INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:35 UTC

MacSync and the ClickFix Ecosystem: How Three Campaigns in Four Months Reveal a Maturing macOS Threat Pipeline

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0028
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	macOS (all versions targeted), Windows, Google Chrome, WordPress sites, ChatGPT/OpenAI platforms (impersonated), Claude Code/Anthropic (impersonated), GitHub (impersonated), Cloudflare Pages (infrastructure), Squarespace (infrastructure), Tencent EdgeOne (infrastructure), Exodus Wallet, Atomic Wallet, Ledger Wallet, Ledger Live
Published	2026-03-16

Executive Summary

Three coordinated ClickFix social-engineering campaigns between November 2025 and February 2026 delivered the MacSync infostealer to macOS users by impersonating AI tools including ChatGPT and Anthropic Claude Code. Targeted users were tricked into manually executing malicious terminal commands, bypassing automated defenses and enabling credential theft and cryptocurrency wallet drainage across Exodus, Atomic, Ledger, and Ledger Live. Organizations with macOS fleets, developer teams using AI tooling, and employees holding cryptocurrency assets face elevated risk from an evolving, shared delivery infrastructure that resists domain-based blocking.

Technical Analysis

MacSync is a macOS infostealer distributed via ClickFix lures, fake error dialogs and CAPTCHA prompts that instruct users to manually run malicious commands in Terminal. Three campaign waves (November 2025, December 2025, February 2026) show iterative development; the February iteration added dynamic AppleScript payload generation and in-memory execution to reduce forensic footprint. No CVE is assigned. Relevant CWEs: CWE-77 (Command Injection via user-executed terminal commands), CWE-116 (Insufficient Output Encoding), CWE-311 (Missing Encryption for Sensitive Data), CWE-693 (Protection Mechanism Failure), CWE-1021 (Improper Restriction of Rendered UI Layers). MITRE ATT&CK coverage includes T1204.002 (Malicious File execution via user action), T1059.002 (AppleScript), T1059.001 (PowerShell on Windows variants), T1555/T1555.001 (Credential/Keychain Access), T1539 (Steal Web Session Cookie), T1056.001 (Keylogging),

T1055 (Process Injection), T1027/T1027.010 (Obfuscation/Command Obfuscation), T1036/T1036.005 (Masquerading), T1140 (Deobfuscate/Decode), T1071.001 (Web Protocol C2), T1102 (Web Service), T1189 (Drive-by Compromise), T1566/T1566.002 (Phishing), T1583.006 (Acquire Infrastructure: Web Services), T1608.001 (Stage Capabilities), T1176 (Browser Extensions), T1552.001 (Credentials in Files). Payload hosting leverages Cloudflare Pages, Squarespace, and Tencent EdgeOne, complicating domain-based blocking. MacSync shares delivery infrastructure with Alien, Atomic Stealer, StealC, Remcos RAT, CastleRAT, and ModeloRAT. According to multiple security vendor reports (referenced in available T3 intelligence aggregation), at least 20 related campaigns targeting AI and developer tooling have been documented across a six-week window; primary sources include reports from Sophos, Jamf, and Guardio Labs. No patch is applicable, the attack vector is user behavior, not a software vulnerability. Source quality score for this item is 0.54 (T3 sources); verify technical details against primary vendor reports from Sophos, Jamf, and Guardio Labs before acting on specific IOCs.

