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

Phantom Stealer Runs Entirely in Memory to Harvest Browser Credentials While Evading Analysis

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0496
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Browser applications, specific vendors and versions not confirmed in available source excerpt; credential stores for Chromium-based and gecko-based browsers are typical targets of this malware class
Published	2026-06-16T18:26:34
Discovery Source	Rss

Executive Summary

Phantom Stealer is a fileless malware campaign targeting credentials stored in web browsers, executing entirely in memory to avoid detection by traditional endpoint security tools. Organizations whose employees use Chromium-based or Firefox-based browsers for accessing business applications, SaaS platforms, or internal systems face direct risk of credential theft without leaving traces on disk. A successful compromise can expose corporate accounts, enable lateral movement, and lead to broader network intrusion with no file-based forensic evidence to support incident response.

Technical Analysis

Phantom Stealer is a fileless infostealer that loads and executes exclusively in system memory, targeting credential stores in Chromium-based and Gecko-based browsers. No on-disk artifacts are written, bypassing signature-based AV and file-integrity scanning. Mapped CWEs include CWE-312 (cleartext storage of sensitive information), CWE-311 (missing encryption of sensitive data), and CWE-522 (insufficiently protected credentials). MITRE ATT&CK techniques include T1055 (Process Injection), T1027 (Obfuscated Files or Information), T1140 (Deobfuscate/Decode Files or Information), T1497 (Virtualization/Sandbox Evasion), T1059 (Command and Scripting Interpreter), T1555.003 (Credentials from Web Browsers), and T1003.001 (OS Credential Dumping: LSASS Memory). No CVE is associated; this is a malware campaign, not a discrete software vulnerability. No patch exists, mitigation is behavioral and architectural. The primary source is Dark

Reading (T3). In-memory execution and browser credential targeting are confirmed behaviors; specific technical details (API calls, sandbox evasion methods) are inferred from malware class patterns and carry medium confidence pending additional threat intelligence.

Action Checklist

1. Step 1: Containment. Identify endpoints where browser credential stores may have been accessed outside normal user sessions. Prioritize systems with administrative privileges and those with access to SaaS, VPN, or identity provider portals. Isolate any host showing unexpected process injection activity or memory-resident code execution, pending investigation. Apply NIST AC-6 (Least Privilege) to restrict which accounts can access sensitive browser-stored credentials.
2. Step 2: Detection. Query EDR telemetry for process injection indicators (T1055): unusual parent-child process relationships, memory-resident executable regions without backing files, and anomalous API calls (VirtualAllocEx, WriteProcessMemory, CreateRemoteThread). Monitor for T1555.003 by alerting on processes accessing browser credential database files (e.g., 'Login Data' in Chrome profile directories) from unexpected processes. Enable behavioral detection rules covering script interpreter launches, obfuscated command execution, and sandbox evasion behaviors (T1497), aligned with NIST AU-2 (Audit Events) log selection criteria. Review LSASS access events for T1003.001 patterns. No confirmed IOC hashes or network indicators are available from the sourced reporting.
3. Step 3: Eradication. No patch applies; this is a malware campaign. Force browser credential store rotation: require users to change passwords for all accounts saved in browser password managers. Disable browser-native password saving via Group Policy or MDM and migrate to a dedicated enterprise password manager. Apply CIS 5.2 (Use Unique Passwords) and CIS 6.3 (Require MFA for Externally-Exposed Applications) to limit the utility of any harvested credentials. Apply CISA DEFEND Credential Rotation practices and require password changes for all accounts accessible via browser-stored credentials.
4. Step 4: Recovery. Validate that EDR behavioral detection coverage for in-memory threats is active across all endpoints. Confirm CISA DEFEND Multi-Factor Authentication enforcement on all externally facing applications and identity providers; harvested credentials are significantly less useful against MFA-protected accounts. Audit active sessions across SaaS and VPN platforms for anomalous logins following the detection window. Monitor CISA DEFEND Local Account Monitoring for unauthorized account activity post-incident.
5. Step 5: Post-Incident. This campaign exposes a control gap in reliance on signature-based or file-scanning endpoint detection. Evaluate EDR coverage for memory-resident threat detection and process injection (T1055). Assess whether browser-native credential storage is an acceptable risk in your environment given NIST AC-3 (Access Enforcement) and CWE-522. Implement CISA DEFEND User Account Permissions to restrict browser access to credential store files at the OS level where feasible. Review the cadence of endpoint behavioral telemetry reviews (NIST AU-6, Audit Record Review, Analysis, and Reporting) to ensure security events are analyzed promptly.

