

Fake Booking.com ClickFix attack abuses Cloudflare verification to deliver malware

 threatlocker.com/blog/fake-booking-com-clickfix-attack-abuses-cloudflare-verification-to-deliver-malware

January 23, 2026

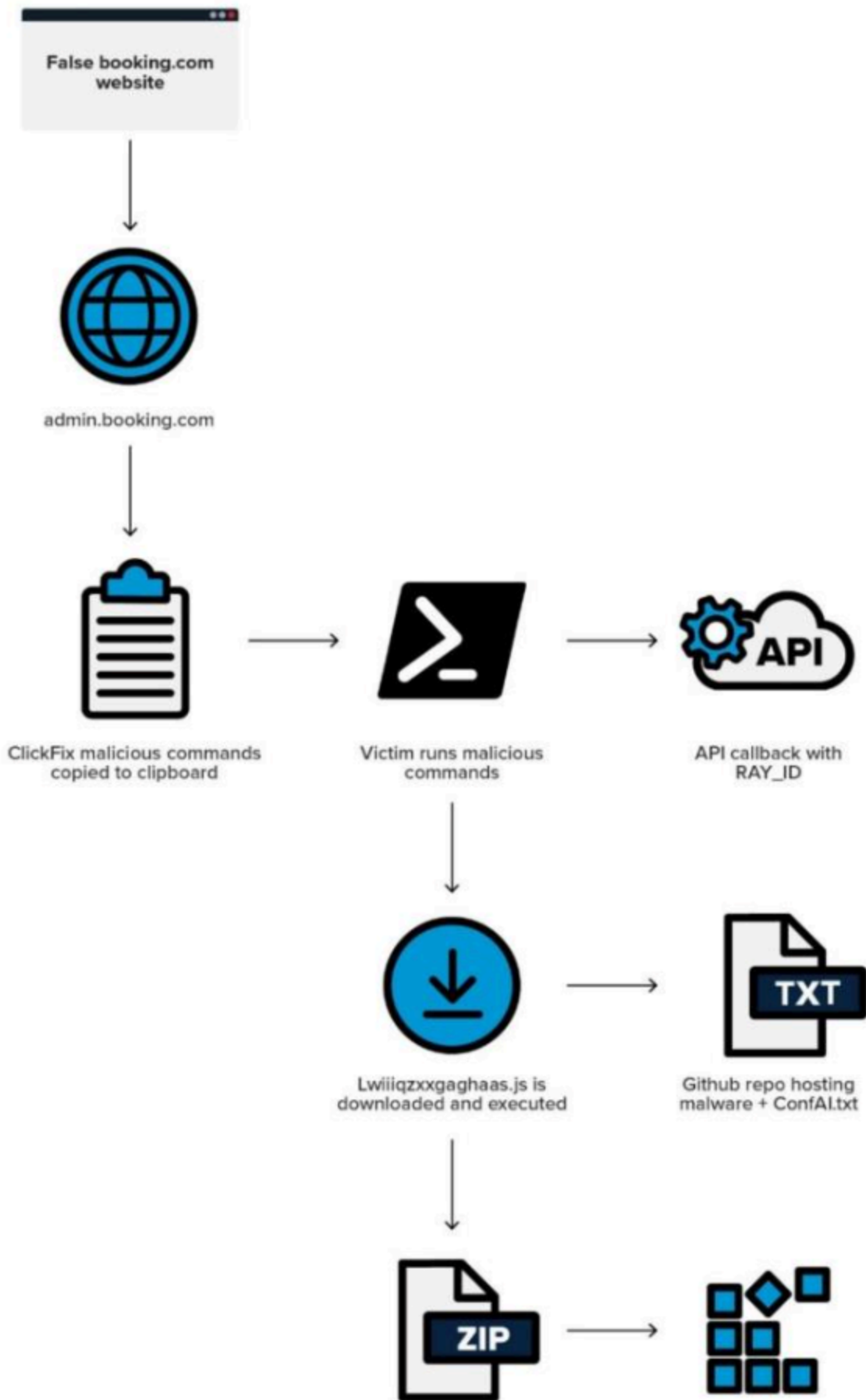
Introduction

ThreatLocker Threat Intelligence identified an active “booking.com” ClickFix website leveraging an attack similar in nature to previously documented BSoD “booking.com” websites. By using a false Cloudflare turnstile verification page, victims are coerced into executing malicious commands on their devices, leading to the eventual deployment of malware. Stages and additional payloads delivered through this campaign were hosted on a public GitHub repository, which also hosts several files relevant to other campaigns.

Attack Path Overview

This campaign targets victims who are trying to make travel or hotel arrangements. Visitors are redirected towards a malicious “*admin[.]booking[.]com*” sub-domain belonging to “*id0718[.]com*”.

When the victim is deceived by the ClickFix turnstile and has executed unknowingly copied malicious commands, the first stage payload is retrieved from the GitHub repository and executed. This initial payload downloads and installs the final stage payload as well as other dependencies, such as Node.js and the Tor binary. This attack is very similar to a [previous ClickFix campaign](#) we’ve covered and uses a Cloudflare verification to lure users into entering malicious commands.



