INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 07:12 UTC

ClickFix Lure Adopted by Lorem Ipsum Malware Campaign With Possible Vice Society Attribution

THREAT CAMPAIGN | **HIGH** | CVSS 5.0

SCC Item ID	SCC-CAM-2026-0492
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	5.0
Affected Products	End users visiting compromised WordPress sites (WordPress used as delivery infrastructure); Windows endpoints targeted for payload delivery
Published	2026-06-16T11:10:48
Discovery Source	Rss

Executive Summary

A malware campaign dubbed 'Lorem Ipsum' is using compromised WordPress websites to trick employees into manually running malicious commands on their Windows computers, bypassing standard security tools. Possible attribution to Vice Society, a threat actor known for ransomware and data theft targeting education and healthcare, elevates the business risk, though attribution remains low-to-medium confidence pending independent corroboration from higher-tier threat intelligence sources. Organizations whose employees browse the public web on Windows endpoints face exposure to ransomware deployment and data exfiltration without requiring any software vulnerability to be present.

Technical Analysis

The Lorem Ipsum campaign uses ClickFix social engineering (MITRE T1204.002, T1566) to deliver malware via compromised WordPress sites (T1584.004, T1189). Victims encounter a fake error or CAPTCHA prompt instructing them to paste and execute a command via the Windows Run dialog or PowerShell (T1059.001), bypassing browser-based and email-based defenses. This user-initiated execution method exploits CWE-693 (Protection Mechanism Failure) by circumventing automated execution controls entirely. Post-execution, payloads likely establish C2 channels (T1041) and may deploy ransomware (T1486), consistent with Vice Society's historical tradecraft where confirmed. Obfuscation techniques (T1027) are used to complicate static analysis. No CVE is associated with this campaign; the attack chain relies entirely on social engineering rather than software vulnerability exploitation. Attribution to Vice Society is described as possible and carries low-to-medium confidence pending independent corroboration from threat intelligence sources. Source quality reflects T3 sourcing only (Dark Reading); no T1 (official threat intel, law enforcement) or T2 (established vendor

threat reports) sources were available at time of writing. Readers should monitor CISA advisories and vendor threat reports (CrowdStrike, Microsoft, Mandiant) for higher-confidence attribution updates.

Action Checklist

1. Step 1: Containment. Block execution of unsigned or untrusted scripts launched from user-context processes (explorer.exe, Run dialog). Apply AppLocker or Windows Defender Application Control (WDAC) policies to restrict PowerShell execution to signed scripts. Identify and block known malicious WordPress domains used as delivery infrastructure if IOCs are released by threat intelligence feeds.
2. Step 2: Detection. Monitor for PowerShell or cmd.exe processes spawned by explorer.exe or the Run dialog (Event ID 4688 with process parent correlation; Sysmon Event ID 1). Alert on encoded PowerShell command-line arguments (base64 strings, -EncodedCommand flags). Query web proxy logs for outbound connections from endpoints immediately following browsing sessions to WordPress-hosted sites. Review NIST AU-2 and AU-6 event logging posture to confirm these process-creation events are captured.
3. Step 3: Eradication. There is no patch for this campaign; the attack vector is social engineering. Remove execution capability by enforcing NIST AC-6 (Least Privilege), standard users should not be able to execute arbitrary commands from the Run dialog in production environments. Disable or constrain PowerShell for non-administrative users via Group Policy (Set-ExecutionPolicy RemoteSigned or AllSigned at minimum). Audit and harden any internet-facing WordPress infrastructure your organization operates per CIS 7.3 and CIS 7.4 automated patch management requirements.
4. Step 4: Recovery. For any endpoint where suspicious execution was detected, isolate, image, and perform forensic triage before returning to service. Validate that WDAC or AppLocker policies are enforced and logging is confirmed active per NIST AU-12. Monitor post-remediation for C2 beaconing behavior (periodic outbound connections to uncommon external IPs) for a minimum of 14 days. Confirm no scheduled tasks, registry run keys, or startup items (MITRE T1547) were created during the infection window using Windows Registry forensic analysis and authorized endpoint detection tools.
5. Step 5: Post-Incident. This campaign exposes a control gap in user execution policy enforcement. Conduct a lessons-learned review against NIST AC-6 (Least Privilege) and AC-3 (Access Enforcement) to determine whether standard users retain unnecessary execution rights. Evaluate security awareness training currency, as ClickFix succeeds because users comply with on-screen instructions without verification. Review CIS 6.3 and 6.4 MFA posture for externally-exposed applications and remote access, particularly if Vice Society attribution is confirmed, as Vice Society historically follows initial access with credential theft.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to senior IR leadership and legal/privacy counsel if forensic triage confirms data staging, exfiltration artifacts, or credential harvesting activity on any endpoint, given Vice Society's established pattern of ransomware deployment and data theft targeting education and healthcare — environments likely subject to FERPA, HIPAA, or state breach notification obligations.

