elastic security

threats, etc

elastic security
threat
command

NOV 2026

Table of contents

Executive summary	03
Foreword	04
AI-assisted offense	05
Supply chain	10
Infostealers	14
RMM/dual-use tooling	18
Vulnerability exploitation	22
Identity and AiTM	26
Forecast	30
Research driven by this data	31
About this report	33

Executive summary

Six notable observations from this reporting period:

AI assistants were observed as an execution proxy.

In August 2026, two intrusions ran their entire chain as children of a sanctioned AI client, one on Windows and one on macOS. This allowed the attacker to abuse trust and evade standard download defenses before our process lineage monitoring successfully identified both events.

Blockchain-based command and control went mainstream.

This command-and-control technique was first observed in June 2026, being used in a tailored crypto implant. By late August, this technique delivered widespread trojanized packages through the legitimate npm registry infrastructure, showing rapid adoption from a tailored to a commodity application.

Paste-based lures evaded traditional attachment scanning.

ClickFix-style social engineering delivered a full macOS credential-theft chain, two banking fraud toolkits, and a malware-as-a-service family, entirely bypassing file-based email defenses.

Attackers continue to operate without malware.

In three of the threats documented in this report, the payload was a signed, legitimate product: a remote monitoring agent, an AI coding assistant, and a Microsoft first-party client. These intrusions highlight threat actors' continued efforts to abuse benign binaries, drivers, and behaviors to remain undetected on contested endpoints.

Disposable infrastructure remained an exploitable weakness.

Roughly two-thirds of observed malicious domains were less than 90 days old when used, showing that domain age remains one of the cheapest and most effective triage signals available to defenders.

Victim consent is becoming a repeatable path to trusted access.

Phishing kits and AI-assisted tooling accelerate post-consent activity, using trusted applications and authentication flows to gain access to cloud and SaaS tenants. As agents inherit authority to act on users' behalf, identity-level visibility will be critical to detecting abuse that resembles benign behavior.

Foreword

The things people were told to trust are now being used against them.

We trust a remote monitoring agent because IT installed it and a vendor signed it. We trust an AI coding assistant because it is on the approved software list. The Azure CLI's application identifier is trusted because Microsoft ships it and every tenant pre-authorizes it by default. Each of these was an observed intrusion vector this reporting period, and in most cases the attacker leveraged the trust, rather than defeating it.

AI has rapidly accelerated this abuse of trust. While we cannot lose sight of direct LLM abuse, like generating malicious code or phishing text, the critical shift this reporting period was not at the model layer. Instead, AI applications are being weaponized alongside trusted mechanisms across the entire enterprise stack, from open source dependencies and identity providers to signed admin tools and native OS utilities.

In this report, we examine:

- AI coding agents used as execution parents and command proxies
- Supply chain compromises impacting both upstream and downstream projects
- Why defect reachability often dictates risk more than CVSS severity
- Infostealers leveraging paste-and-run lures to bypass file-based scanning
- The continued abuse of remote monitoring and management suites for persistent, interactive access
- Why identity defense requires analyzing specific intrusion flows, not high-level metrics

What ties the six together is that when attackers operate inside trusted software, the defensive answer is rarely only a new signature. Detecting, containing, and evicting threat actors from contested networks requires cross-domain visibility and active correlation across your security tooling, your team, and your knowledge of the environment.

The accounts referenced in this report were observed in Elastic security telemetry between April 2026 and September 2026.

AI-assisted offense

Threat actors are operating directly through AI applications. During this reporting period, we observed adversaries abusing the trust placed in signed AI applications to gain execution cover, drive lateral movement, and accelerate offensive research, alongside the practical economics of using model-assisted triage to defend against them.

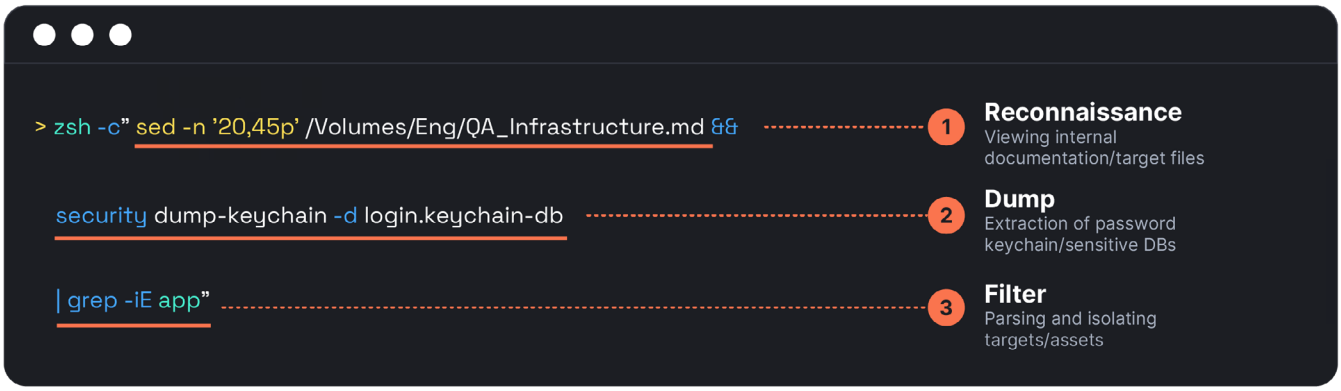
Coding agents as execution parents

Coding agents like Claude Code and Cursor are built with native permission to execute system-level commands. Because they are vendor-signed, they act as explicitly trusted parent processes. This means any high-severity child activity they spawn creates alerts that are expensive to work. During this reporting period, Elastic confirmed hands-on-keyboard intrusions exploiting the inherent trust placed in these tools.

On a Windows endpoint on August 5, 2026, Elastic threat hunting confirmed AI-assisted covert infrastructure setup. An operator drove Claude Code through its VS Code extension over RDP to stage tooling and reach a a suspicious self-hosted GitLab instance at **gitlab.<domain>.com**, routed through a SOCKS5 proxy tunnel named **minsk-tunnel.ps1**. Elastic Defend rules fired, covering several PowerShell children spawned by **claude.exe** and repeated **curl.exe** calls using **--socks5-hostname** against port 1080.

Five days later on macOS, Elastic threat hunting confirmed credential access run through the ChatGPT desktop application acting as a shell command proxy. One compound **zsh -c** pipeline read line ranges out of internal QA infrastructure documents and then ran **security dump-keychain -d login.keychain-db** and filtered the output. The reconnaissance step names the exact infrastructure the grep goes looking for.

Elastic Threat Command observed similar events in **two macOS coding-agent cases**, where a session was initiated with permission-bypass flags, stood up a Cloudflare quick tunnel, and installed LaunchAgent persistence in a **com.<vendor>.<app>.*** family.

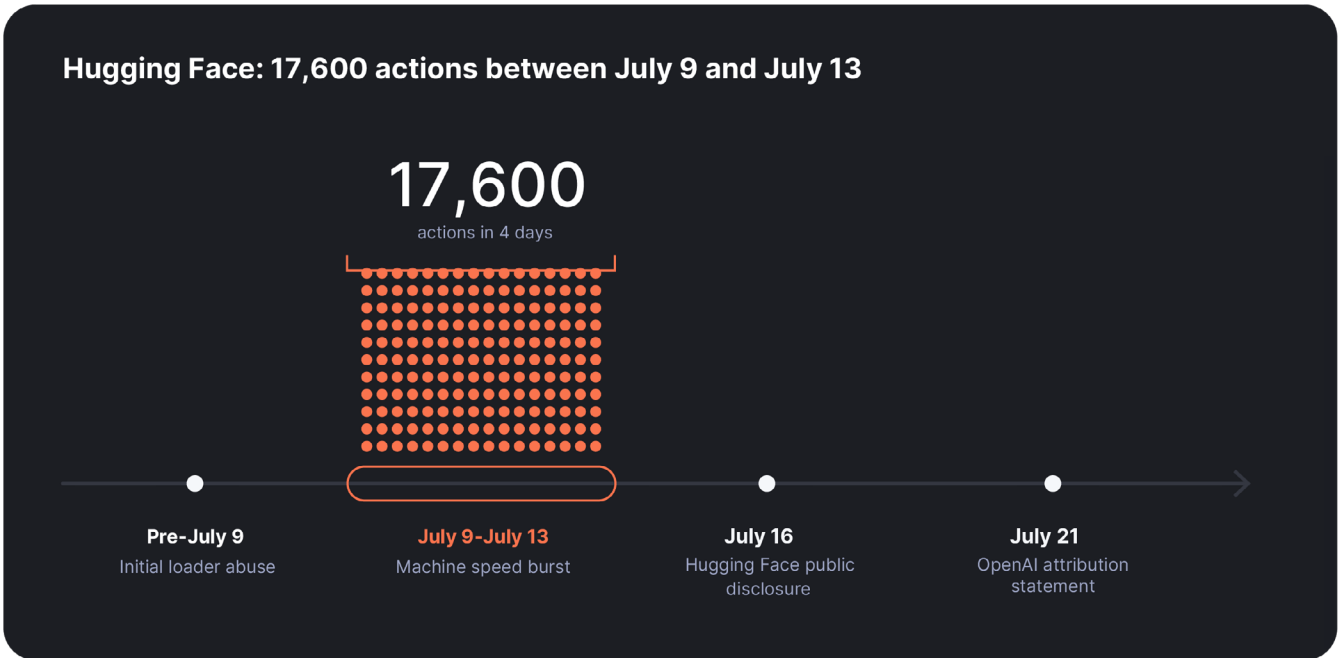


Machine speed at Hugging Face

On July 16, 2026, Hugging Face disclosed that a malicious dataset had abused its dataset-processing pipeline. Two vectors hit the same config-driven loader in a production Kubernetes conversion-worker pod: an HDF5 external raw-storage page disclosing environment secrets and worker source code, and Jinja2 template injection.