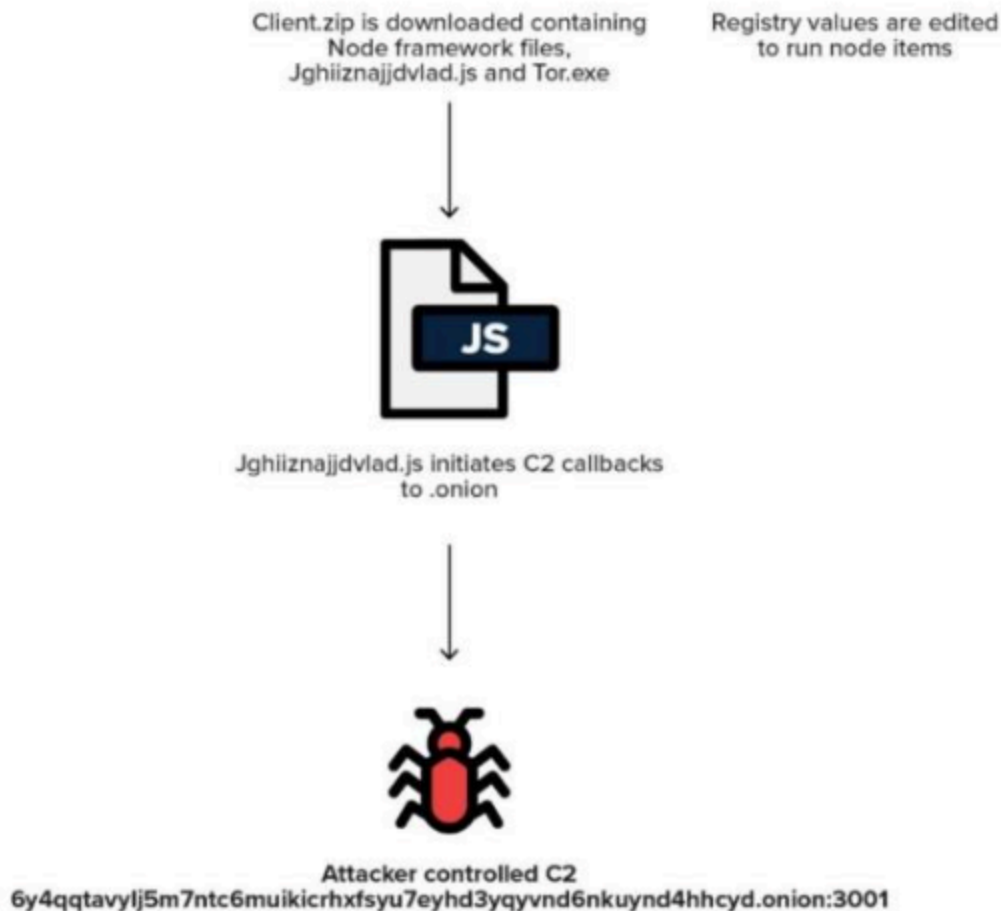


Figure 1: Attack path overview

ClickFix Coercion

The malicious subdomain visually mimics a Cloudflare turnstile verification page, requiring the victim to click the familiar checkbox before continuing to their desired page. When this checkbox is clicked, commands are copied to the clipboard without the user's knowledge.

Figure 2: Fake Captcha Error

Seemingly legitimate instructions appear within the webpage instead of the expected turnstile logic, which normally does not require any further user interaction. The victim is prompted to open the Windows Run dialog box with the "Win + R" keys, paste malicious commands copied to their clipboard with "Ctrl + V", and invoke them with the "Enter" key. Once the copied commands are supplied to the Run dialog box, all user interaction required for this browser-based social engineering attack is complete.

Figure 3: ClickFix Instructions

Continuing the deception, the copied commands open a PowerShell console on the victim's desktop, displaying several messages that show "progress" regarding Cloudflare's validation. In the background, the Powershell cmdlet "Invoke-WebRequest" downloads a malicious JavaScript file "*lwiiiqzxxgaghaas.js*", and the Windows built-in binary wscript.exe is used to execute it, leading to the first full payload of this attack. The screenshot below shows the copied ClickFix script modified with *<TL>* comments to reveal what is being run behind the misleading console window.

Figure 4: ClickFix Execution with Added Comments

Pasting the malicious content into a text file shows that the ClickFix payload displays ASCII art of the word "Cloudflare," downloads the contents of the first stage from a GitHub page, and saves it to the user temporary directory. Wscript is used to execute the downloaded JavaScript file.

