INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-27 06:09 UTC

Malicious Chrome Extension Campaign Steals Session Cookies via Native Messaging Abuse

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0584
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Google Chrome (all versions on Windows), Windows systems
Published	2026-06-26
Discovery Source	Gemini

Executive Summary

Attackers are distributing a malicious Google Chrome extension through phishing emails to steal session cookies, enabling full account takeover on any site where the victim is logged in, without needing passwords. Any Windows system running Chrome is vulnerable if a user installs the malicious extension, regardless of Chrome version, because the attack exploits a legitimate browser feature rather than a software flaw. Organizations face immediate risk of unauthorized access to corporate email, SaaS platforms, and financial accounts without traditional authentication controls providing protection.

Technical Analysis

This campaign delivers a malicious Chrome extension via spear-phishing emails with fake PDF attachments (MITRE T1566.001). Once installed (T1176), the extension abuses Chrome's Native Messaging API, a documented inter-process communication mechanism, to bridge the browser sandbox and the host OS, spawning PowerShell processes (T1059.001) leveraging a design feature that is not classified as a sandbox escape. The extension performs keylogging or form-grabbing (T1056) and directly harvests session cookies (T1539) from the browser's cookie store, enabling browser session hijacking (T1185) for authenticated account takeover. No CVE is assigned; root weakness classifications are CWE-284 (Improper Access Control), CWE-693 (Protection Mechanism Failure), and CWE-539 (Use of Persistent Cookies Containing Sensitive Information). Because no software vulnerability is exploited, no vendor patch exists. Affected scope is all Chrome versions on Windows. Qualitative severity is High. Source material is limited to T3 outlets; no official CISA KEV, NVD entry, or Google security advisory was available at ingestion.

Action Checklist

1. Immediate Actions: Audit and Block Extensions. Audit all Chrome extensions installed across managed endpoints immediately; use Group Policy (ExtensionInstallBlocklist) or your MDM to block all extensions not on an approved allowlist. Revoke and rotate session tokens for any accounts accessed from endpoints where unknown extensions are present.
2. Detection. Query EDR telemetry for chrome.exe or chrome_child.dll spawning powershell.exe or cmd.exe as a child process. Review Windows Event ID 4688 (process creation) for PowerShell invocations with chrome.exe as parent. Check Chrome extension directories (%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions) for unsigned or recently installed extensions. Correlate with audit event logging to identify anomalous cookie-access or native messaging host registrations (HKCU\Software\Google\Chrome\NativeMessagingHosts).
3. Eradication. Remove any unauthorized or unrecognized Chrome extensions from all endpoints. Revoke active session tokens for affected accounts across SaaS platforms, corporate email, and VPN portals. Enforce an extension allowlist policy via Google Workspace Admin Console or Windows Group Policy (ExtensionInstallAllowlist). Ensure unauthorized software is removed and documented. Disable native messaging for non-business-critical applications where feasible.
4. Recovery. Reissue session tokens and force re-authentication for all users on affected endpoints. Validate that no unauthorized native messaging host entries remain in the Windows registry. Monitor audit logs for anomalous authentication events post-remediation, particularly logins from unexpected geolocations or devices following the incident window. Confirm EDR alerts for chrome.exe spawning shells have cleared.
5. Post-Incident. This attack exposed gaps in extension governance, endpoint process monitoring, and phishing resilience. Implement a formal browser extension inventory and approval process. Deploy multi-factor authentication (MFA) on all externally-exposed SaaS platforms to limit session cookie value to attackers. Evaluate local account anomaly monitoring for detection. Add Native Messaging API abuse to your threat hunting hypothesis library.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to senior IR leadership and legal/privacy counsel immediately if evidence of successful session cookie replay is confirmed (i.e., attacker-controlled logins to corporate email, HR systems, or financial platforms are detected), as this constitutes unauthorized access to organizational systems and may trigger breach notification obligations under applicable data protection regulations (e.g., GDPR 72-hour notification, US state breach laws) if PII or PHI was accessible in the compromised sessions.