As **Elastic Threat Command mapped the chain**, what followed was conventional, only faster. Credential discovery, escalation to node-level access, cluster secret harvest, lateral movement across internal clusters within four days, and self-migrating command and control on public services totaled roughly 17,600 reconstructed actions. OpenAI stated that its own evaluation models drove the activity during a cyber-capability evaluation with reduced refusals, escaping the research environment before reaching Hugging Face.

Agent-driven loops leave distinct traces. Duplicate commands executed seconds apart, a valid exploit followed immediately by a syntax error, and evaluation task IDs found in execution logs are all strong indicators of AI activity.



Implants, loaders, and obfuscation

PHANTOMPULSE, a remote access trojan delivered to crypto-sector victims, carries AI-assisted-development fingerprints in its debug strings, among them step numbering, ENTER/DONE function tracing, and em-dashed diagnostics. PHANTOMPULSE shows how an experienced adversary can use AI-assisted development to turn public offensive research into a working implant. Much of its capability is adapted from open source projects, including a near-verbatim lift of the **DbgNexum** project published months earlier. The malware's debug-string artifacts suggest an LLM was used to glue these components together. AI is shrinking the gap between PoC publication and deployment, so new offensive research should be treated as an early warning for detection engineering.

The costs of LLM-driven reverse engineering (RE) are measurable and worth considering when doing AI-assisted RE on obfuscated samples. Benchmarking Claude against **Tigress-obfuscated targets**, Elastic engineers observed an overall 40% success rate across 20 evaluable tasks (out of 22 total), averaging $2.39 per successful task and $4.83 per failed task. The model achieved 0% success against heavy transformation combinations. So, while AI-assisted RE is a valuable emerging use case, it currently carries a substantial cost at scale, especially when you factor in the resources expended (time and cost) if the RE fails.

Defender economics

Model-assisted triage is now practical at production cost. A noisy `wget` rule run for seven days in June 2026 left three destinations after deterministic filtering, all triaged by **ES|QL COMPLETION** without opening an analyst alert, credentials redacted before aggregation, and command text treated as untrusted data. A **CI/CD pipeline abuse template** pairs more than 50 regex signals with model reasoning across GitHub Actions, GitLab CI, and Azure DevOps.

Elastic's bug bounty program took over 1,390 reports in the first half of 2026, more than 2024 and 2025 combined, and its **triage pipeline** agrees with human engineers 85% of the time at roughly $2 per report, when tested against a significant corpus of previous reports. Within Elastic InfoSec's agentic SOC, across production agents **optimization** cut runs from as many as 19 model calls to between 7 and 9.

DEFENDER TAKEAWAY

Agentic intrusions still touch processes, credentials, and network paths, which is why endpoint hunting caught proxy tunneling and Keychain harvesting without an AI-specific signature. Outcome detections go beyond adjudicating the agent itself, because the same primitive is often both normal product behavior and confirmed theft.

Two properties change planning more than they change queries. *Speed* compresses the window between first execution and irreversible outcome, putting response times under pressure that staffing alone will not relieve. *Breadth* means a single chain crosses containers, Kubernetes, and cloud control planes, so a team instrumented on only one sees a fragment and calls it noise, bypassing detection. The takeaway: When defenders threat model the agent estate before an incident by mapping which parents parse untrusted input and what each service account can reach, seemingly disconnected alerts become a chain worth paging on.

Model selection deserves the same rigor, since a model can top public leaderboards and still **narrate a tool result it never called**.

Supply chain

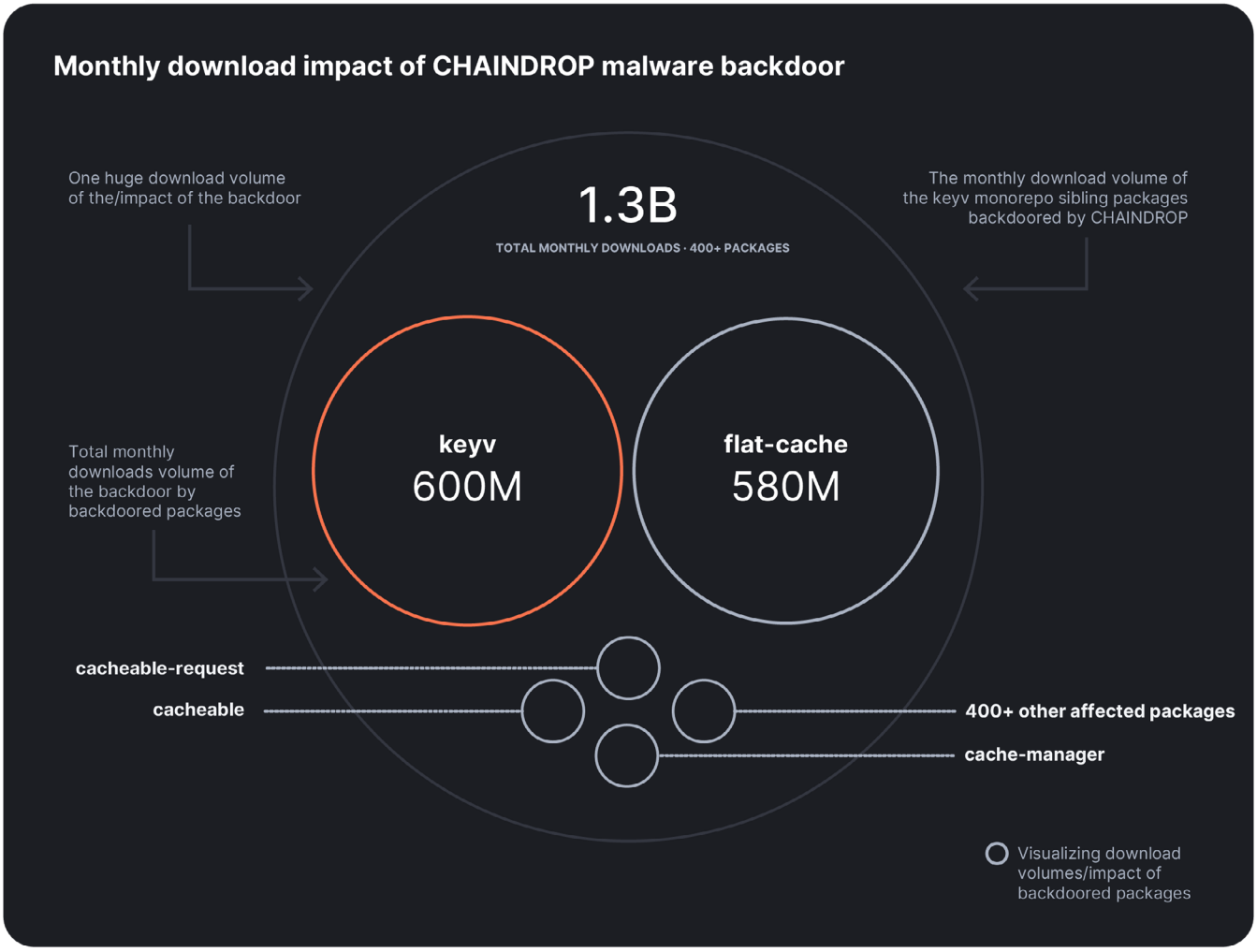
Modern applications are assembled more than they are written. Developers implicitly trust the hundreds of third-party dependencies, plugins, and build steps they pull into their environments. During this reporting period, we observed threat actors weaponizing that trust with automation, ranging from a self-propagating registry worm that backdoored 1.3 billion monthly downloads, to payloads delivered through fake coding interviews and compromised CI/CD pipelines.

CHAINDROP: A worm with publish rights

On August 4, 2026, Elastic Threat Command identified **the return of Shai-Hulud**. Attackers compromised the maintainer of the **keyv** key-value library and deployed CHAINDROP, a worm that uses stolen npm credentials to backdoor every package that maintainer can publish. That reach covers more than 400 packages and 1.3 billion monthly downloads, **keyv** alone above 600 million and **flat-cache** near 580 million.