Figure 5: Unknowingly Copied Commands

The HTML within the Clickfix webpage captures the "CF_RAY_ID", a value within the browser's local storage, and randomly generates one if not found. This variable is regularly used by Cloudflare, however, in this instance it's being leveraged as a unique identifier cataloging victims when calling back to the API.

Figure 6: Webpage Ray ID Generation

Curl is used with the "-s" (--silent) option and output is piped to "\$null" to perform a silent HTTPS request to "*api[.]id0718[.]com/m/api/v1/callback?ray_id=<rayid>*". While this is likely not an actual endpoint on this domain, Threat Intelligence suspects that this callback request is used to track successful ClickFix compromise.

Figure 7: API Victim Callback

Lwiiiqzxxgaghaas.js

The primary purpose of the first stage payload is to download and extract additional files (including the final stage payload) and establish persistence by creating registry keys. The first task that "*Lwiiiqzxxgaghaas.js*" performs is to reach out to the malicious GitHub page and download the contents of "*ConfAI.txt*" into memory. This file holds variables for the Node.js framework, a second file from the GitHub page named "client.zip", the filename of the next stage, and a registry path that will later be written to "HKCU\Software\Microsoft\Windows\Current\Version\Run\MyNodeStartup".

Figure 8: Contents of ConfAI.txt file

After parsing "*ConfAI.txt*" for all required variables, the legitimate Node.js package (version 22.19.0) is downloaded, extracted into the user "AppData\Roaming" directory, and deleted.

Figure 9: Lwiiiqzxxgaghaas.js

Finally, the last payload (contained in client.zip) is downloaded and extracted into the newly created “node-v22.19.0-win-x64” directory, and client.zip is deleted. The final payload, “Jghiiznajjdvlad.js”, is executed with the existing WScript shell.

