INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-14 06:03 UTC

108 Coordinated Malicious Chrome Extensions Exfiltrate OAuth2 Tokens and Telegram Sessions via Shared C2 Infrastructure

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0177
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Google Chrome (extensions ecosystem), Telegram Web, Google OAuth2-dependent services, YouTube, TikTok
Published	2026-04-14T04:35:00
Discovery Source	Rss

Executive Summary

A coordinated browser supply chain campaign placed 108 malicious Chrome extensions on the Chrome Web Store under five fake developer identities, stealing Google OAuth2 tokens and Telegram session cookies from an estimated 20,000 users (per available reports; independent verification recommended). Attackers can use these stolen tokens to take over Google accounts and Telegram sessions without ever knowing the victim's password. Any organization whose employees use Chrome with third-party extensions, particularly those accessing corporate Google Workspace or Telegram, faces risk of account compromise and downstream data exposure.

Technical Analysis

Campaign overview: 108 malicious Chrome extensions operated under five fraudulent developer accounts, all routing stolen data to a single C2 server at 144.126.135[.]238 (reported in T3 sources; verification against primary threat intelligence recommended before production blocking). No CVE assigned; the attack exploits malicious extension logic rather than a browser engine vulnerability. Affected surface: Google Chrome extensions ecosystem, Google OAuth2-dependent services (including Workspace), Telegram Web, YouTube, TikTok.

Attack mechanics:

- T1176 (Browser Extensions): Extensions installed via Chrome Web Store, masquerading as legitimate utilities.

- T1056.001 (Keylogging / Credential Harvesting): Extensions harvest Google OAuth2 access tokens from browser storage and active sessions.

- T1539 (Steal Web Session Cookie): Telegram Web session cookies exfiltrated, enabling session hijacking without credentials.

- T1550.001 (Use Alternate Authentication Material, Application Access Token): Stolen OAuth2 tokens used for account takeover independent of password knowledge.

- T1185 (Browser Session Hijacking): Arbitrary JavaScript injected into victim sessions; gambling overlay content injected into active pages.

- T1564.001 (Hidden Files and Directories): Extensions strip Content-Security-Policy (CSP) response headers, disabling a primary browser-side XSS defense.

- T1071.001 (Application Layer Protocol, Web Protocols): Exfiltration routed over standard web protocols to C2 at 144.126.135[.]238.

- T1583.004 (Acquire Infrastructure, Server): Shared C2 infrastructure across all 108 extensions confirms centralized attacker control.

CWE references: CWE-359 (Exposure of Private Personal Information), CWE-79 (Improper Neutralization of Input During Web Page Generation, XSS), CWE-284 (Improper Access Control).

Attribution: Unknown. Russian-language comments observed in extension source code; no confirmed actor attribution.

Patch status: No vendor patch applicable, mitigation requires extension removal and token revocation. Google Chrome Web Store removal status for these extensions was not confirmed in available source material at the time of this report.

Source quality note: Primary sourcing is The Hacker News (T3 tier). Campaign details, IOCs, and user impact figures require verification against a T1/T2 threat intelligence source (Google Security Blog, CISA advisory, or published threat report) before high-confidence operational deployment.

Action Checklist

1. Step 1: Containment, Immediately audit installed Chrome extensions across managed endpoints using Chrome Enterprise policy (chrome://policy or Google Admin Console > Devices > Chrome > Apps & Extensions). Cross-reference installed extension IDs against any published blocklist from this campaign. Block outbound connections to 144.126.135[.]238 (pending verification from primary threat intelligence source; apply to test environment first, then production if confirmed via independent detection or threat intelligence partner) at perimeter firewall and DNS sinkhole. Suspend corporate Google accounts showing anomalous OAuth2 token activity pending investigation.
2. Step 2: Detection, Query endpoint telemetry (EDR, Chrome Enterprise reporting) for extensions installed from unknown or unreviewed developer accounts. Search proxy and firewall logs for outbound connections to 144.126.135[.]238 (cross-reference against authoritative threat intelligence to avoid false positives). Review Google Workspace Admin > Security > OAuth token grants for unfamiliar third-party application authorizations. In Telegram, audit active sessions via Settings > Devices for unrecognized sessions. Alert on CSP header stripping behavior in web proxy logs if your proxy performs SSL inspection.
3. Step 3: Eradication, Remove all identified malicious extensions from affected endpoints via Chrome Enterprise forced uninstall policy or manual removal. Revoke all Google OAuth2 tokens for affected users

via Google Admin Console > Security > Manage OAuth clients, or instruct users to revoke at myaccount.google.com/permissions. Terminate all active Telegram sessions except verified devices via Telegram Settings > Devices > Terminate all other sessions. Enforce an extension allowlist policy in Chrome Enterprise to block unapproved extensions going forward.