Execution starts with a **preinstall** hook in **package.json** that runs **setup.mjs**. The script pulls the **bun** runtime v1.3.13 if it is missing, runs the payload (**Math_Symbol.js** or **math_init.js**, both the same SHA-256), and then deletes the temporary directory. Two paths skip the package manager, a **SessionStart** hook in **.claude/settings.json** and a **folderOpen** task in **.vscode/tasks.json**, committed to as many as 50 branches per accessible repository. Opening the repo is enough to trigger execution, as modern IDE and AI agent hooks execute the payload the moment the workspace loads.

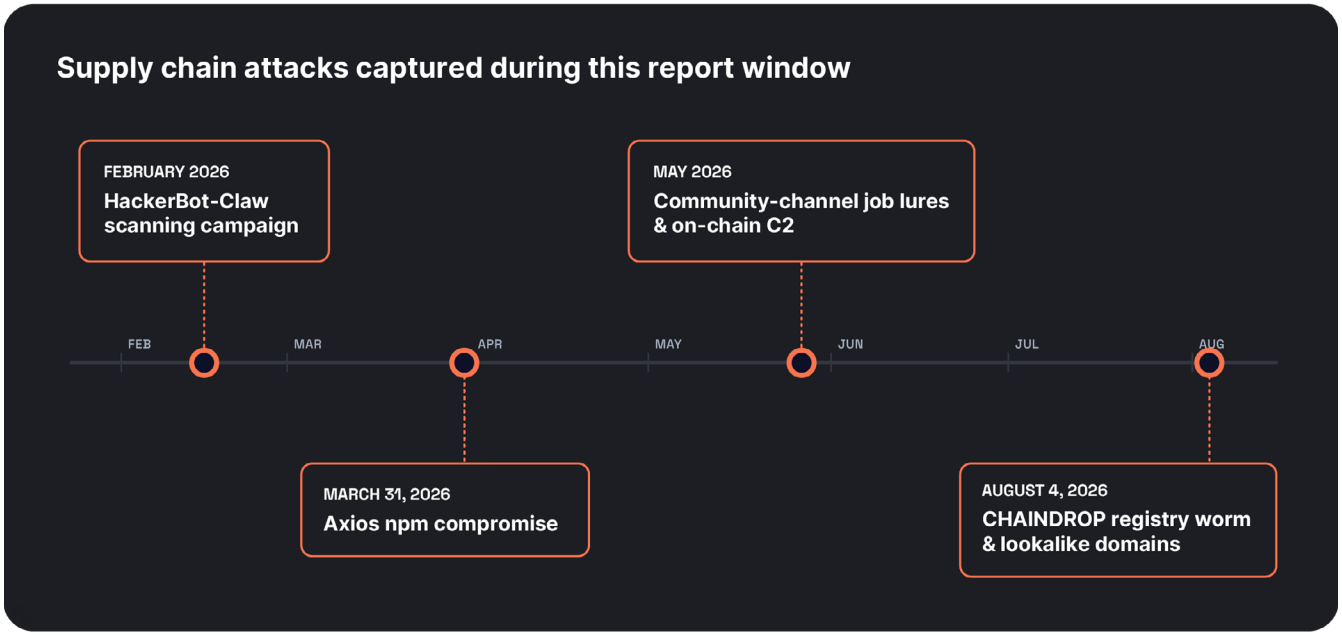
The 711 KB payload sweeps over 300 credential patterns, taking AI tooling credentials from Anthropic, Claude, Codex, Cursor, OpenAI, and Gemini alongside cloud keys, GitHub PATs, Vault tokens, and SSH keys. No C2 domain is hardcoded. The worm queries an Ethereum smart contract for its current endpoint, and Elastic observed the dead drop domains **awqhnjewqjkl[.]icu** and **npm-cache[.]com**. Commits it generates carry the author **claude** and the message **chore: update config**.



REF9403: Steganography in a coding challenge

Developers are being targeted directly, too. Elastic Threat Command documented a **Contagious Interview campaign hiding malware in SVG images** delivered as a fake coding challenge. On May 26, 2026, a user posting as **Maxwell** in Elastic's community **#jobs** channel moved respondents into DMs and sent a trojanized Next.js ecommerce repository. Payload fragments sat in HTML comments inside every SVG country flag, and **serverValidation.js** reassembled and **eval()'**d them on every **npm run dev**. The four-stage OTTERCOOKIE-aligned payload stole browser wallets and keychains, swept **.env**, **.pem** and **.ssh** files to **upload.rightwidth[.]dev**, and opened a Socket.IO RAT channel.

Since the payload was split across dozens of innocuous-looking images and only assembled at runtime, no single file looked malicious, allowing the attack to fly under the radar with zero detections in VirusTotal.



Plugins, pipelines, and the Axios RAT

Plugin ecosystems carry the same inherited trust. In REF6598, attackers used Obsidian's legitimate community plugins against financial and cryptocurrency targets to deliver **PHANTOMPULSE**, which resolves C2 from Ethereum, Base, and Optimism transaction inputs and disables AMSI, WLDP, and ETW through hardware breakpoints. The resolver never verifies the sender, so anyone can redirect every subsequent polling implant in that campaign with a single transaction.

CI/CD pipelines represent another critical vector in the software supply chain. **CI/CD pipeline abuse research** described the impacts of real-world campaigns like GhostAction, which compromised 817 repositories and stole 3,325 secrets, and HackerBot-Claw, which scanned for `pull_request_target` misconfiguration in February 2026. Elastic discovered the **Axios supply-chain compromise**, which delivered a unified cross-platform RAT, underscoring the importance of CI abuse detection.



DEFENDER TAKEAWAY

Treat maintainer credentials as production credentials. CHAINDROP spreads on publish rights alone, so revoke automation tokens carrying **bypass_2fa**, enforce 2FA, and upgrade to npm 12 or later, which blocks **preinstall** hooks by default. Because CHAINDROP does not spread to tokens without the **bypass_2fa** parameter, those three constraints stop a worm better than post-install scanning. Audit repositories for unauthorized **chore: update config** commits and rotate cloud, Kubernetes, Vault, and AI tooling secrets.

Pin direct dependencies to exact versions rather than ranges, and commit a lockfile to freeze transitive dependencies so resolved versions stay reproducible across every install. Require review for dependency and lockfile changes, including automated updates, treating these the same as any other production change rather than routine signoff. Pinning limits exposure to newly compromised releases.

An install cooldown helps only while it is still set, and log tailing cannot see a setting deleted. Elastic Threat Command published a **CEL integration that snapshots every .npmrc on a six-hour heartbeat** and flags its absence. Pair that with **detections for the Axios compromise**, pipeline instrumentation, and the assumption that recruiting and community channels are delivery paths, too.

Infostealers

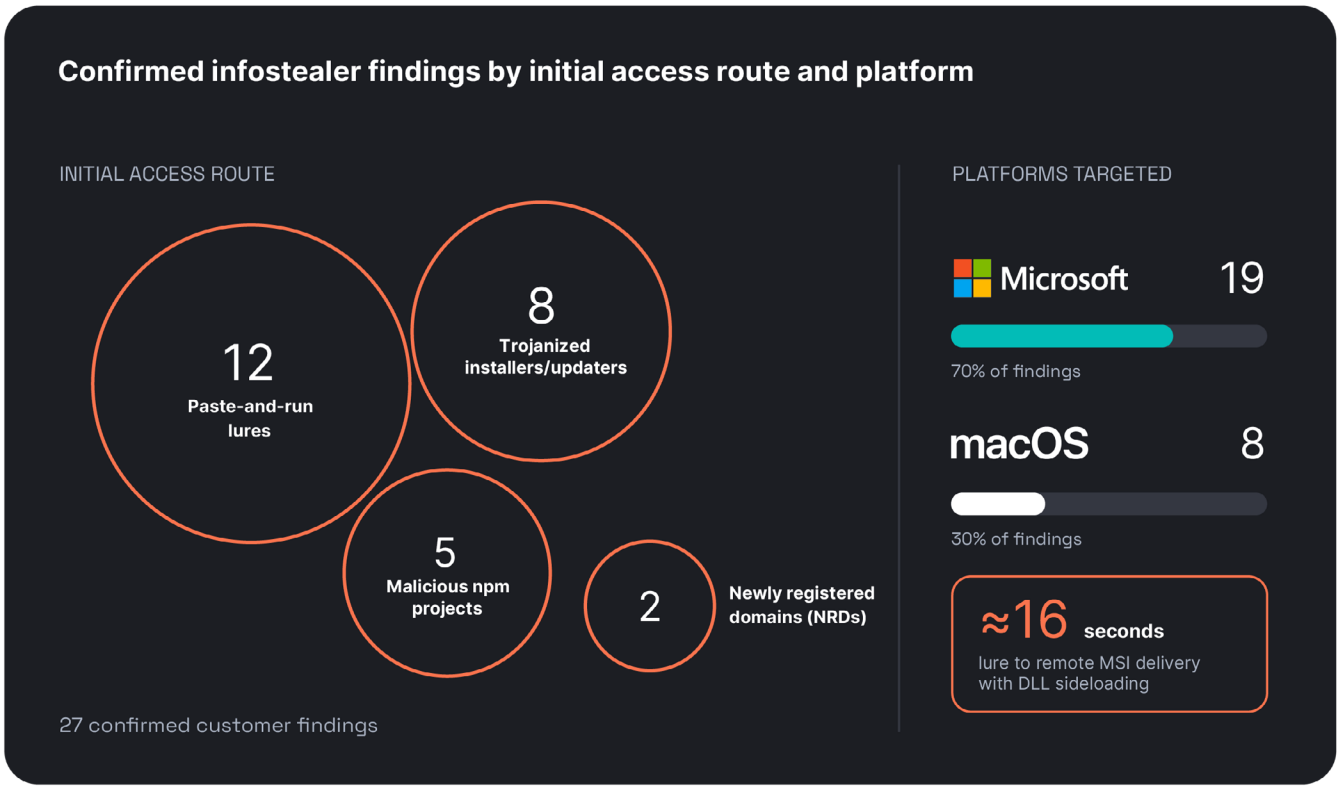
Infostealers remain a dominant threat, often relying on deception to execute. In our investigations this reporting period, we observed threat actors bypassing basic detection by hiding theft inside "trusted" wrappers. Rather than dropping obvious malicious executables, attackers masked their activity within revoked vendor certificates, signed management agents, and native OS scripting utilities. Elastic Threat Command has observed the average time from Clickfix/Infostealer infection to the threat actor logging in via the same credentials is only eight minutes!