IR / Forensic Enrichment

Triage Priority	URGENT

Escalation Criteria	Escalate immediately to CISO and legal counsel if IdP or SaaS audit logs confirm successful authentication using harvested credentials from any account with access to PII, PHI, financial records, or privileged infrastructure — these conditions may trigger breach notification obligations under GDPR, HIPAA, or state privacy statutes, and the absence of disk artifacts means the investigation window for volatile evidence is extremely narrow.
Recovery Notes	Post-containment recovery must prioritize a 30-day continuous monitoring window across all IdP, SaaS, and VPN authentication logs, as Phantom Stealer-harvested credentials are frequently operationalized by secondary actors with significant delay after initial exfiltration. Validate that MFA enforcement is confirmed active — not merely configured — on all externally-facing applications by reviewing bypass exceptions and legacy authentication protocol allowances, which represent the most viable exploitation path for harvested browser-stored credentials. Confirm that browser-native password saving is disabled organization-wide via GPO or MDM policy and that an enterprise password manager has been deployed before considering the recovery phase closed.
Forensic Artifacts	Sysmon Event ID 25 (ProcessTampering) and Event ID 10 (ProcessAccess) logs from Microsoft-Windows-Sysmon/Operational, specifically entries where a non-browser process accesses Chrome 'C:\Users*\AppData\Local\Google\Chrome\User Data\Default\Login Data' or Firefox 'C:\Users*\AppData\Roaming\Mozilla\Firefox\Profiles*.default*\logins.json' — the definitive host-side behavioral indicator for Phantom Stealer's credential harvesting mechanism | Memory image sections flagged by Volatility3 malfind plugin showing MEM_PRIVATE + PAGE_EXECUTE_READWRITE regions with no corresponding file on disk — Phantom Stealer's fileless execution model will appear here and not in any on-disk PE artifact, making this the primary forensic evidence of infection | Windows Security Event Log Event ID 4688 (Process Creation) with full command-line auditing enabled, filtered for script interpreters (powershell.exe, wscript.exe, mshta.exe, cscript.exe) launched without an associated interactive user session or spawned by unusual parent processes such as browser helper objects or Office applications — captures the initial execution vector for in-memory staging | IdP and SaaS platform authentication logs (Okta System Log event type 'user.session.start', Azure AD Sign-In Logs, Google Workspace Admin SDK reports/activity) filtered for the compromise window, specifically flagging new device IDs, impossible-travel events, or legacy authentication protocol usage against accounts whose credentials were confirmed stored in the compromised browser credential stores | Chrome 'Login Data' SQLite database and Firefox 'logins.json' forensic copies captured from affected hosts prior to credential rotation, preserving the 'date_password_modified' and 'time_password_changed' fields that establish the pre-compromise credential inventory and the scope of accounts requiring mandatory password resets

Per-Action IR Details

Step 1: Containment — Identify endpoints where browser credential stores may have been accessed outside normal user sessions. Prioritize systems with administrative privileges and those with access to SaaS, VPN, or identity provider portals. Isolate any host exhibiting anomalous in-memory execution patterns pending investigation. Apply NIST AC-6 (Least Privilege) to restrict which accounts can access sensitive browser-stored credentials.