Recovery Notes	After reimaging or hardening affected endpoints, validate that AppLocker or WDAC policies are actively enforced (not just configured) by running 'Get-AppLockerPolicy -Effective' or reviewing WDAC audit events in Microsoft-Windows-CodeIntegrity/Operational before returning any host to production. Monitor all recovered endpoints and network egress points for 14 days minimum for periodic outbound beaconing patterns to uncommon external IPs or domains, as Lorem Ipsum payloads may establish C2 persistence that survives shallow remediation if registry run keys or scheduled tasks were not fully purged. If Vice Society attribution is confirmed at medium or higher confidence during the post-incident review, expand the recovery scope to include a credential audit across all systems the affected user had access to, prioritizing externally-exposed applications and VPN accounts.
Forensic Artifacts	Windows Security Event Log Event ID 4688 (Process Creation) entries on affected endpoints showing PowerShell.exe or cmd.exe spawned by explorer.exe or rundll32.exe, with full CommandLine field capturing the ClickFix-instructed Run dialog command — primary evidence of user-executed lure delivery | PowerShell Operational Log (Microsoft-Windows-PowerShell/Operational, Event ID 4104 Script Block Logging) containing the decoded Lorem Ipsum payload script, including any download cradles (Invoke-WebRequest, IEX, WebClient) used to fetch the second-stage malware after the user manually triggered execution | Filesystem artifacts in %TEMP%, %APPDATA%, and %LOCALAPPDATA% with creation timestamps correlating to the browsing session on the compromised WordPress lure page — including any dropped executables, DLLs, or script files deposited by the ClickFix-initiated command sequence | Registry export of HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\Software\Microsoft\Windows\CurrentVersion\Run, and scheduled task XML output from 'schtasks /query /fo XML /v', to identify persistence mechanisms consistent with Vice Society TTPs following ClickFix initial access | Web proxy or Windows Firewall logs showing the endpoint's outbound HTTP/HTTPS connections to WordPress-hosted lure URLs and subsequent connections to C2 infrastructure, with timestamps allowing reconstruction of the full kill chain from lure visit to payload callback

Per-Action IR Details

Step 1: Containment — Block execution of unsigned or untrusted scripts launched from user-context processes (explorer.exe, Run dialog). Apply AppLocker or Windows Defender Application Control (WDAC) policies to restrict PowerShell execution to signed scripts. Identify and block known malicious WordPress domains used as delivery infrastructure if IOCs are released by Dark Reading or threat intelligence feeds.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: For teams without enterprise application control tooling: deploy Sysmon with SwiftOnSecurity config to capture process lineage immediately; use Windows Group Policy to set PowerShell ExecutionPolicy to AllSigned for all non-admin OUs (Computer Configuration > Windows Settings > Security Settings > Software Restriction Policies); block outbound DNS and HTTP to identified Lorem Ipsum delivery domains at the perimeter firewall using IP/domain block lists from open-source threat feeds (Abuse.ch, URLhaus). Two-person team can stage and push the GPO within one change window.