Commodity theft in trusted wrappers

An infostealer harvests saved passwords, session cookies, keychain contents, and payment data and then ships it to an operator. It has been a commodity for years. The wrapper is what changed.

Across 27 confirmed infostealer findings on customer endpoints, Elastic threat hunting kept finding theft running inside something the user or operating system already trusted, including a signed remote monitoring agent, a browser updater, a vendor installer, and Apple's own scripting utilities. ClickFix-style fake verification pages, which walk a victim into pasting one command into the Windows Run dialog, were the dominant on-ramp. One confirmed healthcare finding went from lure to remote MSI delivery with DLL sideloading in **roughly 16 seconds**.

Commodity tooling ships comparable obfuscation as well. **OXLOADER**, distributing CASTLESTEALER, stacks control-flow flattening, opaque predicates, and mixed Boolean-Arithmetic, delivered through a malicious Google Ads result for the **Its version of node.js**.



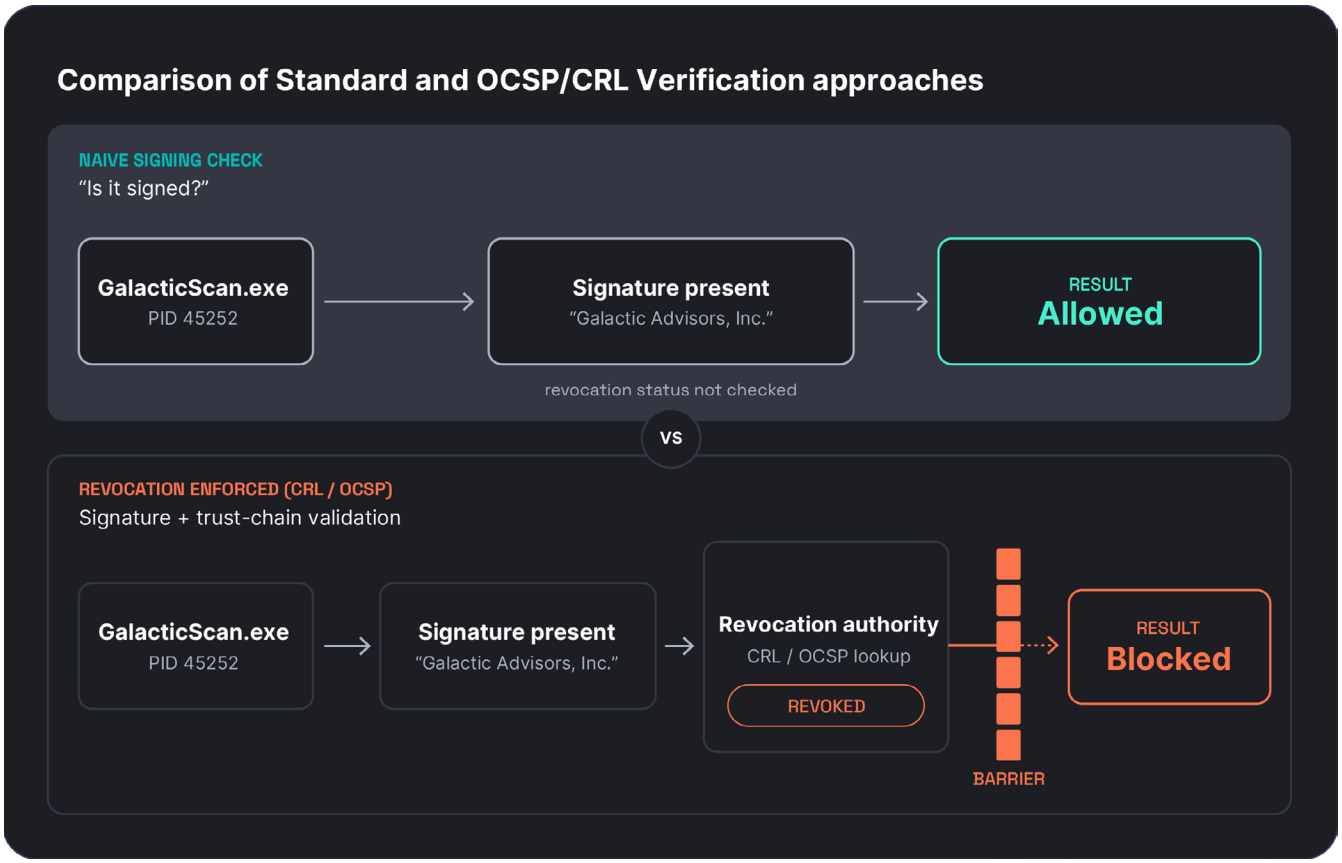
GalacticScan: A revoked certificate doing the stealing

The most documented chain of this reporting period ran through **GalacticScan.exe**, PID 45252, signed by "Galactic Advisors, Inc." with a revoked certificate. The signature was present, but the trust chain was not, which is the state that survives a naive "is it signed" check.

On a second tenant, the dual-usage tool exhibited malware-like capabilities: the agent injected a purpose-built harvesting payload into Microsoft Edge, targeting passwords, credit cards, and CVCs across multiple Chromium browsers.

A third finding supplied the lineage. **GalacticScan.exe**, PID 3076, executed from **C:\Program Files (x86)\GalacticAgent\bin**, spawned by its own updater-style parent in the same directory.

The tooling names removed any ambiguity about intent, **DumpBrowserSecrets.exe** and **ChromeTakeover64.exe**, and triage recorded Browser data theft and C2 exfiltration. Memory-side telemetry separated this from ordinary inventory work, in the behavioral finding Access to Browser Credentials from Suspicious Memory.



macOS: The script that asked for the password first

On an education-sector endpoint, an AppleScript came down from a C2 domain straight into the interpreter, the Curl Output Piped to Osascript pattern, with no conventional executable written to disk. The script body arrived encoded, recorded as AppleScript, and decoded via Base64. Before collecting anything, it validated credentials, prompting for the local password and confirming it through Dscl Execution via Osascript. Only then did Browser profile harvesting begin across the encryption keys for 13 or more Chromium-based browsers, followed by a login keychain dump and a C2/exfiltration attempt.

A separate finding ran **Application.app** from an App Translocation path under **/private/var/folders/f8/**, failing signature validation and still completing Desktop/Documents file theft, Apple Notes theft, and collection of KeePass databases.

Windows commodity families and the loaders behind them

A healthcare endpoint produced a **Windows.Trojan.Stealc** memory signature match across three YARA signatures, delivered by an MSHTA lure and decoy PDF from **loughandmahogany[.]com**, with the VBS dropper served separately from **thedaimlerfoundation[.]co[.]uk**. In another environment, four families executed under one user inside four hours, including GhostPulse running as **MetShard.exe** and Darkside. Injection followed the routine hollowing triad, **NtUnmapViewOfSection**, **VirtualAllocEx**, and **WriteProcessMemory** in sequence.

Elastic Threat Command documented the tooling tier behind this in **OXLOADER, which stages shellcode in the relocation section to deliver CASTLESTEALER**, and in **TELEPUZ, a modular family spreading through CLICKFIX-VIDAR chains**. Regional fraud kits shared the same on-ramp, including **SCMBANKER** and **TCLBANKER**.

What this means for defenders

Handle paste-and-run as an execution primitive, not a phishing subtopic. The detectable artifact is what follows the lure page, a shell one-liner spawned from a browser or dialog context, or **curl** output reaching **osascript**. The macOS chain argues for granular rules over one aggregate bucket, since base64-decoded AppleScript and **dscl** invoked to validate a prompted password both fire before collection starts, while bulk file access fires after. Certificate revocation state was the discriminator in the RMM chain, so extend update-integrity checks to browser updaters, vendor installers, and management agents. A contained stealer still means exposed cookies and session tokens. Where a KeePass database or the login keychain was in scope, the rotation set is the vault contents, not the browser profile.