NIST Phase: Containment

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

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

Compensating: Use Sysmon Event ID 10 (ProcessAccess) filtered on target image paths matching Chrome 'Login Data' (C:\Users*\AppData\Local\Google\Chrome\User Data\Default\Login Data) or Firefox 'logins.json'

(C:\Users*\AppData\Roaming\Mozilla\Firefox\Profiles*.default*\logins.json) accessed by non-browser processes. Run: Get-WinEvent -LogName 'Microsoft-Windows-Sysmon/Operational' | Where-Object {$_.Id -eq 10 -and $_.Message -match 'Login Data'} to surface anomalous accessors. Use osquery to enumerate open file handles: SELECT * FROM process_open_files WHERE path LIKE '%Login Data%' OR path LIKE '%logins.json%';

Evidence: BEFORE isolating any host: capture a full memory image using WinPmem or DumpIt to preserve in-memory PE regions with no backing file on disk — the defining artifact of this fileless campaign. Simultaneously capture active TCP connections (Get-NetTCPConnection or netstat -ano) to identify C2 beaconing from injected processes. Export Sysmon Event ID 10 logs (ProcessAccess to browser credential store paths) and Event ID 8 (CreateRemoteThread) from the Microsoft-Windows-Sysmon/Operational channel before network isolation destroys live process state.

Step 2: Detection — Query EDR telemetry for process injection indicators (T1055): unusual parent-child process relationships, memory-resident executable regions without backing files, and anomalous API calls (VirtualAllocEx, WriteProcessMemory, CreateRemoteThread). Monitor for T1555.003 by alerting on processes accessing browser credential database files (e.g., 'Login Data' in Chrome profile directories) from unexpected processes. Enable behavioral detection rules aligned with AU-2 (Event Logging) covering script interpreter launches, obfuscated command execution, and sandbox evasion behaviors (T1497). Review LSASS access events for T1003.001 patterns. No confirmed IOC hashes or network indicators are available from the sourced reporting.

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: Deploy Sysmon with SwiftOnSecurity or Olaf Hartong's modular config to capture Event ID 1 (ProcessCreate), ID 8 (CreateRemoteThread), ID 10 (ProcessAccess), and ID 25 (ProcessTampering — image mismatch, indicating a stomped PE in memory). Use the Sigma rule 'proc_access_win_browser_credential_dump.yml' (available in SigmaHQ repository) to detect unexpected processes opening Chrome Login Data or Firefox logins.json. For LSASS access patterns, use Sysmon Event ID 10 filtered on TargetImage=lsass.exe from non-system callers. Schedule a nightly osquery query: SELECT p.name, p.pid, pof.path FROM processes p JOIN process_open_files pof ON p.pid = pof.pid WHERE pof.path LIKE '%Login Data%' AND p.name NOT IN ('chrome.exe','msedge.exe','brave.exe');

Evidence: No disk artifacts will be present for this fileless campaign; evidence exists only in volatile memory and behavioral telemetry. Capture: (1) Sysmon EVTX logs from Microsoft-Windows-Sysmon/Operational covering Event IDs 1, 8, 10, and 25; (2) Windows Security Event Log Event ID 4688 (Process Creation with command-line auditing enabled) for script interpreter invocations (powershell.exe, wscript.exe, mshta.exe) spawned without user context; (3) a memory image of any process exhibiting a memory region flagged as MEM_PRIVATE + PAGE_EXECUTE_READWRITE with no mapped file on disk — this is the hallmark of Phantom Stealer's in-memory execution model. Capture BEFORE any process termination or host isolation.