Evidence: BEFORE applying AppLocker/WDAC policies or firewall blocks, capture: (1) active network connections from all endpoints via 'Get-NetTCPConnection | Where-Object {$_.State -eq "Established"}' to identify any live C2 channels established after WordPress lure delivery; (2) running process list with parent PIDs via 'Get-WmiObject Win32_Process | Select-Object ProcessId, ParentProcessId, Name, CommandLine' to baseline whether any PowerShell children of explorer.exe are already active; (3) prefetch files from C:\Windows\Prefetch for powershell.exe

and cmd.exe to establish prior execution history before policy enforcement alters behavior.

Step 2: Detection — Monitor for PowerShell or cmd.exe processes spawned by explorer.exe or the Run dialog (Event ID 4688 with process parent correlation; Sysmon Event ID 1). Alert on encoded PowerShell command-line arguments (base64 strings, -EncodedCommand flags). Query web proxy logs for outbound connections from endpoints immediately following browsing sessions to WordPress-hosted sites. Review NIST AU-2 and AU-6 event logging posture to confirm these process-creation events are captured.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM: enable Windows Security Audit Policy for Process Creation (auditpol /set /subcategory:"Process Creation" /success:enable) and enable command-line logging in process creation events via GPO (Administrative Templates > System > Audit Process Creation > Include command line). Deploy Sysmon with Event ID 1 rules filtering on ParentImage containing explorer.exe and CommandLine containing '-EncodedCommand' or 'base64'. Use the free Sigma rule 'proc_creation_win_powershell_encoded_param.yml' (SigmaHQ repository) converted to a PowerShell script that parses local Security event logs on a scheduled task. For proxy visibility, review Windows Firewall logs at C:\Windows\System32\LogFiles\Firewall\pfirewall.log for outbound port 80/443 connections correlated with browser process timestamps.

Evidence: This is an analytical phase — primary evidence collection rather than alteration. Capture and preserve: (1) Windows Security Event Log (Microsoft-Windows-Security-Auditing, Event ID 4688) entries showing PowerShell.exe or cmd.exe with ParentProcessName of explorer.exe or the Run dialog host (rundll32.exe launching shell32.dll); (2) Sysmon Event ID 1 logs with full CommandLine field, capturing any base64 payloads or -EncodedCommand strings indicative of the ClickFix lure instruction set; (3) web proxy or Windows Firewall logs showing browsing sessions to WordPress-hosted URLs ending in lure page delivery, correlated by timestamp and source endpoint to subsequent PowerShell spawn events; (4) PowerShell Script Block Logging output from Microsoft-Windows-PowerShell/Operational log (Event ID 4104) which captures decoded script content even when base64-encoded on the command line.

Step 3: Eradication — There is no patch for this campaign; the attack vector is social engineering. Remove execution capability by enforcing NIST AC-6 (Least Privilege) — standard users should not be able to execute arbitrary commands from the Run dialog in production environments. Disable or constrain PowerShell for non-administrative users via Group Policy (Set-ExecutionPolicy RemoteSigned or AllSigned at minimum). Audit and harden any internet-facing WordPress infrastructure your organization operates per CIS 7.3 and CIS 7.4 automated patch management requirements.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: For teams without enterprise endpoint management: push PowerShell constrained language mode via GPO registry key (HKLM:\SOFTWARE\Policies\Microsoft\Windows\PowerShell, ExecutionPolicy = AllSigned) across all non-admin workstations using a login script; use the free tool 'PSLockDownPolicy' or manually set __PSLockdownPolicy environment variable to '4' for constrained language mode on legacy systems. For WordPress hardening without a commercial scanner: run WPScan (free tier, wpscan.io) against all organization-owned WordPress instances to enumerate outdated plugins and themes that ClickFix campaigns commonly exploit as the delivery-side compromise vector.

