INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-07 06:38 UTC

VoidStealer Breaks Chrome's App-Bound Encryption: Credential Theft Defense Weakened Again

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0285
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Google Chrome (App-Bound Encryption implementation; specific version range not confirmed in available sources)
Published	2026-05-06T17:19:11
Discovery Source	Rss

Executive Summary

VoidStealer, an infostealer trojan, has implemented a working bypass of Google Chrome's App-Bound Encryption (ABE), a credential-protection control introduced in Chrome 127 (mid-2024). The bypass allows attackers to extract saved passwords and session cookies from Chrome without triggering ABE's process-binding defenses, effectively neutralizing a control many organizations rely on to protect browser-stored credentials. Any workforce using Chrome with saved credentials or active sessions is at elevated risk of credential theft and session hijacking, regardless of whether ABE was considered part of their browser security posture.

Technical Analysis

VoidStealer is an infostealer trojan that bypasses Chrome's App-Bound Encryption (ABE), a defense introduced in Chrome 127 designed to bind DPAPI-based credential and cookie decryption to the browser process. ABE was Google's mitigation against the wave of infostealer campaigns targeting Chrome's DPAPI credential store. VoidStealer's technique circumvents process-binding controls to exfiltrate saved credentials and session cookies without triggering ABE protections. No CVE has been assigned; this is a technique-based bypass, not a discrete software vulnerability. Relevant CWEs: CWE-693 (Protection Mechanism Failure), CWE-312 (Cleartext Storage of Sensitive Information), CWE-522 (Insufficiently Protected Credentials), CWE-284 (Improper Access Control). MITRE ATT&CK techniques include T1555.003 (Credentials from Web Browsers), T1539 (Steal Web Session Cookie), T1134 (Access Token Manipulation), T1574 (Hijack Execution Flow), T1027 (Obfuscated Files

or Information), and T1056 (Input Capture). No vendor-confirmed patch or ABE update has been announced in available sources as of this report. Threat actor attribution beyond the malware family name is unconfirmed. Note: The Forbes, CIS Advisory, SecPod, and YouTube sources listed in this item's bibliography reference separate Chrome zero-day vulnerabilities from April 2026 and should not be conflated with this VoidStealer ABE bypass report.

Action Checklist

1. Step 1: Containment. Audit Chrome deployments across the environment and identify systems where Chrome credential storage (saved passwords, session cookies) is enabled. Disable Chrome's built-in password manager via enterprise policy (PasswordManagerEnabled = false on Windows; see Chrome enterprise policy documentation for macOS and Linux equivalents) to remove stored credentials as an exfiltration target while assessment is underway.
2. Step 2: Detection. Hunt for VoidStealer indicators in endpoint telemetry. Look for process injection patterns (T1134, T1574) targeting chrome.exe, unusual child processes spawned by or injecting into Chrome, and outbound connections following browser activity. Review EDR/AV logs for file writes or memory access events against Chrome's User Data directory (AppData\Local\Google\Chrome\User Data\). No confirmed public IOCs are available from current sources; flag any unknown processes accessing Chrome credential stores.
3. Step 3: Eradication. Remove reliance on Chrome's built-in credential storage as a credential protection control. Migrate saved credentials to an enterprise password manager that does not depend on browser-native storage. Apply Chrome enterprise policies to enforce managed credential storage. Monitor Google's security blog and Chrome release notes for any ABE hardening updates and apply promptly upon release.
4. Step 4: Recovery. After policy changes are deployed, verify via enterprise policy reporting that PasswordManagerEnabled is disabled across all managed endpoints. Rotate credentials and invalidate session cookies for any accounts where Chrome-stored credentials may have been present on potentially compromised hosts. Re-image hosts where VoidStealer infection is suspected rather than attempting in-place remediation.
5. Step 5: Post-Incident. This bypass exposes a systemic control gap: browser-native credential storage cannot be treated as a secure credential vault. Review your credential storage strategy and enforce enterprise password manager adoption. Evaluate whether your EDR solution detects process injection targeting browser credential stores (test against T1555.003 and T1539 ATT&CK techniques). Update browser security baselines to remove dependency on ABE as a standalone credential protection control.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to CISO and legal/privacy counsel immediately if Sysmon or EDR telemetry confirms a non-Chrome process accessed Login Data or Cookies files on any host storing credentials for systems that process PII, PHI, or financial data, as this constitutes a likely credential exfiltration event triggering breach notification assessment under GDPR Article 33, HIPAA 45 CFR §164.412, or applicable state data breach statutes.

