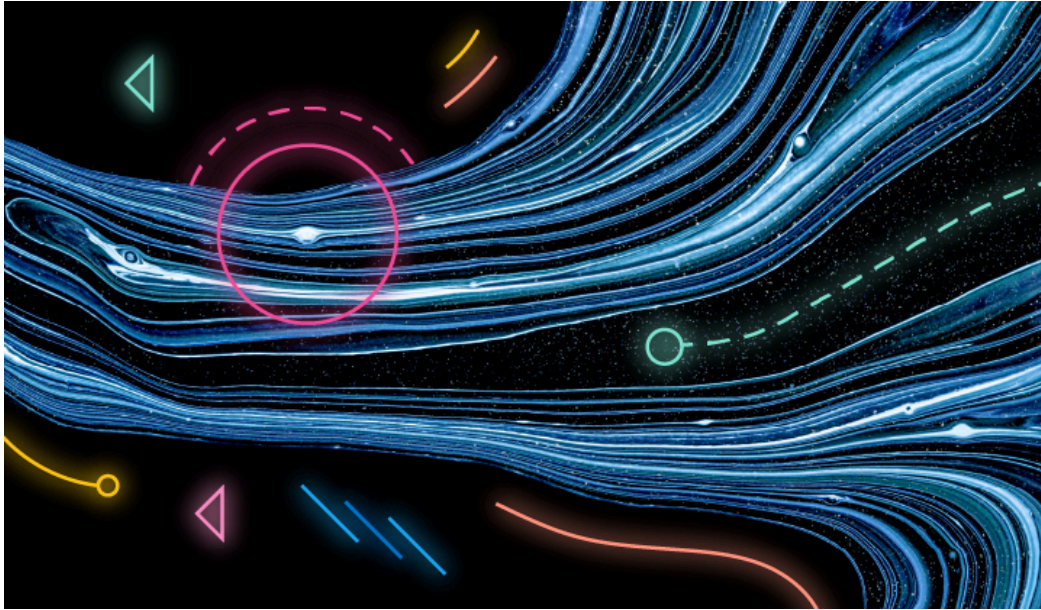
Related Content

[How to correlate Kubernetes audit logs with container runtime data](#)



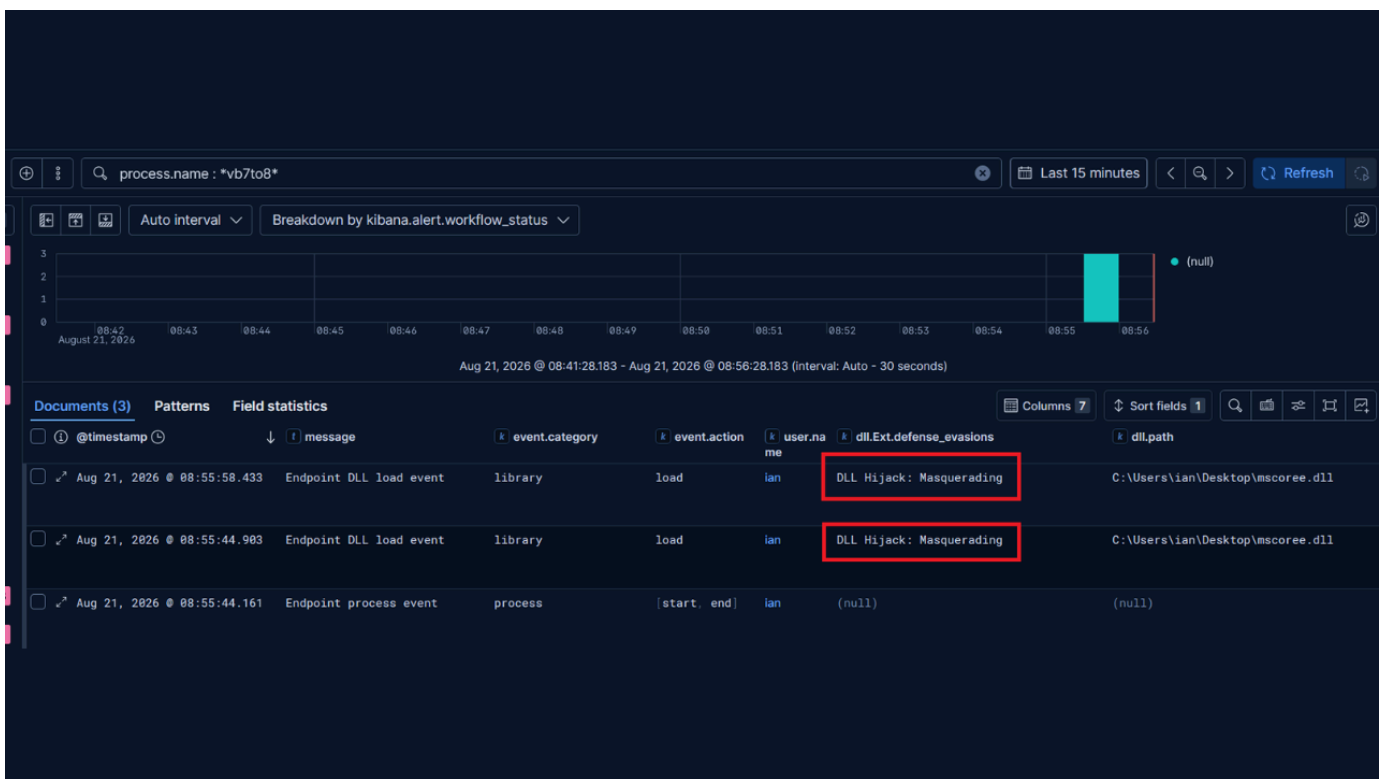
How to correlate Kubernetes audit logs with container runtime data