4. Step 4: Recovery, After token revocation, require affected users to reauthenticate to all Google services and re-establish Telegram sessions on verified devices. Verify no unauthorized OAuth2 applications retain access in Google Admin Console. Monitor affected accounts for 30 days for anomalous login activity, data access, or forwarding rules (particularly Gmail). Confirm C2 IP block is active and no further outbound connections to 144.126.135[.]238 appear in logs.

5. Step 5: Post-Incident, Conduct a gap assessment against CIS Benchmark for Google Chrome (CIS Google Chrome Benchmark) and NIST SP 800-53 SC-18 (Mobile Code) controls. Implement a formal Chrome extension governance policy: periodic reviews, approved publisher lists, and automated alerting for new extension installations on managed devices. Evaluate whether Google Workspace Conditional Access and OAuth app whitelisting controls are enabled. Brief employees on the risk of installing unvetted browser extensions from the Chrome Web Store.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to legal, privacy counsel, and executive leadership immediately if forensic evidence confirms that stolen Google OAuth2 tokens were used to access Google Workspace data (Drive, Gmail, Docs) containing PII, PHI, or financial records, as this may trigger breach notification obligations under GDPR, CCPA, or HIPAA within 72-hour reporting windows.
Recovery Notes	After revoking OAuth2 tokens and terminating Telegram sessions, verify recovery completeness by re-auditing Google Admin Console token grants 24 hours post-revocation to confirm no re-authorization has occurred under the malicious extension IDs — affected users may have reinstalled extensions if Chrome sync restored them. Monitor affected Google Workspace accounts for 30 days specifically for Gmail forwarding rule creation, Google Drive external sharing events, and OAuth re-grants to any application not on the approved whitelist, as attackers may have created persistence mechanisms during the token access window. Enforce Google Workspace Context-Aware Access policies requiring managed, compliant devices for OAuth2 authorization before returning affected accounts to full production use.

Forensic Artifacts	Chrome Extension Activity SQLite database at '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extension Activity' — records every chrome.cookies.getAll(), chrome.identity.getAuthToken(), and outbound XMLHttpRequest API call made by each extension, directly evidencing OAuth2 token and Telegram cookie harvesting behavior | Chrome 'Preferences' and 'Secure Preferences' JSON files at '%LOCALAPPDATA%\Google\Chrome\User Data\Default\' — contain extension installation timestamps, enabled/disabled state, and content script injection records linking the malicious extension to specific browsing sessions on Google and Telegram Web | Perimeter firewall and proxy session logs for outbound HTTPS connections to 144.126.135[.]238 — capture source endpoint IP, bytes uploaded (exfiltration volume), timestamps, and HTTP method (POST) confirming token exfiltration events per affected user | Google Workspace Admin Token Audit report (Admin Console > Reports > Audit > Token) — records exact OAuth2 grant events including application name, granted scopes, and authorizing user, establishing which accounts granted access and what data scopes the malicious application obtained | Malicious extension background.js and content.js source files preserved from '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\\\' — contain obfuscated exfiltration logic, hardcoded C2 IP 144.126.135[.]238, and targeted cookie domain patterns (accounts.google.com, web.telegram.org) that serve as definitive indicators of compromise for YARA rule development

Per-Action IR Details

Step 1: Containment — Immediately audit installed Chrome extensions across managed endpoints using Chrome Enterprise policy (chrome://policy or Google Admin Console > Devices > Chrome > Apps & Extensions). Cross-reference installed extension IDs against any published blocklist from this campaign. Block outbound connections to 144.126.135[.]238 at perimeter firewall and DNS sinkhole. Suspend corporate Google accounts showing anomalous OAuth2 token activity pending investigation.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST AC-3 (Access Enforcement), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 2.3 (Address Unauthorized Software)

Compensating: For teams without Chrome Enterprise: distribute a PowerShell script to enumerate installed extensions on all managed Windows endpoints — 'Get-ItemProperty HKLM:\SOFTWARE\Google\Chrome\Extensions' and 'Get-ChildItem "$env:LOCALAPPDATA\Google\Chrome\User Data\Default\Extensions"' — and compare extension directory names against the campaign's known malicious extension IDs. For the network block, apply an iptables rule on a perimeter Linux host ('iptables -A FORWARD -d 144.126.135.238 -j DROP') or add the IP to a Pi-hole blocklist as an interim DNS sinkhole. Export Google account OAuth token grants manually via myaccount.google.com/permissions for each suspected user if Admin Console access is unavailable.