Recovery Notes	After credential rotation and policy enforcement, monitor IdP/SSO authentication logs (Okta, Azure AD, Google Workspace) for anomalous login events — specifically successful authentications from new geolocations, new device fingerprints, or outside business hours — for a minimum of 30 days, as VoidStealer-harvested session cookies may be replayed against SaaS targets well after the initial exfiltration window. Verify Chrome enterprise policy enforcement weekly for the first month via osquery or MDM compliance reporting to detect any policy rollback or new Chrome installations that bypass the PasswordManagerEnabled=false control. Re-baseline your browser security configuration to explicitly treat Chrome App-Bound Encryption as a defense-in-depth layer only, not a standalone credential protection control, and document this risk acceptance formally in your risk register.
Forensic Artifacts	Chrome Login Data SQLite file (AppData\Local\Google\Chrome\User Data\Default\Login Data) — contains the encrypted password store that VoidStealer's ABE bypass directly targets; last-accessed timestamp and any shadow copies indicate whether the file was read by a non-Chrome process outside of normal browser operation | Chrome Local State file (AppData\Local\Google\Chrome\User Data\Local State) — contains the 'encrypted_key' field holding the AES key wrapped in DPAPI that Chrome uses for v10/v11 credential encryption; VoidStealer's ABE bypass reads this file to reconstruct the decryption key outside of Chrome's process binding, making unexpected access to this file the most direct indicator of the bypass technique | Chrome Cookies file (AppData\Local\Google\Chrome\User Data\Default\Network\Cookies) — contains active session tokens encrypted under ABE; unauthorized reads of this file by non-Chrome processes indicate session cookie harvesting consistent with T1539 (Steal Web Session Cookie) | Sysmon Event ID 10 (ProcessAccess) logs filtered for GrantedAccess 0x10 (VM_READ) or 0x1010 targeting chrome.exe as TargetImage from any SourceImage other than chrome.exe, Google Update (GoogleUpdate.exe), or whitelisted AV — this is the primary telemetry artifact for detecting VoidStealer's process injection approach to bypassing ABE's process-binding defense | Windows Prefetch files (%SystemRoot%\Prefetch*.pf) for unknown executables with creation or last-run timestamps correlating to Chrome User Data directory access events — VoidStealer components executing from %TEMP% or %APPDATA% staging paths will generate prefetch entries that persist after binary deletion, providing execution evidence even on fully wiped endpoints

Per-Action IR Details

Step 1: Containment — Audit Chrome deployments across the environment and identify systems where Chrome credential storage (saved passwords, session cookies) is enabled. Disable Chrome's built-in password manager via enterprise policy (PasswordManagerEnabled = false) to remove stored credentials as an exfiltration target while assessment is underway.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-6 (Configuration Settings), NIST SI-4 (System Monitoring), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Without MDM/enterprise policy infrastructure, deploy a registry-enforced GPO manually or via PowerShell: Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Google\Chrome' -Name 'PasswordManagerEnabled' -Value 0 -Type DWord. Run this across endpoints via PsExec or a simple PowerShell remoting loop (Invoke-Command -ComputerName $hosts -ScriptBlock {...}). Verify enforcement by checking HKLM:\SOFTWARE\Policies\Google\Chrome on each host. Free tool: osquery — query SELECT * FROM registry WHERE path LIKE 'HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome%' to confirm policy presence at

scale.

Evidence: Before disabling the password manager, image or export Chrome's Login Data SQLite file (AppData\Local\Google\Chrome\User Data\Default\Login Data) and Cookies file (AppData\Local\Google\Chrome\User Data\Default\Network\Cookies) from suspect hosts to preserve forensic state. Document last-modified timestamps on both files — VoidStealer's ABE bypass requires reading these files outside of Chrome's process binding, so unexpected recent access timestamps or shadow copies of these files by non-Chrome processes are key indicators. Also capture VSS snapshots of the User Data directory before any policy changes overwrite credential state.

Step 2: Detection — Hunt for VoidStealer indicators in endpoint telemetry. Look for process injection patterns (T1134, T1574) targeting chrome.exe, unusual child processes spawned by or injecting into Chrome, and outbound connections following browser activity. Review EDR/AV logs for file writes or memory access events against Chrome's User Data directory (AppData\Local\Google\Chrome\User Data\). No confirmed public IOCs are available from current sources; flag any unknown processes accessing Chrome credential stores.

NIST Phase: Detection Analysis

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

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysmon with a config that captures Event ID 10 (ProcessAccess) targeting chrome.exe and Event ID 11 (FileCreate) under AppData\Local\Google\Chrome\User Data\. Use this Sysmon filter in your config XML: chrome.exe. Forward Sysmon logs to a local aggregator (e.g., Winlogbeat to an ELK stack or even Windows Event Forwarding to a collector). Use the public Sigma rule sigma/rules/windows/process_access/proc_access_win_browser_credential_access.yml as a detection baseline — adapt it to also match Login Data and Cookies file paths. For network hunting without a SIEM, run Wireshark or tcpdump on a span port filtering for large POST requests or TLS connections to non-browser processes immediately after Chrome activity: tcpdump -i eth0 -w voidstealer_hunt.pcap 'port 443 and not src port 443'.