Recovery Notes	Because this attack harvests session cookies rather than credentials, standard password rotation alone does not revoke attacker access — all active session tokens for affected users must be explicitly invalidated via each platform's admin console before recovery is complete. Monitor authentication logs across all SaaS platforms, corporate email, and VPN for a minimum of 30 days post-remediation for impossible-travel events, new-device logins, or logins from IPs not associated with managed endpoints, as adversaries may have cached stolen cookies for delayed replay. Confirm that Chrome's ExtensionInstallAllowlist Group Policy is enforced and verified via a follow-up endpoint compliance scan before declaring recovery complete.
Forensic Artifacts	Chrome extension directory at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\\` — contains the malicious extension's manifest.json (declaring `cookies` and `nativeMessaging` permissions), background scripts performing cookie access, and content scripts injected into web pages; primary malware artifact specific to this campaign. | Windows registry key `HKCU\Software\Google\Chrome\NativeMessagingHosts\` — records the name and executable path of the native messaging host registered by the malicious extension to exfiltrate cookie data out of the Chrome sandbox to the host OS; present only when the attack has reached the native messaging abuse stage. | Chrome SQLite Cookies database at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Cookies` — records all session cookies stored by Chrome including `host_key`, `name`, `encrypted_value`, and `expires_utc`; forensic copy establishes which session tokens were present and accessible to the extension at time of compromise. | Windows Security Event Log Event ID 4688 (Process Creation) entries where `ParentProcessName` = `chrome.exe` and `NewProcessName` = `powershell.exe` or `cmd.exe` — documents the process-lineage anomaly produced when the native messaging host executable is launched by Chrome to receive and relay stolen cookie data. | SaaS platform authentication logs (Google Workspace Login Audit, Azure AD Sign-in Logs) filtered for the incident timeframe — reveal whether stolen session cookies were replayed by the attacker from an external IP, evidenced by concurrent sessions from geographically disparate locations or user-agent strings inconsistent with the victim's managed device.

Per-Action IR Details

Containment — Audit all Chrome extensions installed across managed endpoints immediately; use Group Policy (ExtensionInstallBlocklist) or your MDM to block all extensions not on an approved allowlist. Revoke and rotate session tokens for any accounts accessed from endpoints where unknown extensions are present.

NIST Phase: Containment

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

Controls: NIST AC-3 (Access Enforcement), NIST CM-7 (from knowledge base: not present — no CM family entries verified in knowledge base reference), CIS 2.3 (IG1/IG2/IG3) — Address Unauthorized Software, CIS 6.2 (IG1/IG2/IG3) — Establish an Access Revoking Process

Compensating: Without MDM, run the following PowerShell on each endpoint to enumerate installed extension directories and flag recently modified ones: `Get-ChildItem "$env:LOCALAPPDATA\Google\Chrome\User Data\Default\Extensions" | Select-Object Name, LastWriteTime | Sort-Object LastWriteTime -Descending`. Cross-reference against a manually maintained allowlist. For token revocation, use each SaaS platform's admin console (e.g., Google Workspace Admin > Security > Sessions, Microsoft 365 Admin > Active Users > Revoke sessions) directly — no SIEM required.

Evidence: Before revoking session tokens or pushing the ExtensionInstallBlocklist policy, capture: (1) full extension directory listing with timestamps from `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\` for each installed profile; (2) Chrome Preferences and Secure Preferences JSON files at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\` which record extension installation source, update URLs,

and permissions granted; (3) current Windows registry snapshot of `HKCU\Software\Google\Chrome\NativeMessagingHosts` to document any registered native messaging host executables before policy enforcement removes them; (4) active network connections via `netstat -ano` or `Get-NetTCPConnection` to identify any live C2 or data-exfil sessions initiated by the extension's native messaging host process before session revocation severs visibility.

Detection — Query EDR telemetry for chrome.exe or chrome_child.dll spawning powershell.exe or cmd.exe as a child process. Review Windows Event ID 4688 (process creation) for PowerShell invocations with chrome.exe as parent. Check Chrome extension directories (%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions) for unsigned or recently installed extensions. Correlate with AU-2 event logging to identify anomalous cookie-access or native messaging host registrations (HKCU\Software\Google\Chrome\NativeMessagingHosts).

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 (IG1/IG2/IG3) — Collect Audit Logs

Compensating: Deploy Sysmon with a configuration that enables Rule Group for ProcessCreate (Event ID 1) with ParentImage matching `*\chrome.exe` and TargetImage matching `*\powershell.exe` or `*\cmd.exe`. Use the following PowerShell to scan all user profiles for recently modified extension directories (within 7 days): `Get-ChildItem "C:\Users\*\AppData\Local\Google\Chrome\User Data\Default\Extensions" -Recurse -ErrorAction SilentlyContinue | Where-Object {$_.LastWriteTime -gt (Get-Date).AddDays(-7)} | Select-Object FullName, LastWriteTime`. For registry-based native messaging detection without EDR, run: `Get-ChildItem "HKCU:\Software\Google\Chrome\NativeMessagingHosts" -ErrorAction SilentlyContinue` on each endpoint and compare against a known-good baseline.

