INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 18:51 UTC

# Reputation Poisoning at Scale: Rust Clipboard Hijacker Weaponizes Platform Trust Signals Across Six Channels

**THREAT CAMPAIGN** | **HIGH** | CVSS 7.5

| | |
|---|---|
| **SCC Item ID** | SCC-CAM-2026-0501 |
| **Type** | Threat Campaign |
| **Severity** | HIGH |
| **CVSS Base Score** | 7.5 |
| **Affected Products** | Windows, macOS, targeting cryptocurrency holders and online gamblers via trojanized software distributed through VirusTotal, GitHub, SourceForge, YouTube, WordPress, and EIN Presswire syndication |
| **Published** | 2026-06-17T14:14:24 |
| **Discovery Source** | Rss |

## Executive Summary

An unattributed threat actor, assessed with low-to-moderate confidence based on secondary reporting, is running a coordinated campaign that distributes Rust-compiled clipboard-hijacking malware across Windows and macOS systems, targeting cryptocurrency holders and online gamblers. The malware silently replaces copied wallet addresses at paste time, redirecting crypto transactions to attacker-controlled wallets. Analysis suggests the same delivery infrastructure could be repurposed with minimal effort to deploy infostealers or ransomware against enterprise targets.

## Technical Analysis

The campaign delivers a Rust-compiled clipper (clipboard hijacker) that monitors clipboard content for cryptocurrency wallet address patterns and substitutes attacker-controlled addresses at paste time. Affected platforms are Windows and macOS. No CVE applies; this is a malware campaign. CVSS 7.5 is an estimated impact score for the malware payload and is not derived from an official CVE entry. Relevant CWEs: CWE-345 (Insufficient Verification of Data Authenticity), CWE-494 (Download of Code Without Integrity Check), CWE-693 (Protection Mechanism Failure). Distribution exploits manufactured reputation signals across six channels: inflated VirusTotal community vote counts, artificial GitHub star counts, boosted SourceForge download metrics, inflated YouTube subscriber numbers, WordPress-hosted phishing pages, and EIN Presswire press release syndication through USA TODAY Network partner syndication. Tools are masqueraded as legitimate

cryptocurrency sniper bots and gambling prediction tools (T1036). MITRE ATT&CK techniques include T1204.002 (User Execution: Malicious File), T1510 (Clipboard Data), T1553 (Subvert Trust Controls), T1608.001 (Stage Capabilities: Upload Malware), T1583.001 (Acquire Infrastructure: Domains), T1566 (Phishing), T1027.002 (Obfuscated Files or Code), T1036 (Masquerading), T1586.003 (Compromise Accounts: Cloud Accounts), and T1588.001 (Obtain Capabilities: Malware). No patch exists; the threat is campaign-based and mitigated through user behavior controls and endpoint monitoring.

## Action Checklist

1. Step 1: Containment. Block downloads originating from untrusted SourceForge projects, unsolicited GitHub repositories with abnormally high star-to-contributor ratios, and press release domains (einpresswire.com) at the web proxy or DNS layer. Flag WordPress-hosted software distribution pages for review. Enforce CIS Controls to restrict outbound connections from newly installed unsigned binaries.
2. Step 2: Detection. Query EDR telemetry for Rust-compiled binaries (PE header: compiler artifact 'rustc') installed outside of managed software channels. Monitor clipboard access events on Windows and macOS (accessibility API abuse patterns). Hunt for processes polling clipboard content at high frequency. Confirm audit logging coverage includes clipboard and process execution events. Apply behavioral hunting pivots for local account monitoring and system file analysis.
3. Step 3: Eradication. Remove any binaries identified as the Rust clipper payload. Revoke and rotate all cryptocurrency wallet credentials and generate new receiving addresses for any wallets accessed on potentially compromised hosts. Remove flagged applications from affected endpoints. Add campaign IOCs to blocklists across endpoint and network controls. Implement credential rotation for any accounts used on affected hosts.
4. Step 4: Recovery. Verify clipboard monitoring processes are no longer present via EDR. Confirm cryptocurrency transactions during the exposure window routed to expected wallet addresses; flag and escalate any anomalous transaction destinations. Re-image hosts where the payload was confirmed present. Monitor reinstated hosts for recurrence of clipboard access events. Validate that software allow-listing policies are enforced on affected endpoints.
5. Step 5: Post-Incident. Review and formalize a process for evaluating third-party software trust signals before installation, recognizing that VirusTotal vote counts, GitHub stars, and download metrics are not integrity guarantees. Implement multi-factor authentication on all accounts with access to cryptocurrency assets or financial systems. Restrict standard user installation privileges to prevent bypassing approved software channels. Document control improvements for external system vetting requirements.