Evidence: BEFORE enforcing new Group Policy and BEFORE hardening or reimaging any endpoint flagged during detection: (1) acquire a full RAM image using WinPmem or Magnet RAM Capture from any endpoint where ClickFix command execution is confirmed — memory will contain decoded payload stages, injected shellcode, or in-memory C2 implants that disappear on reboot or policy enforcement; (2) export current registry run keys

(HKCU\Software\Microsoft\Windows\CurrentVersion\Run, HKLM\Software\Microsoft\Windows\CurrentVersion\Run) and scheduled task XML definitions ('schtasks /query /fo XML /v > tasks_baseline.xml') before policy changes can mask attacker-created persistence; (3) collect PowerShell Operational log (Event ID 4104 Script Block Logging) and transcription logs if enabled, as these capture the decoded Lorem Ipsum payload commands that were manually triggered by the user.

Step 4: Recovery — For any endpoint where suspicious execution was detected, isolate, image, and perform forensic triage before returning to service. Validate that WDAC or AppLocker policies are enforced and logging is confirmed active per NIST AU-12. Monitor post-remediation for C2 beaconing behavior (periodic outbound connections to uncommon external IPs) for a minimum of 14 days. Confirm no scheduled tasks, registry run keys, or startup items (MITRE T1547) were created during the infection window using D3-SICA (System Init Config Analysis).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-12 (Audit Record Generation), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Without EDR for beaconing detection: configure Windows Firewall with Advanced Security to log all outbound allowed connections (netsh advfirewall set allprofiles logging droppedconnections enable, allowedconnections enable) and parse C:\Windows\System32\LogFiles\Firewall\pfirewall.log daily with a PowerShell script filtering on destination IPs not in an approved allowlist. Use Autoruns (Sysinternals, free) on each recovered endpoint to enumerate and verify all ASEP (Auto-Start Extensibility Points) locations — registry run keys, scheduled tasks, startup folders, service entries — against a clean baseline, which directly addresses the Lorem Ipsum campaign's likely persistence mechanisms. For C2 beacon detection on a budget, run Wireshark or Zeek on a network tap during the 14-day watch period, filtering for JA3/JA3S fingerprints or periodic beacon intervals to uncommon ASNs.

Evidence: BEFORE isolating the endpoint and BEFORE imaging: (1) capture live network state via 'Get-NetTCPConnection -State Established | Select-Object LocalAddress, LocalPort, RemoteAddress, RemotePort, OwningProcess' to document any active C2 connections the Lorem Ipsum payload may have established; (2) run 'netstat -ano' and correlate PIDs to process names via 'tasklist /svc' to identify any injected or spawned processes maintaining outbound connectivity; (3) collect the full contents of %APPDATA%, %TEMP%, and %LOCALAPPDATA% directory listings with timestamps ('Get-ChildItem -Recurse -Force | Select-Object FullName, CreationTime, LastWriteTime') to identify dropper artifacts deposited by the ClickFix-initiated command sequence before the image is taken and the live filesystem state is altered.

Step 5: Post-Incident — This campaign exposes a control gap in user execution policy enforcement. Conduct a lessons-learned review against NIST AC-6 (Least Privilege) and AC-3 (Access Enforcement) to determine whether standard users retain unnecessary execution rights. Evaluate security awareness training currency — ClickFix succeeds because users comply with on-screen instructions without verification. Review CIS 6.3 and 6.4 MFA posture for externally-exposed applications and remote access, as Vice Society historically follows initial access with credential theft.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.4 (Require MFA for Remote Network Access), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: For organizations without a security awareness platform: conduct a targeted tabletop exercise simulating a ClickFix lure scenario — show participants a replica of the fake CAPTCHA or browser-error page used in Lorem Ipsum delivery and walk through why the Run dialog instruction is an attack vector, not a legitimate prompt. Document the lessons-learned findings in a one-page control gap memo mapping identified deficiencies to AC-6 and CIS 5.4, and use it to prioritize the GPO hardening from Step 3. For MFA posture assessment without commercial tooling, audit Azure AD / on-prem AD conditional access policies manually and use the free 'MFASweep' tool (GitHub)

to identify accounts lacking MFA on externally-exposed services — relevant because Vice Society credential theft post-access makes unprotected external portals the lateral-movement and re-entry risk.