Evidence: This step is analytical and does not alter live state; however, before acting on findings, capture: (1) a live memory image (using WinPmem or Magnet RAM Capture) if chrome.exe→powershell.exe process lineage is confirmed — the native messaging host executable will be loaded in memory and may not persist to disk after detection; (2) Chrome `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Cookies` (SQLite database) — the malicious extension reads this file to exfiltrate session cookies; a forensic copy preserves which cookies were present and potentially copied at time of compromise; (3) Windows Security Event Log filtered for Event ID 4688 with `ParentProcessName` containing `chrome.exe` and `NewProcessName` containing `powershell.exe` or `cmd.exe`, exported before log rotation; (4) full contents of each installed extension's `manifest.json` to document declared permissions, particularly `cookies`, `nativeMessaging`, and any suspicious `externally_connectable` or background script declarations.

Eradication — Remove any unauthorized or unrecognized Chrome extensions from all endpoints. Revoke active session tokens for affected accounts across SaaS platforms, corporate email, and VPN portals. Enforce an extension allowlist policy via Google Workspace Admin Console or Windows Group Policy (ExtensionInstallAllowlist). Per CIS 2.3, ensure unauthorized software is removed and documented. Disable native messaging for non-business-critical applications where feasible.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 2.3 (IG1/IG2/IG3) — Address Unauthorized Software, CIS 2.1 (IG1/IG2/IG3) — Establish and Maintain a Software Inventory, NIST AC-3 (Access Enforcement), CIS 6.2 (IG1/IG2/IG3) — Establish an Access Revoking Process

Compensating: To remove a specific extension without MDM, use the registry: delete the extension's entry under `HKLM\SOFTWARE\Policies\Google\Chrome\ExtensionInstallBlocklist` and add its Chrome Web Store ID to the blocklist key. Alternatively, manually delete the extension folder from `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\` and remove its entry from the `Preferences` JSON file. To disable native messaging globally without enterprise tooling, delete or rename all entries under `HKCU\Software\Google\Chrome\NativeMessagingHosts` and monitor for re-creation using Sysmon Event ID 13 (Registry Value Set) filtered on that key path.

Evidence: Eradication alters live state — all volatile evidence must be captured before this step. Specifically: (1) acquire a RAM image before removing the extension or killing any associated native messaging host process, as the host executable's in-memory strings may reveal C2 endpoints or exfiltration destinations not present in static artifacts; (2) export the full `HKCU\Software\Google\Chrome\NativeMessagingHosts` registry hive before deletion to preserve the path to the native messaging host executable for malware analysis; (3) copy the malicious extension directory (`%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\\`) in its entirety — including background scripts, content scripts, and the manifest — to forensic storage before removal, as this is the primary malware artifact specific to this campaign; (4) capture all active authenticated sessions via platform admin consoles (Google Workspace session list, Azure AD sign-in logs) before mass revocation to establish which accounts and source IPs were active during the compromise window.

Recovery — Reissue session tokens and force re-authentication for all users on affected endpoints. Validate that no unauthorized native messaging host entries remain in the Windows registry. Monitor AU-6 audit logs for anomalous authentication events post-remediation, particularly logins from unexpected geolocations or devices following the incident window. Confirm EDR alerts for chrome.exe spawning shells have cleared.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AC-12 (Session Termination), CIS 6.3 (IG1/IG2/IG3) — Require MFA for Externally-Exposed Applications, CIS 6.5 (IG1/IG2/IG3) — Require MFA for Administrative Access

Compensating: Without a SIEM, establish a manual authentication monitoring cadence: export sign-in logs daily from Google Workspace (Admin > Reports > Audit > Login) and Microsoft 365 (Azure AD Sign-in logs) for 14 days post-recovery, filtering for logins from IP addresses not seen in the 30 days prior to the incident. Use a simple spreadsheet diff against a pre-incident baseline. For registry validation, schedule a daily automated check via scheduled Task running: `Get-ChildItem 'HKCU:\Software\Google\Chrome\NativeMessagingHosts' -ErrorAction SilentlyContinue | Export-Csv C:\IR\nmh_audit.csv` and alert on any new entries.

Evidence: Recovery does not require new volatile capture if eradication evidence collection was complete. However, before closing monitoring: (1) verify the Chrome `Cookies` SQLite database has been cleared or that session cookies issued before the compromise window have expired — stolen cookies for long-lived sessions (e.g., Google `SAPISID`, Microsoft `ESTSAUTHPERSISTENT`) may still be valid and usable by the attacker even after endpoint remediation; (2) confirm via platform admin consoles that no active sessions exist with device fingerprints or user-agent strings inconsistent with managed endpoints — this detects adversaries using harvested cookies from a remote location; (3) retain the Sysmon Event ID 1 process creation logs and Windows Security Event ID 4688 logs from the detection window for a minimum of 90 days per incident documentation requirements to support post-incident review and potential regulatory reporting.

Post-Incident — This attack exposed gaps in extension governance (CIS 2.1, CIS 2.3), endpoint process monitoring (NIST AU-2, AU-12), and phishing resilience (NIST AC-17). Implement a formal browser extension inventory and approval process. Deploy D3-MFA on all externally-exposed SaaS platforms to limit session cookie value to attackers. Evaluate D3-LAM for local account anomaly detection. Add Native Messaging API abuse to your threat hunting hypothesis library.