The last action performed by this payload is to establish persistence by using the RegWrite shell function to write a command into the previously mentioned “MyNodeStartup” registry key. First, a series of commands is written to a new file “Node.js”: a WScript shell is created, paths to the “node.exe” executable and the final stage “Jghiiznajjdvlad.js” are defined, and the “shell.Run” method is used to execute. To provide visual clarity, the “node.js” file has been recreated.

Figure 10: Contents of Node.js

Once “node.js” has been created, the command is written to registry. For clarity, the command written to registry is effectively: ‘wscript.exe " ..\%appdata%\node-v22.19.0-win-x64\node.js”’.

Figure 11: lwiiqzxxgaghaas.js Payload Execution

Jghiiznajjdvlad.js

The final payload defines several functions designed to extract the victim’s IP address, hostname, OS type, and locations of the AppData and User Profile directories. The main function establishes a local Tor socks proxy and uses it to connect to an embedded “.onion” link, which had been taken down at the time of discovery.

Figure 12: Local Tor Socks Proxy

Upon connection, the script emits a registration message with the captured victim details. Two notable features are provided. The first allows attackers to transfer files to the machine and execute them with several methods such as *InstallUtil*, *RegSvr32*, *RunDLL32*, and reflective assembly loading.

Figure 13: Emit Registration Message

Figure 14: SaveAndExecuteFile with Execution Methods

Figure 15: Reflective LoadInMemory Execution

The second feature allows for arbitrary execution of commands with the NodeJS “exec” function, spawning a single-use CMD shell for each command. In combination with the registry change made by the previous payload, these commands provide the attackers with a method of direct execution for every user login.

Figure 16: Node Arbitrary Execution

Other Repositories

The URLs used to download malicious payloads point to a GitHub account that has three publicly available repositories, with each hosting a collection of malicious files. These repositories were recently updated, showing that this user is still actively developing these collections. Content stored in the “Files” repository includes a variety of malicious software and offensive tooling, such as IDAT Loader, PureRat, PureLog Stealer, StealC, Cobalt Strike, and Remcos Rat. Several instances of these files were found obfuscated with commercial software protectors, including .NET Reactor and VMProtect, posing a substantial hurdle for analysts and responders alike. These samples will be covered in greater detail in our follow-up post.

Figure 16: Main GitHub Page

Figure 18: Files Repository

Figure 19: <user>.github.io Repository

Figure 20: FileX Repository

Conclusion

ClickFix attack chains heavily rely on the initial stage being executed by the victim, a common tactic observed in social engineering-based campaigns. By presenting a familiar page, victims may quickly overlook otherwise obvious indicators of a malicious website and become coerced into entering malicious commands.

It may be easy to dismiss these attacks and state that they are easily avoided. However, widespread incidents like these highlight the importance of employee training and user awareness. Familiarity with tools at hand like PowerShell and the Run dialog box are crucial first steps to implementing a Zero Trust approach to your organization’s security. ThreatLocker Threat Intelligence continues to track related infrastructure and payloads tied to this campaign, with further technical analysis and insights to be covered in an upcoming follow-up blog.