Action Checklist

1. Step 1, Immediate: Alert macOS users and developer teams to the ClickFix lure pattern; instruct them never to copy-paste Terminal or PowerShell commands from browser dialogs, CAPTCHA prompts, or AI tool error pages.
2. Step 2, Detection: Search endpoint logs and EDR telemetry for unexpected AppleScript execution (osascript), Terminal spawned from browser processes, and PowerShell invocations on Windows hosts; flag in-memory payload execution with no associated file drop.
3. Step 3, Assessment: Inventory macOS endpoints running AI developer tooling (Claude Code, ChatGPT desktop, GitHub CLI); prioritize review for users with access to cryptocurrency wallets (Exodus, Atomic, Ledger Live) or high-value credential stores.
4. Step 4, Detection: Add detection rules for ClickFix-pattern clipboard injection and AppleScript payload generation; configure alerts (not blanket blocks) on outbound connections to Cloudflare Pages, Squarespace, and Tencent EdgeOne domains, prioritizing newly registered or unrecognized domains not matching approved business infrastructure.
5. Step 5, Communication: Notify affected users and, where cryptocurrency asset access is confirmed, recommend immediate wallet rotation and credential reset; brief leadership on business risk to crypto holdings and developer credential exposure.
6. Step 6, Long-term: Review macOS endpoint security policy to restrict or monitor AppleScript and Terminal execution by non-administrative users; evaluate browser security controls to reduce drive-by exposure (T1189); incorporate ClickFix social-engineering scenarios into security awareness training.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to executive leadership and external IR firm if any confirmed cryptocurrency wallet drainage is detected, or if investigation reveals compromise of developer API credentials (GitHub, AWS, npm tokens) used in production systems.

Recovery Notes	Post-eradication: (1) Require full wallet seed phrase rotation for any user with Exodus/Atomic/Ledger access; validate new wallet addresses via blockchain explorer before funds transfer. (2) Revoke all developer API credentials (GitHub PATs, AWS keys, npm tokens) for affected users and require re-authentication to all production systems. (3) Run 30-day forensic monitoring on affected endpoints for osascript, process spawning, and outbound C2 connections; retire endpoints showing persistent indicators. (4) Conduct post-incident review with dev and security teams within 2 weeks to incorporate ClickFix TTPs into threat model.
Forensic Artifacts	macOS: /var/log/system.log, /var/log/unified.log (process execution, osascript events) | macOS: ~/Library/Safari/History.db, ~/Library/Caches/Google/Chrome/Default/Cache (browser history, ClickFix lure domain visits) | macOS: ~/.bash_history, ~/.zsh_history (terminal command history for executed payloads) | Windows: Security Event Log 4688/4689 (process creation, parent-child relationships) | All platforms: DNS query logs, firewall egress logs (outbound connections to Cloudflare Pages, Squarespace, Tencent EdgeOne infrastructure) | macOS: ~/Library/LaunchAgents, ~/Library/LaunchDaemons (persistence mechanisms, scheduled tasks) | Cryptocurrency wallets: blockchain transaction history for wallet addresses, wallet.db modification timestamps (Exodus, Atomic, Ledger)

Per-Action IR Details

Step 1, Immediate: Alert macOS users and developer teams to the ClickFix lure pattern; instruct them never to copy-paste Terminal or PowerShell commands from browser dialogs, CAPTCHA prompts, or AI tool error pages.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (Preparation phase: awareness and training)

Controls: NIST 800-53 AT-2 (Security Awareness and Training), NIST 800-53 AT-3 (Role-Based Security Training), CIS 6.5 (Security Awareness Program)

Compensating: Send alert email with screenshot examples of ClickFix dialogs impersonating ChatGPT/Claude; include rule: never execute pasted Terminal commands without manual inspection of each line. Post visual checklist on internal wiki (Confluence/Notion) with 'AI tool will never ask for terminal commands' banner. Include this in daily standup for dev teams.

Evidence: Capture user awareness acknowledgment logs (email read receipts, training platform completion records) to demonstrate timeliness of warning. Document timestamp of alert distribution for incident timeline.

Step 2, Detection: Search endpoint logs and EDR telemetry for unexpected AppleScript execution (osascript), Terminal spawned from browser processes, and PowerShell invocations on Windows hosts; flag in-memory payload execution with no associated file drop.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 (Detection and Analysis phase: indicators and log review)

Controls: NIST 800-53 SI-4 (Information System Monitoring), NIST 800-53 AU-12 (Audit Generation), CIS 8.5 (Log Monitoring and Alerting)

Compensating: On macOS without EDR: run 'log show --predicate "processImagePath contains[cd] osascript OR processImagePath contains[cd] /usr/bin/ruby" --style syslog' for past 30 days; export to CSV. For Windows: query Event Log 4688 (Process Creation) for powershell.exe spawned from chrome.exe or firefox.exe parent; use wevtutil export to file. Check ~/.bash_history and PowerShell history ($PROFILE logs) for suspicious curl/wget downloads to /tmp or %TEMP%.