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | URGENT |
| **Escalation Criteria** | Escalate immediately to senior IR leadership and legal counsel if blockchain transaction review confirms any cryptocurrency funds were redirected to attacker-controlled wallets during the exposure window, as this constitutes confirmed financial loss and may trigger financial regulatory notification obligations or cyber insurance claim procedures; also escalate if EDR or Sysmon telemetry reveals the Rust clipper payload has been replaced with or is co-resident with an infostealer or ransomware component, consistent with Check Point Research's warning that the same delivery infrastructure can be repurposed with minimal effort. |

| | |
|---|---|
| **Recovery Notes** | After re-imaging confirmed-compromised hosts, generate entirely new cryptocurrency receiving addresses from a clean, air-gapped or verified-clean device, and do not reuse any address that was copied to clipboard on a host during the exposure window, as attacker infrastructure may have logged harvested addresses for future targeting. Monitor reinstated hosts for a minimum of 30 days using Sysmon clipboard access event rules and unsigned binary execution alerts, given that this campaign's multi-platform distribution (SourceForge, GitHub, YouTube, WordPress, EIN Presswire) suggests the threat actor has broad reach and may attempt reinfection through a different delivery channel. Validate software allowlisting enforcement weekly for the first month by running an osquery or PowerShell inventory sweep comparing installed executables against the approved software baseline, specifically checking for Rust-compiled PE signatures in user-writable directories. |
| **Forensic Artifacts** | Rust-compiled PE binary on disk in user-writable paths (%APPDATA%, %TEMP%, %USERPROFILE%\Downloads on Windows; ~/Library/Application Support or /tmp on macOS) — identifiable by PE section layout and compiler metadata strings containing 'rustc' version markers in the .rdata section, extractable with strings or pe-sieve before quarantine \| Windows Security Event Log Event ID 4663 (Object Access — Clipboard) and Sysmon Event ID 10 (ProcessAccess) records showing the clipper process handle-opening clipboard objects belonging to browser, wallet application, or exchange client processes at polling intervals under 2 seconds \| macOS Unified Log entries from the AXRuntime category (accessibility API) showing an unsigned or ad-hoc-signed process invoking NSPasteboard readStringForType: at anomalous frequency, which is the macOS mechanism this clipboard hijacker class uses in the absence of a native clipboard event API \| Windows Registry Run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run) or macOS LaunchAgent plist files in ~/Library/LaunchAgents containing an entry pointing to the Rust clipper binary path, establishing the persistence mechanism and installation timestamp \| Blockchain ledger records (retrievable via Etherscan, Blockchain.com, or equivalent public explorer) for all wallet addresses that were active — copied, pasted, or transacted — on potentially compromised hosts during the exposure window, which are the primary evidence of financial impact and will show whether destination addresses match attacker-controlled wallets identified in Check Point Research's IOC set |

### Per-Action IR Details

**Step 1: Containment — Block downloads originating from untrusted SourceForge projects, unsolicited GitHub repositories with abnormally high star-to-contributor ratios, and press release domains (einpresswire.com) at the web proxy or DNS layer. Flag WordPress-hosted software distribution pages for review. Enforce CIS 4.4 and CIS 4.5 (host-based firewall rules) to restrict outbound connections from newly installed unsigned binaries.**

**NIST Phase:** Containment

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

**Controls:** CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices), NIST AC-4 (Information Flow Enforcement)

**Compensating:** On Windows hosts, use Windows Defender Firewall with Advanced Security to create outbound block rules scoped to unsigned binaries in user-writable paths (e.g., %APPDATA%, %TEMP%, %USERPROFILE%\Downloads). On macOS, use pf rules or Little Snitch (free tier) to block outbound from non-code-signed processes. At the DNS layer, deploy Pi-hole or RPZ on the organization's recursive resolver and import a blocklist seeded with einpresswire.com and any campaign-associated SourceForge/GitHub subdomains identified from Check Point Research's IOC release. For proxy-based blocking, Squid with an ACL denying the flagged domains requires no commercial licensing.