Evidence: Post-incident evidence collection focuses on campaign intelligence and control gap documentation rather than volatile host state: (1) export the complete timeline of PowerShell Script Block logs (Event ID 4104) and process creation logs (Event ID 4688) from affected endpoints to reconstruct the full ClickFix execution chain — what command the user ran, what payload was fetched, and what persistence was attempted; (2) collect web proxy logs covering the 30-day window prior to detection to identify other endpoints that visited the same compromised WordPress delivery domains, expanding the potential scope beyond the initially identified host; (3) document the specific WordPress URLs and any dropped file hashes (from %TEMP% or %APPDATA% artifact collection in Step 4) and submit to internal threat intelligence or share via ISAC if your sector (education, healthcare — Vice Society's known targeting verticals) has a relevant sharing community.

Detection Guidance

Primary detection focus is process lineage: look for PowerShell.exe or cmd.exe spawned directly by explorer.exe or RunDLL32 without an administrative context. Windows Security Event ID 4688 (process creation, with command-line logging enabled) and Sysmon Event ID 1 are the primary log sources. Alert specifically on: (1) -EncodedCommand or base64-encoded arguments in PowerShell invocations; (2) mshta.exe, wscript.exe, or cscript.exe launched from user desktop sessions; (3) outbound HTTP/HTTPS connections initiated within 60 seconds of a new PowerShell process from a non-admin user account. Web proxy or DNS logs showing resolution of newly registered or low-reputation domains immediately following WordPress site visits should be treated as a behavioral IOC. No confirmed IOC hashes, IPs, or domains were available in source material at time of writing; monitor threat intelligence feeds and Dark Reading for updated indicators as the campaign is analyzed further. Per NIST AU-6, audit records should be reviewed at defined frequency for these behavioral patterns. This advisory emphasizes behavioral and process-lineage detection; specific indicators will be added as they become available.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`No confirmed IOCs available in source material`	No hashes, IPs, domains, or URLs were disclosed in the referenced Dark Reading coverage at time of writing. Monitor threat intelligence feeds for updated indicators as campaign analysis matures.	LOW

Framework Mappings

MITRE-ATTACK

- **T1566** — Phishing
- **T1059.001** — PowerShell
- **T1041** — Exfiltration Over C2 Channel
- **T1486** — Data Encrypted for Impact

- **T1204.002** — Malicious File
- **T1584.004** — Server
- **T1189** — Drive-by Compromise
- **T1027** — Obfuscated Files or Information

NIST-800-53R5

- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **CM-7** — Least Functionality
- **SI-7** — Software, Firmware, and Information Integrity
- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution
- **SI-10** — Information Input Validation
- **IR-4** — Incident Handling

OWASP-TOP10-2021

- **A03:2021** — Injection

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

ISO-27001-2022

- **A.8.26** — Application security requirements
- **A.5.29** — Information security during disruption

NIST-CSF-2

- **RS.MI-01** — Incidents are contained
- **DE.CM-01** — Networks and network services are monitored

HIPAA-SECURITY

- **164.308(a)(7)(ii)(A)** — Data Backup Plan

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1566	Phishing	Initial-Access

Technique ID	Technique Name	Tactic
T1059.001	PowerShell	Execution
T1041	Exfiltration Over C2 Channel	Exfiltration
T1486	Data Encrypted for Impact	Impact
T1204.002	Malicious File	Execution
T1584.004	Server	Resource-Development
T1189	Drive-by Compromise	Initial-Access
T1027	Obfuscated Files or Information	Defense-Evasion

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/cyberattacks-data-breaches/lorem-ipsum-...	T3
State of WordPress Security in 2026 - Patchstack	https://patchstack.com/whitepaper/state-of-wordpress-security-in-2026/	T3
Apart from brute force attacks, how do wordpress sites get hacked?	https://www.reddit.com/r/ProWordPress/comments/1c8ob3d/apart_from_b.. .	T3
A major WordPress security breach has put thousands of websites at ...	https://www.instagram.com/p/DXKXr-lElhS/	T3
Why do hackers target WordPress sites so frequently? - Quora	https://www.quora.com/Why-do-hackers-target-WordPress-sites-so-freq...	T3

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-06-17 07:12 UTC by TJS Security Command Center