Evidence: Preserve /var/log/system.log and /var/log/unified.log (macOS) with focus on osascript and process spawning events. On Windows: export Security Event Log (4688, 4689 process creation/termination), Application log, and PowerShell Operational log. Capture process command-line arguments in full (not truncated). Snapshot browser cache and download history before analysis.

Step 3, Assessment: Inventory macOS endpoints running AI developer tooling (Claude Code, ChatGPT desktop, GitHub CLI); prioritize review for users with access to cryptocurrency wallets (Exodus, Atomic, Ledger Live) or high-value credential stores.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2.5 (Prioritization of response effort)

Controls: NIST 800-53 RA-3 (Risk Assessment), NIST 800-53 CA-7 (Continuous Monitoring), CIS 2.1 (Asset Inventory and Management)

Compensating: Run 'system_profiler SPApplicationsDataType | grep -i "claude\|chatgpt\|github" > ai_tools_inventory.txt' on each macOS endpoint. Query LDAP or local password manager records to identify users with Exodus/Atomic/Ledger account access; cross-reference with endpoint list. For credential exposure: check ~/.ssh/config, ~/.aws/credentials existence and modification times (stat -f %Sm ~/.aws/credentials). Manually review users in sudo group: dscacheutil -q group -a name sudo.

Evidence: Capture installed application inventory (Applications folder timestamps, LaunchAgents plist modification times in ~/Library/LaunchAgents). Document user group memberships and sudo eligibility. Preserve cryptocurrency wallet application logs if present (~/.exodus, ~/.atomic folders with wallet.db timestamps). Screenshot credential store locations before access.

Step 4, Detection: Add detection rules for ClickFix-pattern clipboard injection and AppleScript payload generation; configure alerts (not blanket blocks) on outbound connections to Cloudflare Pages, Squarespace, and Tencent EdgeOne domains, prioritizing newly registered or unrecognized domains not matching approved business infrastructure.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.4.1 (Detection rules and signatures)

Controls: NIST 800-53 SI-4(a) (System monitoring with tools), NIST 800-53 CA-7(a) (Continuous monitoring program), CIS 8.6 (Endpoint Detection and Response)

Compensating: macOS without EDR: Use Little Snitch (free tier) to log all outbound connections; export daily to CSV and grep for pages.dev, squarespace-cdn, or *.tencent-cloud domains. For AppleScript detection: monitor /var/log/system.log for 'osascript' with stderr redirects or eval patterns; set up cron job 'log stream --predicate "processImagePath contains osascript" --level debug' for real-time capture. Windows: enable PowerShell Script Block Logging (Group Policy: Computer Configuration > Administrative Templates > Windows Components > Windows PowerShell > Turn on PowerShell Script Block Logging) and monitor for IEX (Invoke-Expression) with URL patterns. Parse firewall logs: grep for connection attempts to known ClickFix infrastructure (maintain updated domain list from threat intel feeds).

Evidence: Capture DNS query logs (macOS: /var/log/system.log DNS lookups; Windows: DNS client event log 3008-3016) for 7 days pre-alert. Export firewall connection logs showing destination IPs, domains, and timestamps. Preserve clipboard content captures if available (macOS pbpaste logs are not standard; use manual snapshot during investigation). Screenshot osascript process trees with full command-line arguments.

Step 5, Communication: Notify affected users and, where cryptocurrency asset access is confirmed, recommend immediate wallet rotation and credential reset; brief leadership on business risk to crypto holdings and developer credential exposure.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 (Containment phase: stakeholder notification) and §2.3.9 (Communication plan)

Controls: NIST 800-53 IR-4 (Incident Handling), NIST 800-53 IR-6 (Incident Reporting), CIS 6.2 (Incident Management Process)

Compensating: Document affected user list with risk tier (crypto access: critical; dev credentials: high; general user: medium). Send tiered notifications: (1) critical-tier users: phone call + email with wallet rotation procedure (new seed phrase generation, funds transfer to new wallet); (2) high-tier users: email with credential reset instructions (password manager reset, API token revocation); (3) management briefing: one-page risk summary including estimated exposure

(# users, # wallets, credential types affected). Include timeline: when exposure was possible, when detection occurred, when remediation began.