[Linux Detection Engineering - Fileless Execution](#)



Linux Detection Engineering - Fileless Execution

[From 88 lines to 1: Detecting DLL hijacking with Elastic Defend](#)



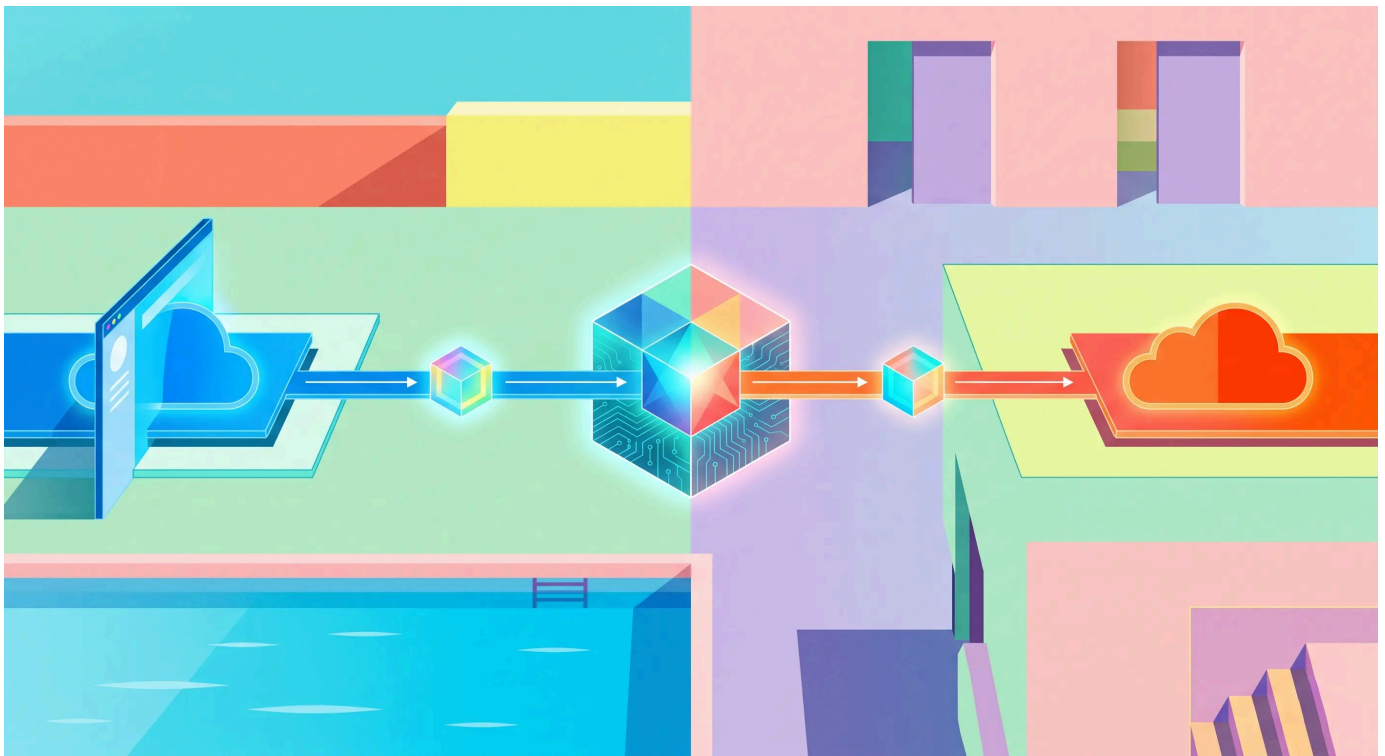
From 88 lines to 1: Detecting DLL hijacking with Elastic Defend

[Exploring the Hugging Face Breach: mapping AI agent tactics to Elastic Defend](#)



Exploring the Hugging Face Breach: mapping AI agent tactics to Elastic Defend

[Detecting Tycoon 2FA AiTM attacks across Entra ID and Google Workspace](#)



Detecting Tycoon 2FA AiTM attacks across Entra ID and Google Workspace