IoCs

Hashes

SHA-256

01dc05218cb437f664bf4659a4e17bb0fa619d1108cba75b0586f30616c6dad7

0364ccd4d33d20a30c7a3c372ee8bdaad1dcda220322bceac7fe1338be844b43

08a5d0d8ec398acc707bb26cb3d8ee2187f8c33a3cbdee641262cfc3aed1e91d

SHA-256

119d7d0830e42ce0ccd55f4c0c68159a5cca2925887c3aff8aca1fa6513551b2

14bd9f218bca5efe13a69da6ae3c60756f036ede6526a67e8baddc1117b70def

16773927dfe08ced23ac7b9f64741f2688ab45f81ad124679f73f8900ddd7cc2

16bf64dd808678661774fdf88a95366898ae36a3bff70117f57a9f1745738eb

19cb261169990706041fe819f7d60be000d5c3e0f4d92be114827309cc8aafd6

221e280026b6a95cb4e412c6b49d40369ad919f7c63d5431318d6252340b0352

225a40104f94dc247ea62de47f600eac74c7c6b1f5964aeebeac2670e2eea878

2805ca562b170c64c45a57756fde4d2988e7470a96c9c62b573cba0ee9c245d8

2cc30e8dc3265b550825fb538e5a3c09d41f09db1ffa8d4a3c42024eef0a7233

2cdcb205931930acb0c2f5495b5d8f0909ecb6c274d5333bb35ce138dad8102c

307bb0547ae8b747cf0376e29d4f7a528f76a31c2df16bf9165ba83bdf97315c

31a697d77dfb56f99f0cdd2ab1772c26f8bc0ad2b8f5da9da2ccf7fcf2721218

31c4d219371336e6ecc39dcc21711eab8bb4829795fc5082ea022eefc1caa675

34a66cb4c33453d8f01ae9b57abb470d980c2e402dedc11b5d9599753e073f16

34a923a743a6fe56ff6e0bbfb0ec4e51525dca31ebe2865aadcd25291949cc37c

36ca725ffdc27f2a057b83c9862abda60036920b265ac68d3cc5a67e40a81083

3a75d8a8a2fa457f13f228f588cc3304b407cc54dd36aab862a393368a93f19f

3e376324369088d3a848d253b8f9e08b90a7a549a143e758433b1265d0f34a29

40a3dcfe5184cfaa4e1fd2633e2f2278e8049e839ca662ade7ba8ebd659a37b7

428e087bcfccd102189a2e350a35b328c1a4ac3be2f6edc043e5d8082f529693

445a097d665db3c5ac55cff1cab4e5c459e6fbe65a469cf96c07af83037b7ad7

4843b91c6093e5a917c30449db00d4c006227561284ea1dac77f1b325795517e

4a916d14d4cc997a862da720eca0bd5f2bceb8df81bcbcf3d24009dc24c7cd4c

4c194fa76095d46b0e286f7dead970ef15d8b70d5cf579a96385db71c5e4cfb1

511bb467991f5f24578beae81d945b2ae26d0aa25e8acbf4cf69a630be1904b1

SHA-256

57916bb2fc6c674d608c5d0c829e12790560826e9b612bf4aacdfc82f8acc8d9

57b8080f989e0cd81c06c1c231eb5e52cbeed2c8d14ea6b0b842791612051cdd

5b281e0a7cead02b2b4e6683509239362b95d069c93bbd3a629faf30b617ddab

5e061b9648924bad9f1f0ee38f1e8a01c0468b06e29ef1e792329ec914a58ac2

6195e13fed37aa3a4da485d52dc108ad49ac033d602c8402e0d3733264beff51

628b3d1725dd4d33e8e2a4052a3f91603088198c8629d6aa386a868977c49879

6660c69660c49fab056b872efb250badea61ae1f62cc4e460bc6c974b6d127f1

6cd36db0d5f0d7add5d4430ce9094ac1dc2ca16655f21766a550c7b8ae179194

70cb4ff5f30014ed63c21c94c0daaf2d9f23d1895b654d60c2daa91ab54f841b

796f06c18310d4a37ea1d6c1917e7fdc6b2e739978b75aec441ab23d4573bfe

7d4783295cf3bddcb58cf0f73beeadf99c5c6e4ab3f44259f87bdc98b8997b47

7dab1281c8876e56a29974a5f25939c2354bca6004dde31c685da4f91928203f

7df4290325a19e0d34d803bc8445ecb21359f935b1433e5c3e47c93287b06d63

7e203002c32a99fd7e82d52cb5419f604b5e812d3faf6f07b4166e4ee718bde1

84061ee691cd8795e6ed14e98c4d3a3ad87517330249d82a75c8a7ff95873773

84931e32aa3a27bfd4bfde88412a23bffdd6cd94eeeafdf91fbc49229da48f67

89b90da48fa73a4557b25d5acb673824135c8d513f7ee915782d9d6717b6ab36

8c32b28d5c217f2c51fb1233527b31c8ef447eddddbf45f9005406dbc355c182