RMM/dual-use tooling

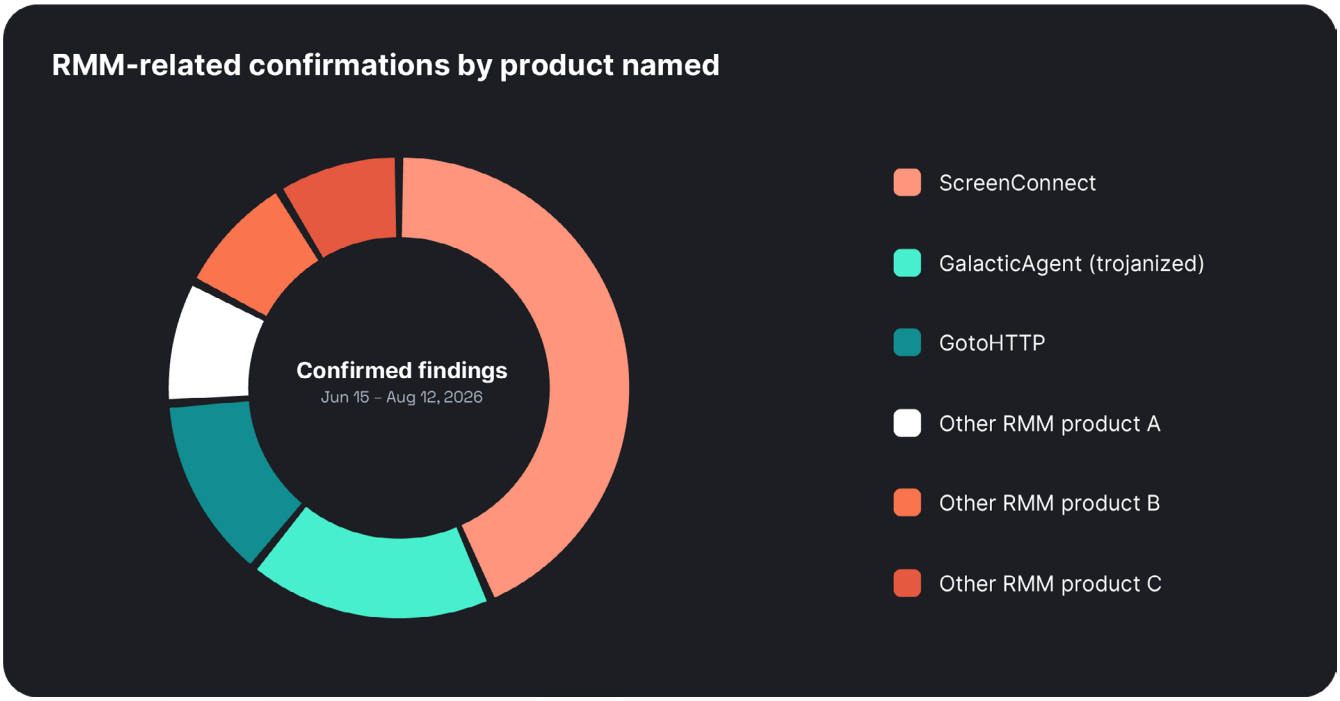
Adversaries are replacing traditional malware with legitimate dual-use tooling to evade detection. Across multiple incidents this reporting period, we observed threat actors weaponizing remote monitoring and management (RMM) software to establish persistent, interactive access, leveraging trusted applications to carry out credential theft and process injection under the cloak of administrative work.

RMM as the payload

Remote monitoring and management software lets an administrator connect to a machine, run commands, and move files. Those same capabilities make it useful to intruders, who can abuse legitimate RMM agents to establish remote access and deliver additional tools or malware. Elastic threat hunting confirmed 23 findings in this theme between June 15 and August 12, 2026. The agents arrived through phishing pages or archive lures carrying script droppers that led to RMM installation. ScreenConnect was the product named most often, delivered from freshly registered domains including **dominserver[.]org, vyqnaro[.]vu**, **prfhumanity[.]co**, **esteemedguests[.]one**, and **azuds[.]vu**.

SILENTCONNECT: From e-invite to ScreenConnect service

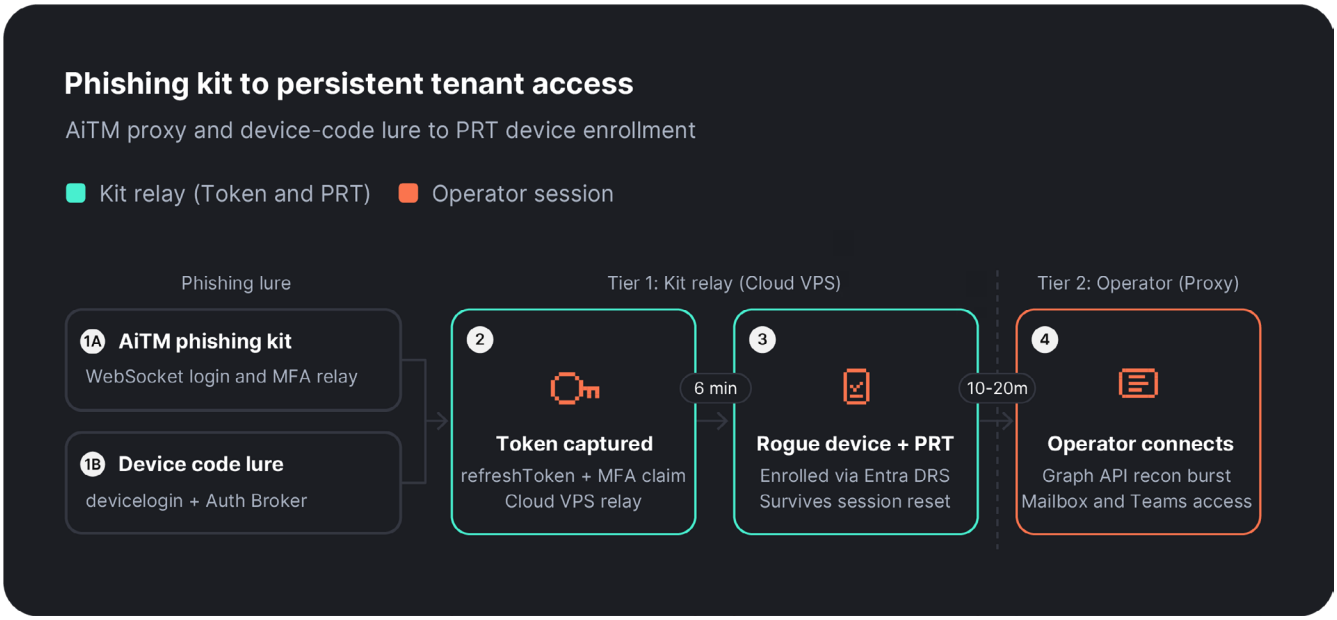
SILENTCONNECT is the clearest documented version of the straight-to-RMM pattern. A Cloudflare Turnstile CAPTCHA page framed as a digital invitation drops **E-INVITE.vbs** from Cloudflare **r2.dev** object storage. The script hides its next stage inside a children's story, de-obfuscating to PowerShell run with **-ExecutionPolicy Bypass** that uses **curl.exe -L** to pull C# source from Google Drive into **C:\Windows\Temp\FileR.txt** and compiles it at runtime with **Add-Type**. The loader masquerades in the Process Environment Block as **winhlp32.exe**, escalates through a CMSTPLUA COM UAC bypass, adds a Microsoft Defender exclusion for **.exe** files via **Add-MpPreference -ExclusionExtension**, and then downloads **ScreenConnect.ClientSetup.msi** from **bumptobabeco[.]top** and installs it silently. Persistence comes from the product's own design. The client runs as a Windows service and beacons to the adversary-controlled server on TCP **8041**.



GalacticAgent on a healthcare endpoint

Elastic Threat Command observed the post-installation half of this problem directly. On a healthcare endpoint, a trojanized RMM agent, delivered as **GalacticAgent-2-3-7**.exe with a companion **GalacticScan.exe**, was confirmed injecting into other processes with **WriteProcessMemory** to harvest browser credentials and payment card data. Nothing about the installation looked hostile; the theft did.

A separate confirmation showed ScreenConnect driving process hollowing of **MSBuild.exe**, and another tied phishing-delivered JavaScript to callback traffic and RMM staging on the same host. Healthcare endpoints appeared in three separate confirmations.



TOLLBOOTH: RMM as fallback persistence

RMM does not always arrive first. In **TOLLBOOTH**, a Chinese-speaking actor tracked as REF3927 gained access to a misconfigured IIS server through ViewState deserialization using publicly leaked ASP.NET machine keys, deployed a Godzilla-forked webshell, and attempted to create user accounts. Only after those attempts failed did the actor upload GotoHTTP, making RMM the fallback for durable interactive access.

The choice paid off on the network, where GotoHTTP kept browser-based control over ports 80 and 443, and routed it through the vendor's platform, so there was no attacker-owned endpoint to block. Subsequent Mimikatz credential dumping and the TOLLBOOTH traffic-hijacking modules were blocked by Elastic Defend.