Step 3: Eradication — No patch applies; this is a malware campaign. Force browser credential store rotation: require users to change passwords for all accounts saved in browser password managers. Disable browser-native password saving via Group Policy or MDM and migrate to a dedicated enterprise password manager. Apply CIS 5.2 (Use Unique Passwords) and CIS 6.3 (Require MFA for Externally-Exposed Applications) to limit the utility of any harvested credentials. Enforce D3-CRO (Credential Rotation) for all accounts accessible via browser-stored credentials.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 5.2 (Use Unique Passwords), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.2 (Establish an Access Revoking Process), NIST AC-2 (Account Management)

Compensating: Export the list of saved site origins from Chrome Login Data using sqlite3: sqlite3 'C:\Users\USERNAME\AppData\Local\Google\Chrome\User Data\Default\Login Data' 'SELECT origin_url, username_value FROM logins;' — use this to scope which downstream accounts require forced password resets. For Firefox, parse logins.json with python3 -c "import json,sys; [print(e['hostname'],e['encryptedUsername']) for e in json.load(open(sys.argv[1]))['logins']]" to enumerate targeted origins without enterprise tooling. Disable browser password saving via Group Policy key: HKLM\Software\Policies\Google\Chrome\PasswordManagerEnabled = 0 (Chrome) and lockPref('signon.rememberSignons', false); in Firefox mozilla.cfg.

Evidence: BEFORE forcing credential rotation across SaaS and VPN platforms, capture active authenticated session tokens from IdP audit logs (e.g., Okta System Log, Azure AD Sign-In Logs, Google Workspace Admin SDK) for the suspected compromise window — these will be invalidated by forced session termination and are critical for establishing the lateral movement blast radius. Also export Chrome Login Data and Firefox logins.json from affected hosts as forensic copies BEFORE any user-initiated password changes alter the 'date_password_modified' fields that help establish the pre-compromise credential inventory.

Step 4: Recovery — Validate that EDR behavioral detection coverage for in-memory threats is active across all endpoints. Confirm MFA enforcement (D3-MFA) on all externally facing applications and identity providers — harvested credentials are significantly less useful against MFA-protected accounts. Audit active sessions across SaaS and VPN platforms for anomalous logins following the detection window. Monitor D3-LAM (Local Account Monitoring) for unauthorized account activity post-incident.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.4 (Require MFA for Remote Network Access), CIS 6.5 (Require MFA for Administrative Access), NIST AC-17 (Remote Access), NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: For teams without a SIEM, query SaaS authentication logs directly via CLI: use the Microsoft Graph API (Get-MgAuditLogSignIn) or Okta's /api/v1/logs endpoint filtered by the compromise window to surface impossible-travel events, new device fingerprints, or logins from ASNs inconsistent with employee geolocations. For VPN, parse gateway authentication logs for sessions originating from IP ranges not seen in the 90-day baseline. Schedule a 30-day monitoring window using a cron job or Task Scheduler to run these queries daily and alert on new anomalies, given that harvested credentials may be operationalized days after exfiltration.

Evidence: At recovery entry, confirm the following volatile and semi-volatile evidence has already been collected from all investigated hosts: memory images, Sysmon EVTX exports, and process-open-file snapshots. During recovery monitoring, preserve IdP and SaaS authentication logs with timestamps extending 30 days beyond the last confirmed malicious activity — Phantom Stealer-harvested credentials may be sold and used by secondary threat actors with significant delay relative to the initial infection window.

Step 5: Post-Incident — This campaign exposes a control gap in reliance on signature-based or file-scanning endpoint detection. Evaluate EDR coverage for memory-resident threat detection and process injection (T1055). Assess whether browser-native credential storage is an acceptable risk in your environment given NIST AC-3 (Access Enforcement) and CWE-522. Implement D3-UAP (User Account Permissions) to restrict browser access to credential store files at the OS level where feasible. Review AU-6 (Audit Record Review, Analysis, and Reporting) cadence for endpoint behavioral telemetry.