Evidence: Collect Sysmon Event ID 10 logs showing GrantedAccess flags (0x10 = VM_READ) on chrome.exe from any process other than chrome.exe itself, Google Update, or known AV. Pull Windows Security Event Log Event ID 4688 (Process Creation with command line logging enabled) filtering on processes that access %LOCALAPPDATA%\Google\Chrome\User Data\Default\Login Data or \Cookies. Capture network flow data (NetFlow or full PCAP) for outbound connections from non-browser processes in the 60-minute window following any detected Login Data file access — VoidStealer exfiltrates extracted credentials over C2 channels immediately after extraction. Also collect MFT ($MFT) records for the Chrome User Data directory to detect file reads not visible in standard audit logs.

Step 3: Eradication — Remove reliance on Chrome's built-in credential storage as a credential protection control. Migrate saved credentials to an enterprise password manager that does not depend on browser-native storage. Apply Chrome enterprise policies to enforce managed credential storage. Monitor Google's security blog and Chrome release notes for any ABE hardening updates and apply promptly upon release.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-6 (Configuration Settings), NIST IA-5 (Authenticator Management), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 5.2 (Use Unique Passwords)

Compensating: For teams without an enterprise password manager budget, use Bitwarden (free tier, self-hostable) as an immediate interim replacement — it operates entirely outside Chrome's native storage and is not subject to ABE bypass. To force-clear Chrome's stored credentials at scale without enterprise tooling, run: Remove-Item -Path "$env:LOCALAPPDATA\Google\Chrome\User Data\Default\Login Data" -Force via PowerShell remoting, then enforce PasswordManagerEnabled=false via registry. Create a YARA rule to scan for VoidStealer dropper artifacts in common staging paths (%TEMP%, %APPDATA%, Startup folders) using YARA patterns matching strings associated with

Chrome credential extraction (e.g., references to 'Login Data', 'os_crypt', 'v10' DPAPI blob headers in untrusted binaries).

Evidence: Before deleting Chrome credential stores, preserve forensic copies of Login Data (SQLite — contains encrypted passwords and the encryption key metadata), Cookies (SQLite — contains session tokens encrypted with ABE or DPAPI), and the Local State file (AppData\Local\Google\Chrome\User Data\Local State — contains the AES key used for v10/v11 DPAPI-wrapped credential encryption, which VoidStealer's ABE bypass specifically targets). The Local State file's 'encrypted_key' field is the exact artifact VoidStealer reads to reconstruct the decryption key outside of Chrome's process binding — preserve this file with hash verification before any remediation touches the User Data directory.

Step 4: Recovery — After policy changes are deployed, verify via enterprise policy reporting that PasswordManagerEnabled is disabled across all managed endpoints. Rotate credentials and invalidate session cookies for any accounts where Chrome-stored credentials may have been present on potentially compromised hosts. Re-image hosts where VoidStealer infection is suspected rather than attempting in-place remediation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST IA-5 (Authenticator Management), NIST CM-6 (Configuration Settings), NIST SI-2 (Flaw Remediation), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Verify policy enforcement without MDM by running the following osquery query across all endpoints: SELECT * FROM registry WHERE path = 'HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\PasswordManagerEnabled' AND data = '0'. For credential rotation prioritization, use PowerShell to query Chrome's Login Data SQLite file (before deletion) to enumerate which accounts had stored credentials: invoke sqlite3.exe against the Login Data file with SELECT origin_url, username_value FROM logins — this gives you the exact account list requiring forced password resets. Prioritize SaaS and identity provider accounts (SSO, Okta, Azure AD) first as those session cookies enable lateral movement far beyond the initial compromised host.

Evidence: Before re-imaging, collect a full memory dump (using WinPmem or Magnet RAM Capture — both free) from any host suspected of active VoidStealer infection to preserve in-memory credential material and any injected code that may not be present on disk. Capture the Windows Prefetch files (%SystemRoot%\Prefetch) for evidence of VoidStealer executable runs — prefetch entries persist across reboots and will show execution timestamps even after the malware binary is deleted. Also collect the SYSTEM, SAM, and SECURITY registry hives from suspect hosts to support post-recovery forensic analysis of any local credential access.