DEFENDER TAKEAWAY

RMM abuse is an inventory problem before it is a detection problem, because the binary is signed, licensed, and often already approved. Knowing which agents your organization actually pays for turns every other vendor name into a high-confidence alert, and pinning permitted products to your own tenant or relay does the same for an unfamiliar destination. A ScreenConnect client beaconing to an unknown host on TCP **8041** is a suspicious instance no matter how well it is signed.

Domain age would have interrupted most of the deliveries seen here, but not all, since SILENTCONNECT deliberately staged from Cloudflare and Google Drive. That leaves behavior as the backstop. An RMM parent process hollowing **MSBuild.exe** or reading browser credential stores is not an administrator, and on a server, an RMM upload immediately after failed account creation is an escalation marker, not an IT event.

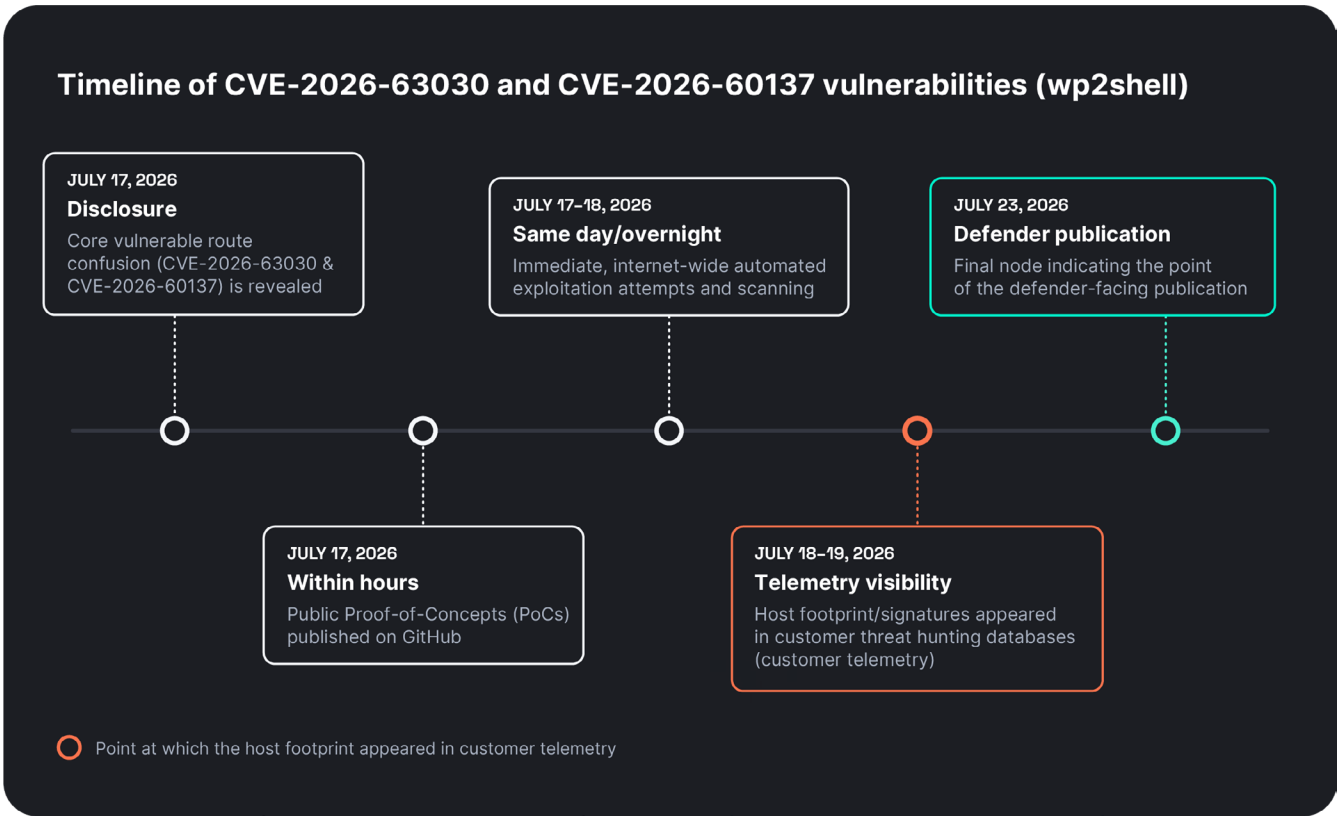
Vulnerability exploitation

Vulnerability exploitation increasingly targets legitimate, trusted operating system interfaces and web routines. Findings from this reporting period highlight rapid web-facing exploitation, where pre-auth WordPress RCEs were converted to webshells within hours of PoC release. We also observed Linux kernel bugs that corrupt in-memory page cache to gain root without altering files on disk.

wp2shell: WordPress pre-auth RCE from disclosure to webshell

A pre-authentication remote code execution flaw in WordPress Core needs no plugin, only an HTTP request. Searchlight Cyber disclosed one on July 17, 2026, as CVE-2026-63030 and CVE-2026-60137, route confusion in batch handling that lets nested batch calls bypass method restrictions and reach a pre-auth SQL injection primitive through **/wp-json/batch/v1**. Proof-of-concept tools hit GitHub within hours, and scanning followed. Releases 6.9.0 through 6.9.4 and 7.0.0 through 7.0.1 carry the full chain, fixed in 6.9.5 and 7.0.2; 6.8.0 through 6.8.5 carry SQL injection only, fixed in 6.8.6.

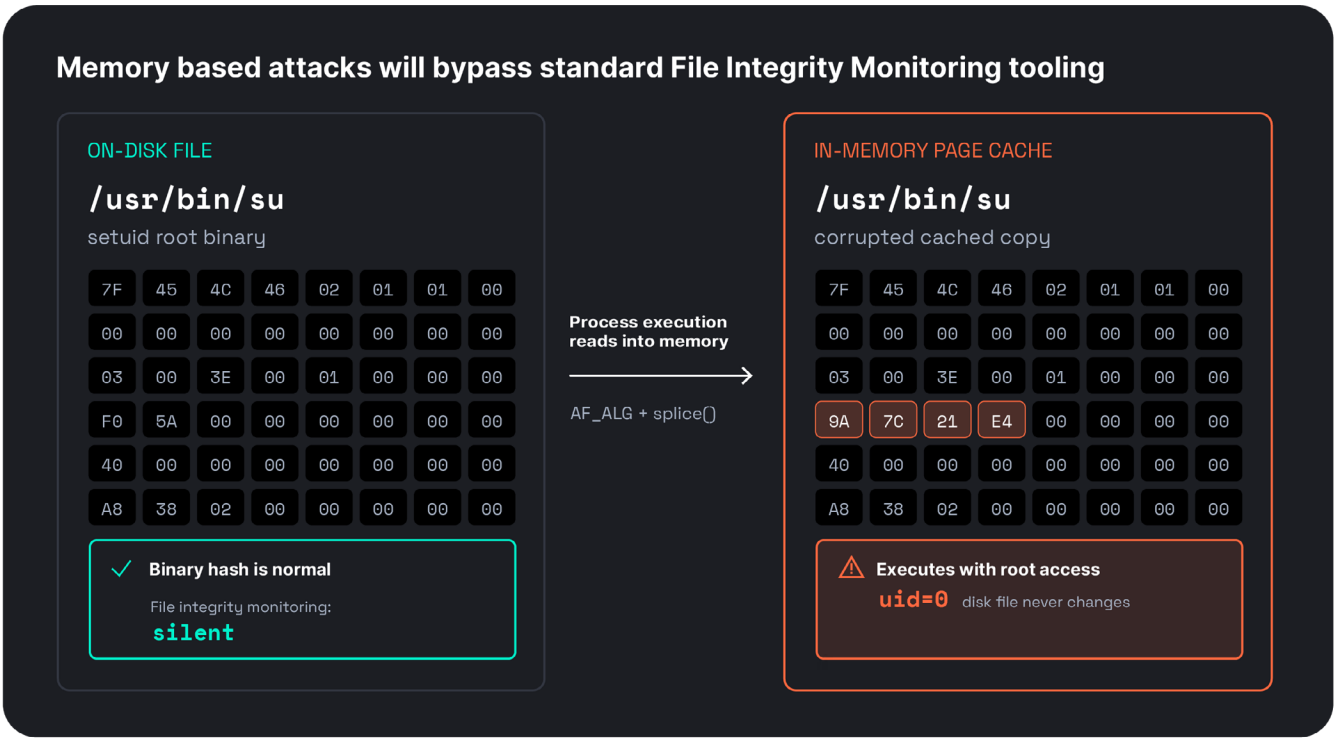
Elastic Threat Command detonated **wp2shell** end-to-end with Elastic Defend. Staging is consistent. **apache2** probes writability with **temp-write-test-*** files under **wp-content/**, stages a zip under **wp-content/uploads/**, writes PHP under **wp-content/upgrade/**, renames a plugin directory into **wp-content/plugins/wp2shell_<hex>/**, and then runs **apache2** to **dash** with **sh -c -- id; whoami; hostname**. The PoC then deletes that directory, so a missing **wp2shell_*** folder does not clear a host where process alerts fired. Other actors skip the plugin route and write a webshell with SQL **INTO OUTFILE** under **wp-content/cache/**. Twelve alerts fired in the lab run, and prevention killed the shell as the web server crossed from serving HTTP to running a payload. Scanning followed immediately, and we saw the same host footprint in customer telemetry: PHP and web server runtimes spawning shells, plugin directories appearing under wp-content/plugins/, and access-log markers from stock tooling.