NIST Phase: Post Incident

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

Controls: NIST AC-3 (Access Enforcement), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Implement a YARA rule targeting MEM_PRIVATE + PAGE_EXECUTE_READWRITE regions without a backing file on disk, deployable via YARA-X or Volatility3's malfind plugin, to establish a repeatable hunt for Phantom Stealer's in-memory execution model across the environment on a weekly basis. Draft a Sigma detection rule (class: process_access) matching any process other than chrome.exe, msedge.exe, brave.exe, firefox.exe opening 'Login

Data' or 'logins.json' paths, and integrate it into Windows Event Forwarding subscriptions for ongoing detection without EDR licensing. Document browser credential store file ACLs and apply icacls to restrict 'Login Data' read access to the owning user SID only, denying access to SYSTEM and other user contexts where operationally feasible.

Evidence: Post-incident artifacts to retain for lessons-learned and threat intelligence sharing: (1) Sysmon EVTX archives covering the full compromise window; (2) memory images from confirmed affected hosts for submission to internal threat intelligence or sharing via ISAC if permitted; (3) IdP and SaaS authentication logs for the 90-day pre-detection baseline plus 30-day post-remediation window; (4) the forensic copies of Chrome Login Data and Firefox logins.json captured during eradication, which establish the credential inventory at time of compromise. Note: no confirmed IOC hashes or network indicators exist for this campaign per available sourcing — detection artifacts are behavioral, not signature-based, and must be documented as such in the post-incident report.

Detection Guidance

No confirmed IOC hashes, domains, or IP addresses were available from the sourced reporting; the following guidance is behavioral. Query EDR for process injection patterns (T1055): flag processes with memory-resident executable pages lacking a mapped file on disk, and alert on VirtualAllocEx, WriteProcessMemory, or CreateRemoteThread calls originating from scripting interpreters (T1059). Detect T1555.003 by monitoring file system access to browser credential stores, specifically Chrome 'Login Data' SQLite files and Firefox 'logins.json', from any process other than the browser itself. Alert on obfuscated script execution (T1027/T1140) and processes that terminate abnormally after short execution windows, which may indicate sandbox evasion (T1497). For SIEM, correlate short-lived process creation events with subsequent outbound network connections to unknown destinations. Cross-reference against NIST AU-2 event categories: logon events, process creation, and object access. Given medium confidence in behavioral details from available sourcing, treat these as detection hypotheses requiring tuning against your environment's baseline before production deployment.

Framework Mappings

MITRE-ATTACK

- **T1055** — Process Injection
- **T1027** — Obfuscated Files or Information
- **T1140** — Deobfuscate/Decode Files or Information
- **T1497** — Virtualization/Sandbox Evasion
- **T1059** — Command and Scripting Interpreter
- **T1555.003** — Credentials from Web Browsers
- **T1003.001** — LSASS Memory

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

- **IA-5** — Authenticator Management

OWASP-TOP10-2021

- **A04:2021** — Insecure Design
- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **5.2** — Use Unique Passwords
- **6.3** — Require MFA for Externally-Exposed Applications
- **8.2** — Collect Audit Logs

HIPAA-SECURITY

- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

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

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1055	Process Injection	Defense-Evasion
T1027	Obfuscated Files or Information	Defense-Evasion
T1140	Deobfuscate/Decode Files or Information	Defense-Evasion
T1497	Virtualization/Sandbox Evasion	Defense-Evasion
T1059	Command and Scripting Interpreter	Execution
T1555.003	Credentials from Web Browsers	Credential-Access
T1003.001	LSASS Memory	Credential-Access

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/cyberattacks-data-breaches/fileless-pha...	T3

Source	URL	Tier
Vendor refuses CVEs for third-party findings. Anything you can do?	https://www.reddit.com/r/cybersecurity/comments/1spov5s/vendor_refu...	T3
Findings Filters - Tenable documentation	https://docs.tenable.com/vulnerability-management/Content/Explore/f...	T3
Known Exploited Vulnerabilities Catalog | CISA	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1
Zero-Day Vulnerabilities in the Browser: A Growing Crisis - Seraphic	https://seraphicsecurity.com/resources/blog/zero-day-vulnerabilitie...	T3

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