Step 5: Post-Incident — This bypass exposes a systemic control gap: browser-native credential storage cannot be treated as a secure credential vault. Review your credential storage strategy and enforce enterprise password manager adoption. Evaluate whether your EDR solution detects process injection targeting browser credential stores (test against T1555.003 and T1539 ATT&CK techniques). Update browser security baselines to remove dependency on ABE as a standalone credential protection control.

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Conduct EDR detection gap testing against T1555.003 (Credentials from Web Browsers) and T1539 (Steal Web Session Cookie) using Atomic Red Team's free test library — run atomics for T1555.003 and T1539 on a test endpoint and verify your EDR/Sysmon alerting fires. If no EDR is available, create a Sigma rule targeting Sysmon Event ID 10 (process access to chrome.exe with VM_READ from non-browser processes) and test it against your Windows Event Forwarding setup. Subscribe to Google's Chrome Releases blog

(https://chromereleases.googleblog.com/) and the Chrome Enterprise release notes for ABE hardening updates — set a calendar-based review trigger so any Chrome release above the current version is evaluated within 72 hours of release for ABE-relevant changes. Add MITRE ATT&CK T1555.003 and T1539 to your threat model as persistent detection requirements, not one-time responses.

Evidence: Document the full timeline of Chrome Login Data and Cookies file access events across the incident window as the primary forensic record of what credentials were exposed. Preserve this as the evidentiary basis for any breach notification assessment — the Login Data SQLite file's logins table enumerates every credential that was at risk by URL and username, and the Cookies table identifies every active session token that may have been harvested. Retain all collected Sysmon logs, memory images, and prefetch artifacts per your evidence retention policy (NIST AU-11 — Audit Record Retention) to support any regulatory or legal review.

Detection Guidance

No confirmed public IOCs for VoidStealer are available in current sources. Focus detection on behavioral indicators. Monitor EDR telemetry for: (1) processes other than chrome.exe or legitimate browser components accessing Chrome's User Data directory, particularly Login Data, Cookies, and Local State files; (2) process injection events targeting chrome.exe, look for OpenProcess/VirtualAllocEx/WriteProcessMemory API call sequences against browser processes; (3) unusual outbound network connections from or immediately following Chrome process activity, particularly to unknown or low-reputation destinations. SIEM queries should correlate file access events on Chrome credential store paths with process creation events for unsigned or low-prevalence executables. On Windows, Sysmon Event ID 10 (ProcessAccess) targeting chrome.exe and Event ID 11 (FileCreate) in Chrome profile directories are relevant sources. Flag any endpoint where Chrome's User Data directory is accessed by a process outside the browser's own process tree. Note: ABE bypass techniques generally require the malware to execute on the host first, initial access detection (phishing, malicious downloads) remains the highest-leverage detection point in the kill chain.

Framework Mappings

MITRE-ATTACK

- **T1134** — Access Token Manipulation
- **T1574** — Hijack Execution Flow
- **T1027** — Obfuscated Files or Information
- **T1056** — Input Capture
- **T1555.003** — Credentials from Web Browsers
- **T1539** — Steal Web Session Cookie

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **IA-5** — Authenticator Management
- **AC-3** — Access Enforcement
- **SC-13** — Cryptographic Protection
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

- **A04:2021** — Insecure Design
- **A07:2021** — Identification and Authentication Failures
- **A01:2021** — Broken Access Control

CIS-V8

- **5.2** — Use Unique Passwords
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **6.3** — Require MFA for Externally-Exposed Applications

HIPAA-SECURITY

- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication
- **164.312(e)(1)** — Transmission Security

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.8.24** — Use of cryptography

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1134	Access Token Manipulation	Defense-Evasion
T1574	Hijack Execution Flow	Persistence
T1027	Obfuscated Files or Information	Defense-Evasion
T1056	Input Capture	Collection
T1555.003	Credentials from Web Browsers	Credential-Access
T1539	Steal Web Session Cookie	Credential-Access

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/endpoint-security/yet-another-way-bypas...	T3
Google Issues Zero-Day Attack Alert For 3.5 Billion Chrome Users	https://www.forbes.com/sites/daveywinder/2026/04/03/google-issues-z...	T3
A Vulnerability in Google Chrome Could Allow for Arbitrary Code ...	https://www.cisecurity.org/advisory/a-vulnerability-in-google-chrom...	T3
Update Chrome Now to Fix a ZERO-DAY Security Vulnerability ...	https://www.youtube.com/watch?v=hhIUKzw_iMI	T3
Chrome Security Update: Google Fixes Another Actively Exploited ...	https://www.secpod.com/blog/chrome-security-update-google-fixes-ano...	T3

Generated 2026-05-07 06:38 UTC by TJS Security Command Center