9bac5331260f9b10ec17d94876fa28733fae5a7a17a8af23d0517a6c37512508

9bbcade7f7a36dc2dbc76a50e2ef6fe951738f6d99c12fb9c553f469428bace0

9cc2d4137a7823a889de6b5aaf12c54e1449406056e5e5ec820c8cdac1696749

a5ebb9d2e75e5ef497613a935e48888e7eb8b5ae80a9c087f204868aff93e311

b19e017b3a2215a548f70b8bbcb3793bc99b99cdf8ee550fecc3bb2496a5fa29

b30160e759115e24425b9bcd606ef6ebce4657487525ede7f1ac40b90ff7e49

b3a437e1ed42f62277861a9e5d0b70ab54786766a9b0299b0ef5ee0cd646b9a8

SHA-256

b5b7c09b3411d0b847bdd9c786d904c83d78fa82f67d54a8680c08ecd2f7c9e2

b783a18411be49d8d9e938dcef2f03ed9c9cb7e74781a7a0ee9d2b4e46a5c7ad

b8e096b46ef9df2a46d66d9988211aa6f6a54b406ebb18cafa629f69c7682b6a

b8eaffa62c7a56f8f4ee579a290b1af6923c5ca632a6e9c9eb4cc5fc1768d3e8

c208d8d0493c60f14172acb4549dcb394d2b92d30bcae4880e66df3c3a7100e4

c427fda1986e7b5a7006b4a95a782e5736297a92fdc950509bcb24d3e8965d24

c8d5572ca8d7624871188f0acabc3ae60d4c5a4f6782d952b9038de3bc28b39a

c9ab0945fec5b1690c3c267f736c932018e6efc564016b3dd52894dc77d2072f

cb91a991908c052790f6b2ffcd5ecf13f04641f30f38f54ee996359a4da5eada

cc1e2fe61480c59f973359d50ba6242e12c1b8157c4889a05ae83cf3ead0e71c

ce8a9a01cea73e501b37f74e075194b33afc01f6562b15837cd155e62cef3796

cf3d3912404edd93774fa0a441771daf162b2977cd7138b9214c1824431b35a5

d217a19f0a8ff60e5289e9cdf61146df6041e6164aac01ce330cf593f18de962

dfbc97ffa63457b24af333e1b1d05b6fe3a61f214abf3e4e0c5b55a2a9a20bdb

e3f3c6161ed2ec8f175a18b7713a77bd35e59badc87e5e356aac116c40da5834

e69d9d0711abef6801540c7f088792e5982a321ef77b06c864f6ac1e51ae2c83

e6e4d308a93467772e0cc12d13d44a75acb68f2abf2a5d7d171bc3061ea6436c

e9df5386204c4072078126d46a6ba453a469656e51702263c2aa989976eb417f

f0e54788771931aa10bf059e9d913ad363d72168fe8120bde2f3a1e7d15b6bf6

f70fa01eb5a6c348013f7a582b7cb55cbb8334afc8a54c7621d206333b1d18e7

Domains

hxxp[:]//sberbank[.]sberbank[.]admin[.]booking[.]id0718[.]com

hxxp[:]//pay[.]nalozhka[.]admin[.]booking[.]id0718[.]com

hxxp[:]//nalozhka[.]nalozhka[.]admin[.]booking[.]id0718[.]com

hxxp[:]//cdek[.]nalozhka[.]admin[.]booking[.]id0718[.]com

hxxp[:]//sbermegamarket[.]avito[.]admin[.]booking[.]id0718[.]com

hxxps[:]//pixeldrain[.]com/api/file/dyEewy42

hxxp[:]//cdek[.]sberbank[.]admin[.]booking[.]id0718[.]com

hxxp[:]//youla[.]pochtabank[.]admin[.]booking[.]id0718[.]com

hxxp[:]//youla[.]nalozhka[.]admin[.]booking[.]id0718[.]com

hxxp[:]//sberbank[.]pochtabank[.]admin[.]booking[.]id0718[.]com

hxxp[:]//publicyoula[.]admin[.]booking[.]id0718[.]com

hxxp[:]//youla[.]avito[.]admin[.]booking[.]id0718[.]com

hxxp[:]//cdek[.]admin[.]booking[.]id0718[.]com

hxxp[:]//youla[.]admin[.]booking[.]id0718[.]com

hxxp[:]//ozon[.]admin[.]booking[.]id0718[.]com

hxxps[:]//pixeldrain[.]com/api/file/cUzffqau

hxxp[:]//pochtabank[.]pochtabank[.]admin[.]booking[.]id0718[.]com

hxxp[:]//avito[.]admin[.]booking[.]id0718[.]com

hxxp[:]//nalozhka[.]sber[.]admin[.]booking[.]id0718[.]com

6y4qqtavylj5m7ntc6muikicrhxfsyu7eyhd3yqyvnd6nkuynd4hhcyd.onion:3001

IPs

185[.]163[.]204[.]176

185[.]201[.]252[.]130

How to prevent ClickFix-style attacks