Evidence: Preserve all communication logs (email delivery receipts, call records) for compliance audit. Document user acknowledgment of notifications. Capture pre- and post-rotation wallet transaction history (export from blockchain explorers for affected wallet addresses to detect theft). Snapshot credential reset audit logs (password manager, GitHub PAT revocation events, AWS API key rotation timestamps).

Step 6, Long-term: Review macOS endpoint security policy to restrict or monitor AppleScript and Terminal execution by non-administrative users; evaluate browser security controls to reduce drive-by exposure (T1189); incorporate ClickFix social-engineering scenarios into security awareness training.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §3.4 (Post-Incident Activities: lessons learned) and NIST 800-53 AC-3 (Access Control Policy)

Controls: NIST 800-53 AC-2 (Account Management), NIST 800-53 AC-3 (Access Control), NIST 800-53 AT-2 (Security Awareness Training), CIS 5.2 (User and Administrative Account Management)

Compensating: macOS without MDM: Deploy LaunchAgent script to log all osascript and Terminal launches to syslog (use launchd plist in /Library/LaunchDaemons); alert on non-admin user execution. Remove Terminal.app and PowerShell from non-admin user launchd PATH (edit /etc/shells, restrict Execute permission on /usr/bin/security, /usr/bin/osascript to admin group). Browser hardening: disable JavaScript execution in plugin contexts (disable Flash, Java plugins); enable browser sandbox restrictions (Safari: Develop > Disable JavaScripting; Chrome policy: Disable 3rd-party cookies, enforce HTTPS-only). Create ClickFix scenario for phishing simulations: mock ChatGPT error page requesting copy-paste command, track click and paste rates monthly, brief teams on results.

Evidence: Document policy change log (date effective, approval chain). Preserve before/after screenshots of access control settings. Capture training completion metrics (attendance, quiz scores). Archive lessons-learned meeting notes with action items and owners. Establish baseline: measure AppleScript execution by non-admin users weekly for 60 days post-remediation to confirm reduction.

Detection Guidance

Behavioral indicators: osascript (AppleScript) execution launched from a browser process (Safari, Chrome, Firefox) or spawned unusually from a non-developer context; Terminal.app opening immediately after browser interaction; clipboard content containing base64-encoded strings or curl/bash one-liners followed by user-initiated Terminal paste. EDR queries: look for process trees where browser PID is parent or grandparent of osascript, sh, bash, or curl. On Windows: PowerShell launched via browser process with encoded commands (-EncodedCommand flag or base64 argument). Log sources: macOS Unified Log (process launch events), EDR process telemetry, endpoint DLP for keychain or wallet file access. Network indicators: configure alerts on outbound HTTP/S to Cloudflare Pages domains (*.pages.dev) and Tencent EdgeOne CDN endpoints not matching approved asset inventory; note that Cloudflare Pages hosts legitimate content, so alert on new or unrecognized domains, not the CDN wholesale. File system: watch for new unsigned executables written to user home directories (~/.local, ~/Library, /tmp) following browser sessions. Wallet targeting: monitor for file reads against known wallet paths (~/Library/Application Support/Exodus, Atomic, Ledger Live). Note: specific IOC values (domains, hashes, IPs) are not confirmed in the available T3 sources for this item; pull current IOC lists from Sophos Threat Intelligence, Jamf Threat Labs, and Guardio Labs reports before deploying signature-based blocks.