**Evidence:** Before implementing proxy/DNS blocks that would sever command-and-control or exfiltration channels, capture: (1) full DNS query logs from the affected host's resolver cache ('ipconfig /displaydns' on Windows; 'scutil --dns' and mDNSResponder logs on macOS) to preserve pre-block resolution history; (2) active network connections showing any outbound sessions from the suspected Rust clipper process ('Get-NetTCPConnection -State Established | Where-Object {$_.OwningProcess -ne 0}' on Windows; 'lsof -i -n -P' on macOS); (3) browser download history and proxy access logs timestamped to the initial download event from SourceForge, GitHub, or the WordPress distribution page, which will anchor the initial access timeline.

**Step 2: Detection — Query EDR telemetry for Rust-compiled binaries (PE header: compiler artifact 'rustc') installed outside of managed software channels. Monitor clipboard access events on Windows (Event ID 4663 on objects of type Clipboard, where supported by EDR) and macOS (accessibility API abuse patterns). Hunt for processes polling clipboard content at high frequency. Review AU-2 (Event Logging) coverage to confirm clipboard and process execution events are captured. Reference D3-LAM (Local Account Monitoring) and D3-SFA (System File Analysis) for behavioral hunting pivots.**

**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), CIS 8.2 (Collect Audit Logs)

**Compensating:** Without EDR, deploy Sysmon with a configuration that enables Event ID 10 (ProcessAccess) targeting clipboard-touching calls and Event ID 1 (ProcessCreate) for new process execution. Use the SwiftOnSecurity Sysmon config as a baseline and add a rule matching Image paths in %APPDATA% or %TEMP% with CommandLine not matching any managed installer. On macOS, enable Unified Log collection and query with 'log stream --predicate "category == 'AXRuntime'" ' to surface accessibility API abuse by unsigned applications polling clipboard state. Write a PowerShell hunt script that enumerates running processes, checks their module paths against known software inventory, and flags any PE whose version info strings contain 'rustc' compiler markers ('Get-Process | ForEach-Object { try { [System.Diagnostics.FileVersionInfo]::GetVersionInfo($_.MainModule.FileName) } catch {} }').

**Evidence:** This is a detection/analysis step that does not itself alter live state, but evidence that must be captured before any downstream containment action includes: (1) a live process list with full image paths and parent-child relationships (Sysmon Event ID 1 chain or 'Get-WmiObject Win32_Process | Select-Object ProcessId, ParentProcessId, Name, ExecutablePath, CommandLine'); (2) clipboard API call stack — if EDR supports it, capture the call stack of any process invoking OpenClipboard/GetClipboardData (Windows) or NSPasteboard readStringForType (macOS) at intervals under 2 seconds; (3) the binary on disk at its installed path before any quarantine action, including its PE headers and import table (use 'dumpbin /headers' or 'pe-sieve' to extract compiler artifacts confirming rustc provenance); (4) Prefetch files (C:\Windows\Prefetch*.pf) and Shimcache/Amcache entries confirming first-execution timestamp of the suspicious binary.

**Step 3: Eradication — Remove any binaries identified as the Rust clipper payload. Revoke and rotate all cryptocurrency wallet credentials and generate new receiving addresses for any wallets accessed on potentially compromised hosts. Apply CIS 2.3 (Address Unauthorized Software) to remove flagged applications. Enforce CIS 7.1 (Vulnerability Management Process) to add campaign IOCs to blocklists across endpoint and network controls. Implement D3-CRO (Credential Rotation) for any accounts used on affected hosts.**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication

**Controls:** CIS 2.3 (Address Unauthorized Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), NIST AC-2 (Account Management)

**Compensating:** Use ClamAV with a custom YARA rule targeting the Rust clipper's behavioral signature (e.g., PE sections named '.rdata' containing wallet address regex patterns, combined with import of 'GetClipboardData' and 'SetClipboardData' with no corresponding UI thread) to scan all user-writable directories before removal. For macOS, use 'find / -name "*.app" -exec codesign -dv {} \; 2>&1 | grep -i "not signed"' to enumerate unsigned application bundles for review. Wallet credential rotation must be performed from a known-clean device — not the potentially compromised

host — to prevent the clipper from intercepting the new address during the rotation workflow itself.