These attacks rely on social engineering rather than technical exploits, meaning awareness and behavior-based controls are critical to prevention.

Individual users should be cautious of any website that:

- Requests you to manually paste commands into the Windows Run dialog or PowerShell

Claims action is required to complete CAPTCHA, Cloudflare or security verification

Displays instructions instead of performing an automatic verification check

For organizations:

Train employees to recognize social engineering techniques that abuse trusted brands and security prompts

Restrict or closely monitor PowerShell, script bins, and LOLBins like wscript.exe

Enforce least privilege and application control policies

Recommendations for ThreatLocker customers

You can use ThreatLocker solutions to reduce the risk of a ClickFix-style attack by leveraging layered controls designed to limit initial access and post-execution activity.

Application control

Helps prevent the attack from progressing beyond the initial users by blocking unauthorized payloads, including secondary-stage scripts, binaries, and dependencies downloaded after the first execution.

ThreatLocker Detect & Cyber Hero® Managed Detection and Response

Provides visibility into malicious behavior and known IoCs associated with ClickFix campaigns. Alerts can be reviewed internally or by Cyber Hero® MDR analysts to assist with investigation, containment, and response.

Web control

Mitigate the attack at the earliest stage by blocking access to unapproved websites, including known malicious domains.

FAQs

What is a ClickFix attack?

A ClickFix attack is a social engineering technique that tricks users into executing malicious commands. Instead of exploiting software vulnerability, it convinces victims to copy and paste commands into tools like Windows Run dialog or PowerShell.

How does the fake Cloudflare verification work?

The malicious site displays a page resembling Cloudflare's Turnstile or CAPTCHA verification, but instead of performing a real verification, the page secretly copies malicious commands to the

user's clipboard and instructs them to paste and execute those commands manually.

Why does the attack ask users to press Win + R and paste commands?

Requesting users to open the Windows Run dialog and paste commands allows attackers to bypass traditional browser-based protections. Because the victim executes the commands themselves, this technique can evade some security controls that rely on exploit detection rather than user behavior.

Why are GitHub repositories used to host malware?

Platforms like GitHub offer attackers a trusted hosting environment that blends in with legitimate traffic. Using GitHub can help attackers avoid reputation-based blocking and make malicious downloads appear less suspicious to both users and security tools.

What kind of access do attackers gain from this malware?

The final payload allows attackers to gather system information, execute arbitrary commands, transfer files, and maintain persistence across reboots. Communication is routed through Tor, making tracking and attribution more difficult.