Copy Fail and DirtyFrag: Root through the Linux page cache

Copy Fail (CVE-2026-31431) chains **AF_ALG** and **splice()** through a logic bug in the kernel's **authencesn** template into a controlled 4-byte write into the page cache of any readable file. It corrupts the in-memory view of a setuid binary such as **/usr/bin/su** and escalates to root with the on-disk file unchanged. The original public exploit is a 732-byte Python script covering Ubuntu, Amazon Linux, RHEL, and SUSE, and CISA added it to the Known Exploited Vulnerabilities catalog after in-the-wild exploitation reports. **DirtyFrag** extends the class into the networking stack, using XFRM security associations over **AF_NETLINK** or an **AF_RXRPC** fallback to clear root's password field in **/etc/passwd**. Neither DirtyFrag path depends on **algif_aead**, so hosts that only blocked that module remain exposed.

Detection targets the primitive rather than the implementation, since Copy Fail already has public Python, Go, Rust, C, and Metasploit reimplementations. Auditd surfaces socket calls with a0 values of 26, 21 plus splice from non-root processes, or a network socket bound with AF_ALG, in the burst both exploits produce, where Copy Fail iterates 48 times to write 192 bytes. Mitigations carry costs, since module blocking can break IPsec VPNs, dropping the page cache spikes I/O, and namespace restrictions affect container runtimes.



Beyond the CVE queue

Elastic observed 20 or more high or critical CVEs across July and August, including an NGINX heap overflow (CVE-2026-42533), silently patched OpenSSL HollowByte, Zoom zero-click RCE, Rails (CVE-2026-66066), and more kernel defects. Coordinated research, rather than attacker discovery, drove most third-party product and API vulnerability reports Elastic handled. To keep advisory drafting from becoming the bottleneck, Elastic built **a generative AI agent in Elastic Agent Builder** that drafts CWE classification, CAPEC methodology, CVSS scoring, and mitigation guidance from raw reports, grounded in crawled MITRE CWE and CAPEC indices so it cannot invent IDs; ESA-2026-01 is already in production.

Not every confirmed intrusion needed a CVE. Elastic threat hunting confirmed ScreenConnect remote access delivery through a newly registered phishing domain on August 5, 2026, and active payload execution from a zero-day-old domain, **fabric-anvil[.]com**, on a macOS endpoint two days later.

DEFENDER TAKEAWAY

CVSS alone did not set this quarter's priorities. Copy Fail earned attention because exploitation needs only legitimate kernel interfaces, local execution, and short public code. With 20 or more high and critical CVEs in two months, reachability and evidence of exploitation set the order. A web server or PHP runtime spawning a shell is actionable, whichever CVE opened the door, and syscall-pattern detections outlive exploit signatures. Assume remediation is incomplete wherever a mitigation was scoped to one exploit or path, since blocking `algif_aead` leaves DirtyFrag open and `wp-content/plugins/` detections miss the `INTO OUTFILE` webshell. Silently patched defects such as OpenSSL HollowByte never reach an advisory feed, so dependency version tracking has to catch them.

Identity and AiTM

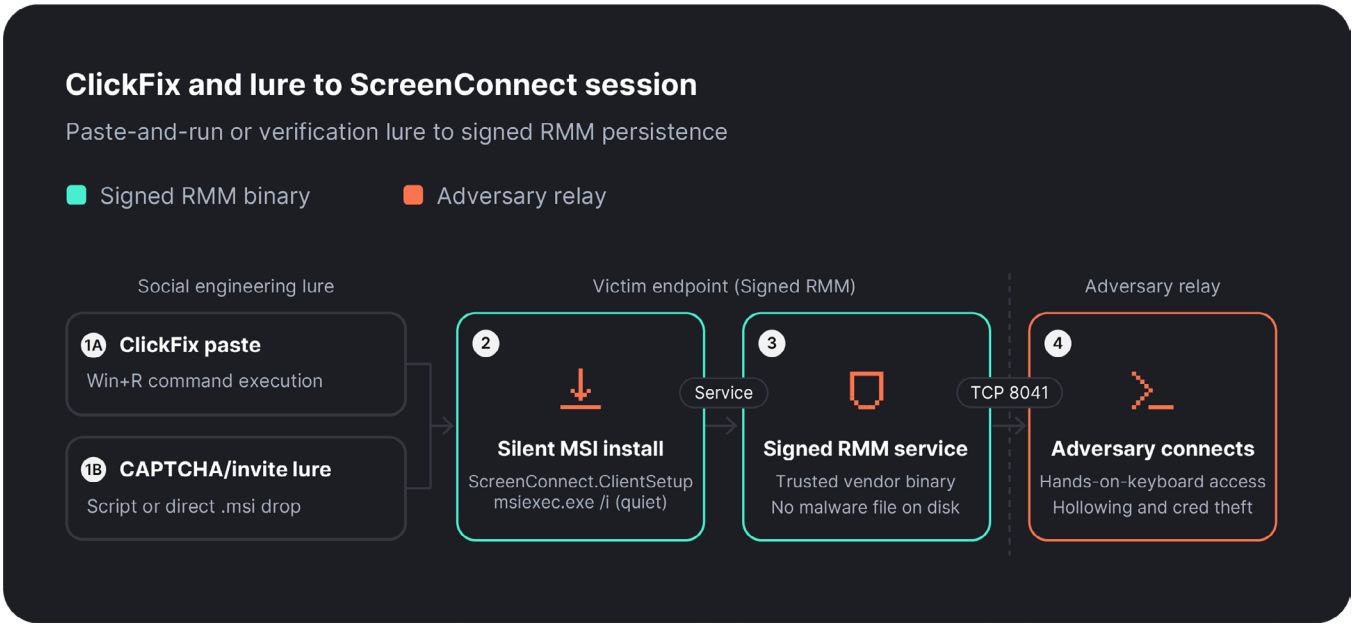
MFA is routinely bypassed when adversaries turn trusted identity mechanisms and preapproved applications against the enterprise. Findings from this reporting period found adversaries intercepting sessions via Adversary-in-the-Middle (AiTM) phishing kits, as well as bypassing interactive prompts entirely by exploiting legacy authentication flows. Beyond initial access, we observed a critical escalation tactic where attackers registered virtual devices to secure persistent access.

The proxy in the middle, and the cheaper path around it

Software-as-a-Service (SaaS) suites with integrated identity provider (IdP) services, such as Okta, Microsoft 365, and Google Workspace, are heavily targeted. Adversary-in-the-Middle (AitM) phishing kits put a reverse proxy between the victim and the real login page. The victim types credentials and completes the genuine MFA challenge, and the relay completes the session token workflow with the IdP.

Tycoon 2FA, first observed in August 2023 and attributed by Microsoft to Storm-1747, is the best-known kit of this type and remains in active use after the March 2026 seizure of over 300 domains. The relays behind these kits include established infrastructure, geolocation matching, and heavy tailoring to the identity provider they interact with. With Tycoon, a primary refresh token (PRT) — a highly effective persistence token in Entra ID — was issued within minutes of credential submission when analyzed. Tycoon is but a fish in a pond of other phishing kits that are continuously emerging, developed, and serviced to adversaries. Elastic Threat Command continuously monitors prominent phishing kits observed in-the-wild (ItW) via **ANY.RUN trends**.

In Elastic telemetry this quarter, operators also used valid credentials over a legacy authentication flow. As the revolving doors of Microsoft identity settings, features, services, and more continued to spin, we observed adversaries taking advantage of grey areas where legacy meets modern authentication.



A university account, ROPC, and Azure Resource Manager

On July 9, 2026, Elastic Threat Command threat hunting confirmed a compromised university account reaching Azure Resource Manager from an anonymized IP known to be malicious. Nothing executed on any endpoint, so no host artifact anchored the analysis. The authentication protocol was ROPC, the resource owner password credential grant, which submits a username and password straight to the token endpoint. The Azure AD PowerShell client carried the request, a legitimate first-party identifier that no application allow list would typically reject.

Further analysis of authentication results and successful sign-ins showed this was not spray noise resolving into failures; every successful attempt returned a token, the signature of credentials already in hand. Conditional Access Policies (CAP) were bypassed, since ROPC never traverses the surface the policy governs. The destination was Azure Resource Manager, the control plane managing subscriptions rather than a mailbox or Microsoft Graph. Elastic Threat Command continues to observe nefarious ROPC traffic, often with BAV2RPOC user-agents.

The same shape recurs across late June and July; the source is repeatedly logged as **Anonymized IP from hosting provider**.