**Evidence:** CRITICAL — volatile capture must precede binary removal, credential revocation, and any host action: (1) acquire a full RAM image using WinPmem (Windows) or osxpmem (macOS) before terminating the clipper process, as the in-memory process may hold decrypted attacker wallet address strings not present in the on-disk binary; (2) record all active clipboard contents at time of discovery ('Get-Clipboard' on Windows; 'pbpaste' on macOS) to document whether an attacker wallet address is currently staged for injection; (3) export the binary's file hash (SHA-256), full file path, creation timestamp, and last-modified timestamp before deletion, as these anchor the installation timeline and support IOC sharing; (4) capture Windows Registry run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run, HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run) and LaunchAgent/LaunchDaemon plist paths on macOS to document persistence mechanisms before removal; (5) preserve any scheduled task XML definitions ('schtasks /query /fo LIST /v' on Windows) that may reference the clipper binary path.

**Step 4: Recovery — Verify clipboard monitoring processes are no longer present via EDR. Confirm cryptocurrency transactions during the exposure window routed to expected wallet addresses; flag and escalate any anomalous transaction destinations. Re-image hosts where the payload was confirmed present. Monitor reinstated hosts for recurrence of clipboard access events per AU-6 (Audit Record Review, Analysis, and Reporting). Validate that software allow-listing policies (CIS 2.1, CIS 2.2) are enforced on affected endpoints.**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery

**Controls:** NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

**Compensating:** For teams without EDR on reinstated hosts, deploy Sysmon immediately post-reimage and baseline clipboard-touching process activity within the first 24 hours to establish a clean reference. Use osquery with a scheduled query against the 'processes' table filtering on executables in user-writable paths to detect any recurrence of unsigned Rust binaries. For transaction verification, use public blockchain explorers (Etherscan, Blockchain.com) to audit outbound transfers from wallets that were active on affected hosts during the exposure window, comparing destination addresses against the user's own documented receiving addresses. Microsoft AppLocker (free, built into Windows Enterprise) or Windows Defender Application Control can enforce software allowlisting without additional cost.

**Evidence:** Before re-imaging (which destroys all live state and on-disk artifacts), ensure the following have been preserved: (1) a forensic disk image of the compromised host using a write-blocker and FTK Imager or dd, preserving the full filesystem including deleted file entries that may show clipper installation artifacts; (2) all Windows Event Logs exported to .evtx format, specifically the Security, System, Application, and Microsoft-Windows-Sysmon/Operational channels; (3) browser history and download records from Chrome (AppData\Local\Google\Chrome\User Data\Default\History), Firefox (AppData\Roaming\Mozilla\Firefox\Profiles*.default\places.sqlite), and Safari (~/Library/Safari/History.db) that may reveal the SourceForge, GitHub, or WordPress download origin; (4) a final clipboard state capture and running process list taken immediately before initiating the reimage, as a last-chance volatile collection.

**Step 5: Post-Incident — Review and formalize a process for evaluating third-party software trust signals before installation, recognizing that VirusTotal vote counts, GitHub stars, and download metrics are not integrity guarantees. Implement D3-MFA (Multi-factor Authentication) on all accounts with access to cryptocurrency assets or financial systems. Apply CIS 5.4 (Restrict Administrator Privileges) to prevent standard users from installing software outside approved channels. Map a control improvement to NIST AC-20 (Use of External Systems) to formalize vetting requirements for externally sourced tools.**

**NIST Phase:** Post Incident

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

**Controls:** CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.5 (Require MFA for Administrative Access), NIST AC-20 (Use Of External Systems), NIST AC-6 (Least Privilege)

**Compensating:** Produce a one-page software vetting checklist specific to the trust-signal weaponization observed in this campaign: require that any externally sourced tool pass (1) cryptographic signature verification against the developer's published signing key, not just VirusTotal scan results; (2) a manual review of GitHub repository commit history for artificially compressed timelines (high star counts accrued within days of repo creation); and (3) a sandbox detonation using any.run or Cuckoo Sandbox before deployment to production hosts. Distribute this checklist to all personnel who install software. For MFA on cryptocurrency accounts, recommend hardware security keys (YubiKey, which offers a free tier for personal use) rather than SMS-based OTP, given that clipboard hijackers can intercept soft-token codes if the authenticator app is on the same compromised device.

**Evidence:** No live-state evidence is at risk in the post-incident phase, but the following artifacts from earlier phases must be consolidated for the lessons-learned record: (1) the complete timeline of download events sourced from browser history, proxy logs, and DNS query logs, documenting which platform (VirusTotal, SourceForge, GitHub, YouTube description link, WordPress page, or EIN Presswire syndication) served as the initial delivery vector for this specific compromise; (2) the SHA-256 hash and VirusTotal detection ratio at time of download versus at time of discovery, quantifying the detection gap that allowed the binary to appear legitimate; (3) blockchain transaction records covering the exposure window for all wallets active on confirmed-compromised hosts, which establish the financial impact scope and may support any required regulatory or insurance notification.