NIST Phase: Post Incident

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

Controls: CIS 2.1 (IG1/IG2/IG3) — Establish and Maintain a Software Inventory, CIS 2.3 (IG1/IG2/IG3) — Address Unauthorized Software, NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), CIS 6.3 (IG1/IG2/IG3) — Require MFA for Externally-Exposed Applications, CIS 7.1 (IG1/IG2/IG3) — Establish and Maintain a Vulnerability Management Process

Compensating: For the extension inventory without an enterprise tool, generate a monthly automated report using PowerShell across all endpoints: `Get-ChildItem 'C:\Users\*\AppData\Local\Google\Chrome\User Data\*\Extensions' -Directory -ErrorAction SilentlyContinue | Select-Object FullName, LastWriteTime | Export-Csv C:\IR\extension_inventory.csv`. Publish the output to a shared drive and have a team member diff it against the

approved allowlist monthly. For hunting hypothesis development, author a Sigma rule targeting Sysmon Event ID 1 where `ParentImage|endswith: '\chrome.exe'` and `Image|endswith` matches `powershell.exe`, `cmd.exe`, or `wscript.exe`, and submit it to the team's detection backlog.

Evidence: Post-incident activity does not alter live state; retain the following artifacts from the incident for the lessons-learned review and future detection development: (1) the full malicious extension package (`manifest.json`, background scripts, content scripts) as a reference sample for writing future YARA rules targeting the `nativeMessaging` permission combined with suspicious cookie-access patterns; (2) the native messaging host executable recovered during eradication for static and dynamic malware analysis to identify C2 infrastructure, obfuscation techniques, and exfiltration destinations used in this specific campaign; (3) the registry export of `HKCU\Software\Google\Chrome\NativeMessagingHosts` showing the malicious host registration path, to serve as an IOC for future threat hunts across other endpoints; (4) platform sign-in logs showing the geographic and device anomalies associated with stolen cookie replay, to calibrate detection thresholds for future impossible-travel or new-device authentication alerts.

Detection Guidance

Primary signal: chrome.exe or chrome_child.dll spawning powershell.exe or cmd.exe. Query EDR for parent process = chrome.exe AND child process = powershell.exe. On Windows, correlate with Event ID 4688 with ProcessCommandLine containing -EncodedCommand, -WindowStyle Hidden, or IEX. Secondary signal: new or unsigned entries under HKCU\Software\Google\Chrome\NativeMessagingHosts in the Windows registry, especially recently created keys pointing to executables in user-writable directories. Tertiary signal: anomalous outbound HTTPS POST requests from Chrome to non-Google infrastructure immediately following extension installation, check proxy or DNS logs for domains registered within the past 30 days. Behavioral indicator: user accounts showing authenticated sessions from a new IP or device within hours of the phishing email delivery timestamp. No specific IOC hashes or domains were confirmed in source material at pipeline ingestion; hunt by behavior rather than static IOCs.

Framework Mappings

MITRE-ATTACK

- **T1176** — Software Extensions
- **T1566.001** — Spearphishing Attachment
- **T1056** — Input Capture
- **T1539** — Steal Web Session Cookie
- **T1185** — Browser Session Hijacking
- **T1059.001** — PowerShell

NIST-800-53R5

- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **CM-7** — Least Functionality

- **SI-7** — Software, Firmware, and Information Integrity
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **6.3** — Require MFA for Externally-Exposed Applications
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

SOC2-TSC

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

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1176	Software Extensions	Persistence
T1566.001	Spearphishing Attachment	Initial-Access
T1056	Input Capture	Collection
T1539	Steal Web Session Cookie	Credential-Access
T1185	Browser Session Hijacking	Collection
T1059.001	PowerShell	Execution

Sources

Source	URL	Tier
A Vulnerability in Google Chrome Could Allow for Arbitrary Code ...	https://www.cisecurity.org/advisory/a-vulnerability-in-google-chrom...	T3
Chrome Security Update: Google Fixes Another Actively Exploited ...	https://www.secpod.com/blog/chrome-security-update-google-fixes-ano...	T3
How To Fix Google Chrome's 14 New Critical Security Vulnerabilities	https://www.forbes.com/sites/daveywinder/2026/05/15/how-to-fix-goog...	T3
Google Chrome security update for 1 high severity vulnerability	https://www.youtube.com/watch?v=Mk_SN9A0er8	T3
"Known exploited" vulnerability in Chrome and Chromium. Be sure ...	https://www.reddit.com/r/linux/comments/1ls4bfr/known_exploited_vul...	T3

Generated 2026-06-27 06:09 UTC by TJS Security Command Center