When a device code becomes a device

On June 1, 2026, Elastic Threat Command threat hunting confirmed a Kali365 device-code compromise that ended with a registered device and a primary refresh token. The user arrived through a shared Evernote note and Aha page, both of which handed the session to the Microsoft Authentication Broker. The kit opens a device-code request with the broker client ID specified and sends the user to Microsoft's own device-login page with a code to enter. That page asks the user to sign in and confirm they authorize the broker client with access to the intended resource: Microsoft Graph. The resulting token is issued to the client that opened the request, a.k.a. the session the kit generated.

A Chrome session from AT&T reached that broker for Microsoft Graph, but Entra ID returned error 50199, "user confirmation is required for this request." After it succeeded, we observed requests from Tencent with a Python user agent that made successful API calls to Microsoft Graph with the same broker client, using the device code protocol, confirming the successful device code phishing grant.

The same poller (phishing kit) kept redeeming what the broker had issued before, suggesting scripting behavior baked into the relay. Another Tencent address presented a refresh token for Windows Azure Active Directory after adding an Azure AD-joined virtual device. This indicates the device was used from F.N.S. Holdings addresses, later minting another token for OfficeHome with the PRT it created after the virtual device registration. We then observed device code grants from Allo Communications; however, we identified failures for an invalid and revoked grant.

DEFENDER TAKEAWAY

Legacy authentication is a high-value control here. The activity Elastic confirmed was valid credentials over a flow with no second factor, and scoping that flow removes the cheapest route observed. Authentication protocol, authentication result, and whether all sign-ins succeeded are queryable in any tenant today.

A first-party application identifier is not a trust signal. Several compromises used legitimate Microsoft clients, so geolocation anomaly plus flow type did the work an allow list could not. Extend detection past the login event to the session. Alert on anomalous device registration, especially if not managed or enrolled in MDM and PRT issuance, and use domain age as a triage input on delivery infrastructure. Speed is another factor to consider. Phishing kits and respective relays/infrastructure obtained a PRT within minutes of initial phishing signals.

Resistance efforts such as MFA, conditional access policies (CAP), MDM solutions, and others are proving ineffective against downgrade attacks, phishing with first-party OAuth workflows, and metadata. User consent remains high on adversaries' priority list. Historical uses of signals like geo-location anomalies are irrelevant with modern features baked into phishing kits. Detection signals require a deep understanding and baseline of expected identity flows in tenants to identify account takeover (ATO). It's not just at the IdP layer either, with hybrid environments where Entra Connect Sync servers live, or endpoints using Web Account Manager (WAM) to store authenticated sessions for Microsoft 365 applications; visibility across all data planes is needed.

Forecast

We have four calls for the next two quarters. Each ties to something above, and each includes what would change our mind.

Blockchain-resolved command and control becomes standard in commodity loaders.

Our previous report asked whether this would leave the crypto sector. It did, inside one quarter, and it now shows up in a package a general developer would install. A dead drop costs one transaction and cannot be seized. We would revise this if the technique stays with the same small set of operators, which is possible, since three cases is a direction rather than a trend.

Agentic and LLM applications become a routinely abused execution path rather than a novel one.

Since the number of endpoints running an agent with shell access is climbing fast, and agentic applications need no exploit, malware, or privileges the user doesn't already have, this threat vector will keep growing. We're seeing LLM providers ship new models designed to defend against misuse, which highlights security priorities.

The human in the loop keeps consenting, and the kits or agents keep the speed.

We expect tooling, phishing kits, and AI to keep doing post-consent behavior effortlessly. First-party metadata and an allow list will trust a device and a PRT within minutes, sometimes a downgrade the policy never sees, and access brokered into Office or Microsoft Graph. CAPs and MDM may prove ineffective as long as the victim brokers consent. Agentic identity is where consent goes next, a judgment of trust that agents will do "the right thing." An inherited trust that can have severe consequences when telemetry is the same as an application or script, so observability by identity is key. We expect to be writing about that grant again, and about legacy authentication flows never quite deprecated.

Paste-based social engineering keeps expanding to new platforms.

It moved decisively onto macOS this window with a chain that defeated Gatekeeper using two built-in utilities. Mobile and Linux developer workstations are the obvious next surfaces and have already been observed in the wild.

Research driven by this data

The Elastic Threat Command research below is behind this edition.
Every publication listed here is public.

Agents vs. agents: how we triage HackerOne reports for $2 each, 85% as well as a human

AI-ASSISTED OFFENSE

Benchmarking the Agentic SOC: How we evaluate LLMs for security workflows

AI-ASSISTED OFFENSE

CI/CD pipeline abuse: the problem no one is watching

AI-ASSISTED OFFENSE SUPPLY CHAIN

ClickFix to Cash-Out: Anatomy of a Mexican Banking-Fraud Toolkit

INFOSTEALERS

Copy Fail and DirtyFrag: Linux Page Cache Bugs in the Wild

VULNERABILITY EXPLOITATION

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

IDENTITY/AITM

Elastic releases detections for the Axios supply chain compromise

AI-ASSISTED OFFENSE SUPPLY CHAIN

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

AI-ASSISTED OFFENSE

From Invitation to Infection: How SILENTCONNECT Delivers ScreenConnect

RMM/DUAL-USE TOOLING

From vulnerability report to CVE draft in minutes: how Elastic automated security adv...

VULNERABILITY EXPLOITATION

How Elasticsearch ES-QL COMPLETION turns noisy curl and wget rules into high-fidelity...

AI-ASSISTED OFFENSE

Inside Elastic InfoSec's agentic SOC: How we cut AI agent LLM calls by 60%

AI-ASSISTED OFFENSE

Inside the Axios supply chain compromise - one RAT to rule them all

SUPPLY CHAIN

Living off the coding agent: Two tales of tunnels and LaunchAgents

AI-ASSISTED OFFENSE

Lost in relocation: analysis of a new loader distributing CASTLESTEALER

AI-ASSISTED OFFENSE | INFOSTEALERS

New North Korean campaign uses fake coding interviews to steal developer credentials

SUPPLY CHAIN

PHANTOMPULSE: anatomy of a hijackable blockchain-C2 RAT

AI-ASSISTED OFFENSE | SUPPLY CHAIN

Shai-Hulud strikes again: CHAINDROP worm hits 400+ npm packages

SUPPLY CHAIN

TCLBANKER: Brazilian Banking Trojan Spreading via WhatsApp and Outlook

INFOSTEALERS

TELEPUZ: a modular MaaS malware spreading via CLICKFIX-VIDAR chains

INFOSTEALERS

The Cost of Understanding: LLM-Driven Reverse Engineering vs Iterative LLM Obfuscation

AI-ASSISTED OFFENSE

The security signal log tailing can't see: tracking npm cooldown removals with Elasti...

SUPPLY CHAIN

TOLLBOOTH: What's yours, IIS mine

RMM/DUAL-USE TOOLING

wp2shell hits WordPress: detecting pre-auth RCE from plugin drop to command execution

VULNERABILITY EXPLOITATION

About this report

This edition covers activity Elastic Security Threat Command observed between April 4, 2026, and September 8, 2026. Findings come from Elastic telemetry and from Elastic Security Threat Command threat hunting and research and are published alongside the research that documents them.

How to read the numbers

Counts in this report are observations from Elastic telemetry inside the edition window. They are not market share, not a census of global activity, and not a measure of detection efficacy. A count of confirmed findings reflects what Elastic saw and confirmed in that period, and nothing beyond it. Several findings can share one host or one tenant, so a count of findings is not a count of victims.

Where a claim rests on published Elastic Threat Command research, that research is linked at the claim. Where it rests on Elastic's own observations, it is attributed to Elastic Threat Command in the sentence. Host, account, and customer identifiers are removed before analysis.

About Elastic Threat Command

Elastic Threat Command is the threat research arm of Elastic Security Labs, publishing open research on malware, adversary tradecraft, and detection engineering, along with the detection rules and artifacts behind it.

This report is for informational and defensive security purposes and reflects Elastic Threat Command's assessments based on evidence available at publication, and telemetry provided voluntarily by users. Assessments, including attributions and AI-involvement findings, may change. Product and company names are used for identification only. They imply no affiliation or endorsement, and do not mean the named products or companies are not secure, were compromised, or acted improperly. References to legitimate software describe misuse by threat actors after access was obtained. Indicators of compromise support detection and hunting, and do not imply an owner is malicious or complicit. Validate before blocking. Descriptions of Elastic product behaviour reflect the specific events or conditions described and are not guarantees of detection or prevention.

threats, etc quarterly cybersecurity report

NOV 2026



© 2026 elasticsearch B.V. All Rights Reserved.
Elastic, Elasticsearch and other related marks are trademarks, logos or registered trademarks of Elasticsearch B.V. in the United States and other countries. Microsoft, Azure, Windows and other related marks are trademarks of the Microsoft group of companies. Amazon Web Services, AWS, and other related marks are trademarks of Amazon.com, Inc. or its affiliates. All other brand names, product names, or trademarks belong to their respective owners.