## Detection Guidance

Primary behavioral indicator: a process repeatedly reading and writing clipboard content at short intervals, particularly one spawned from a user-downloaded binary not present in the organization's software inventory. On Windows, look for clipboard access events tied to unsigned or recently installed executables; EDR tools that surface API-level clipboard hooks (SetClipboardData/GetClipboardData patterns) will be most effective. On macOS, monitor for accessibility API permission grants to unknown applications. Hunt for Rust-compiled PE binaries (presence of 'rustc' compiler strings in binary metadata) in user-writable directories (Downloads, AppData, temp paths). Network indicators: outbound connections from clipboard-monitoring processes to unknown external IPs shortly after a paste event may indicate exfiltration or C2 check-in. Block and alert on downloads sourced from einpresswire.com-linked URLs and unvetted SourceForge or GitHub repositories distributing executable content. No confirmed hash-based IOCs or command-and-control domains were present in the available source data; treat all binaries from the identified distribution channels as suspect pending additional threat intelligence. Behavioral hunting via system file analysis and local account monitoring are recommended detection pivots.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| DOMAIN | `einpresswire.com (distribution vector)` | EIN Presswire press release syndication used to lend mainstream credibility to malware distribution campaigns; treat software download links originating from this network with elevated suspicion | MEDIUM |
| URL | `VirusTotal community vote manipulation (platform-level signal)` | Inflated VirusTotal community votes used as false legitimacy signal; not a discrete URL IOC but a detection-relevant pattern indicating vote-farming behavior on submitted samples | MEDIUM |

## Framework Mappings

### MITRE-ATTACK

- **T1588.001** — Malware
- **T1417.001** — Keylogging
- **T1583.001** — Domains
- **T1608** — Stage Capabilities
- **T1036** — Masquerading
- **T1566** — Phishing
- **T1027** — Obfuscated Files or Information
- **T1027.002** — Software Packing
- **T1204.002** — Malicious File
- **T1608.001** — Upload Malware
- **T1553** — Subvert Trust Controls
- **T1510**
- **T1496** — Resource Hijacking
- **T1608.006** — SEO Poisoning
- **T1608.003** — Install Digital Certificate
- **T1586.003** — Cloud Accounts

### 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
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-3** — Configuration Change Control
- **CP-9** — System Backup
- **IR-4** — Incident Handling

### OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

### CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **8.2** — Collect Audit Logs

### 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

### ISO-27001-2022

- **A.5.29** — Information security during disruption
- **A.8.8** — Management of technical vulnerabilities

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1588.001 | Malware | Resource-Development |
| T1417.001 | Keylogging | Collection |
| T1583.001 | Domains | Resource-Development |
| T1608 | Stage Capabilities | Resource-Development |
| T1036 | Masquerading | Defense-Evasion |
| T1566 | Phishing | Initial-Access |
| T1027 | Obfuscated Files or Information | Defense-Evasion |
| T1027.002 | Software Packing | Defense-Evasion |
| T1204.002 | Malicious File | Execution |
| T1608.001 | Upload Malware | Resource-Development |
| T1553 | Subvert Trust Controls | Defense-Evasion |
| T1510 | Clipboard Data (clipboard hijacking payload) | |
| T1496 | Resource Hijacking | Impact |
| T1608.006 | SEO Poisoning | Resource-Development |
| T1608.003 | Install Digital Certificate | Resource-Development |
| T1586.003 | Cloud Accounts | Resource-Development |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Security News** | https://thehackernews.com/2026/06/crypto-clipper-campaign-abuses-fa... | T3 |
| **Known Exploited Vulnerabilities Catalog \| CISA** | https://www.cisa.gov/known-exploited-vulnerabilities-catalog | T1 |
| **Huntress CVE Library: Common Vulnerabilities Database** | https://www.huntress.com/threat-library/vulnerabilities | T3 |
| **Cybersecurity Vulnerabilities: Key Trends and Strategies - YouTube** | https://www.youtube.com/watch?v=8i7Wfv4TNPw | T3 |
| **Software vulnerabilities \| ReversingLabs Glossary** | https://www.reversinglabs.com/glossary/software-vulnerabilities | T3 |



Generated 2026-06-17 18:51 UTC by TJS Security Command Center