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`Tencent EdgeOne CDN endpoints (specific domains not confirmed in available sources)`	Identified as payload hosting and redirection infrastructure in campaign description. Specific domains not available from T3 source material — obtain from Sophos, Jamf, or Guardio Labs reporting.	LOW
DOMAIN	`Squarespace-hosted domains (specific values not confirmed in available sources)`	Used for payload staging and redirection. Specific domains not extractable from available T3 sources — validate against primary vendor IOC feeds.	LOW
URL	`Not confirmed — source quality insufficient for specific URL IOCs`	No verified payload URLs available from T3 sources. Do not deploy URL-based blocks without confirmation from primary vendor reports.	LOW
URL	`Source-specific IOCs not extractable from available T3 sources`	Confirmed campaign IOCs (domains, IPs, hashes) were not present in the provided source data. Pull current indicators from MITRE ATT&CK, your threat intelligence platform, and vendor feeds referencing MacSync and ClickFix AI-lure campaigns active November 2025 to February 2026.	LOW

Framework Mappings

MITRE-ATTACK

- **T1552.001** — Credentials In Files
- **T1102** — Web Service
- **T1189** — Drive-by Compromise
- **T1027.010** — Command Obfuscation
- **T1056.001** — Keylogging
- **T1055** — Process Injection
- **T1555.001** — Keychain
- **T1539** — Steal Web Session Cookie
- **T1583.006** — Web Services
- **T1059.001** — PowerShell
- **T1204.002** — Malicious File
- **T1027** — Obfuscated Files or Information
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1140** — Deobfuscate/Decode Files or Information
- **T1071.001** — Web Protocols

- **T1036** — Masquerading
- **T1608.001** — Upload Malware
- **T1566** — Phishing
- **T1059.002** — AppleScript
- **T1176** — Software Extensions
- **T1566.002** — Spearphishing Link
- **T1555** — Credentials from Password Stores

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SI-8** — Spam Protection
- **SI-10** — Information Input Validation
- **SC-13** — Cryptographic Protection

OWASP-TOP10-2021

- **A03:2021** — Injection

CIS-V8

- **16.10**
- **6.3** — Require MFA for Externally-Exposed Applications
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.312(e)(1)** — Transmission Security

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

ISO-27001-2022

- **A.8.24** — Use of cryptography
- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552.001	Credentials In Files	Credential-Access
T1102	Web Service	Command-And-Control
T1189	Drive-by Compromise	Initial-Access
T1027.010	Command Obfuscation	Defense-Evasion
T1056.001	Keylogging	Collection
T1055	Process Injection	Defense-Evasion
T1555.001	Keychain	Credential-Access
T1539	Steal Web Session Cookie	Credential-Access
T1583.006	Web Services	Resource-Development
T1059.001	PowerShell	Execution
T1204.002	Malicious File	Execution
T1027	Obfuscated Files or Information	Defense-Evasion
T1036.005	Match Legitimate Resource Name or Location	Defense-Evasion
T1140	Deobfuscate/Decode Files or Information	Defense-Evasion
T1071.001	Web Protocols	Command-And-Control
T1036	Masquerading	Defense-Evasion
T1608.001	Upload Malware	Resource-Development
T1566	Phishing	Initial-Access
T1059.002	AppleScript	Execution
T1176	Software Extensions	Persistence
T1566.002	Spearphishing Link	Initial-Access
T1555	Credentials from Password Stores	Credential-Access

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/03/clickfix-campaigns-spread-macsync...	T3

Source	URL	Tier
Fake GitHub tools are wiping wallets of Windows users■ | Cybernews	https://cybernews.com/security/github-malware-microsoft-chrome-pass...	T3
AI Fingerprinting: Claude, ChatGPT, Gemini Exposed - LinkedIn	https://www.linkedin.com/posts/luke-harby_k3ym-k3ym0infosecexchange...	T3
Is any of you getting the error "Please unblock challenges.cloudflare ...	https://www.reddit.com/r/CloudFlare/comments/1axpzfs/is_any_of_you_...	T3
I Got A Remote Code Execution On A Wordpress Site Using AI	https://www.youtube.com/watch?v=AnVONITvWw4	T3
Please unblock challenges.cloudflare.com to proceed.	https://www.reddit.com/r/Anthropic/comments/1p0aamp/please_unblock_...	T3
10K Claude Desktop Users Exposed by Zero-Click ...	https://www.esecurityplanet.com/threats/10k-claude-desktop-users-ex...	T3
ChatGPT Vulnerability - Security Flaws within ...	https://salt.security/blog/security-flaws-within-chatgpt-extensions...	T3

Generated 2026-03-29 18:35 UTC by TJS Security Command Center