Evidence: Before blocking the C2 IP, capture full firewall session logs showing all source endpoints that have already communicated with 144.126.135[.]238, including bytes transferred and session duration — this identifies which users successfully exfiltrated tokens. Preserve Chrome extension manifest files ('manifest.json') from affected endpoints at '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\\\' before removal; these contain the declared permissions (notably 'cookies', 'identity', 'webRequest', 'tabs') confirming malicious capability. Capture the Chrome Local State file at '%LOCALAPPDATA%\Google\Chrome\User Data\Local State' which records installed extension metadata and timestamps.

Step 2: Detection — Query endpoint telemetry (EDR, Chrome Enterprise reporting) for extensions installed from unknown or unreviewed developer accounts. Search proxy and firewall logs for outbound connections to 144.126.135[.]238. Review Google Workspace Admin > Security > OAuth token grants for unfamiliar third-party application authorizations. In Telegram, audit active sessions via Settings > Devices for unrecognized sessions. Alert on CSP header stripping behavior in web proxy logs if your proxy performs SSL

inspection.

NIST Phase: Detection Analysis

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

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without EDR, deploy osquery with a query targeting Chrome extension directories: 'SELECT name, identifier, version, permissions FROM chrome_extensions WHERE identifier IN ("","")' run against all managed endpoints via osquery fleet. For proxy log analysis without a SIEM, run 'grep "144.126.135.238" /var/log/squid/access.log | awk "{print $3, $7, $8}"' (Squid) or equivalent to surface source IPs and request timestamps. For CSP stripping detection without SSL inspection, use Wireshark with display filter 'http.response.code == 200 && !(http.response_for.uri contains "144.126.135.238")' on mirrored traffic to look for responses lacking Content-Security-Policy headers on OAuth-serving Google domains. Export Google Workspace Admin SDK token audit report via 'gam report token' (free GAM CLI tool) to enumerate all third-party OAuth grants without logging into Admin Console.

Evidence: Collect Chrome browser history and extension event logs from '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extension Activity' — this SQLite database records API calls made by each extension, including 'chrome.cookies.getAll()', 'chrome.identity.getAuthToken()', and 'XMLHttpRequest' calls to 144.126.135[.]238, directly proving exfiltration behavior. Pull Google Workspace Admin audit logs (Admin Console > Reports > Audit > Token) filtered for OAuth2 grant events within the installation window of the malicious extensions, noting any grants to application IDs not matching approved enterprise apps. For Telegram, document the session list screenshot or export from Settings > Devices before termination — active unrecognized sessions are forensic evidence of session cookie hijack.

Step 3: Eradication — Remove all identified malicious extensions from affected endpoints via Chrome Enterprise forced uninstall policy or manual removal. Revoke all Google OAuth2 tokens for affected users via Google Admin Console > Security > Manage OAuth clients, or instruct users to revoke at myaccount.google.com/permissions. Terminate all active Telegram sessions except verified devices via Telegram Settings > Devices > Terminate all other sessions. Enforce an extension allowlist policy in Chrome Enterprise to block unapproved extensions going forward.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), NIST AC-2 (Account Management), CIS 2.3 (Address Unauthorized Software), CIS 5.3 (Disable Dormant Accounts)

Compensating: Without Chrome Enterprise policy enforcement, distribute a PowerShell remediation script that deletes the malicious extension directories from '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\' and removes corresponding registry entries under 'HKCU:\SOFTWARE\Google\Chrome\Extensions', then forces a Chrome restart. For OAuth revocation without Admin Console access, use the Google OAuth2 token revocation endpoint directly: 'curl -X POST https://oauth2.googleapis.com/revoke?token=' for each affected token identified during detection. Enforce extension allowlisting without enterprise licensing by pushing a managed 'chrome.json' policy file via Group Policy Object (GPO) setting 'ExtensionInstallBlocklist' to '*' and 'ExtensionInstallAllowlist' to approved IDs only.

Evidence: Before running forced uninstall policy, image or copy the full extension directory for each malicious extension from '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\\' — preserve background scripts (background.js, content.js) which contain the obfuscated exfiltration code targeting 'chrome.cookies', 'chrome.identity.getAuthToken', and HTTP POST to 144.126.135[.]238. Capture the Chrome 'Preferences' JSON file at '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Preferences' before remediation — it records extension installation timestamps and enabled state. Document Google Admin Console token grant records for affected accounts before revocation as evidence of scope.

Step 4: Recovery — After token revocation, require affected users to reauthenticate to all Google services and re-establish Telegram sessions on verified devices. Verify no unauthorized OAuth2 applications retain access

in Google Admin Console. Monitor affected accounts for 30 days for anomalous login activity, data access, or forwarding rules (particularly Gmail). Confirm C2 IP block is active and no further outbound connections to 144.126.135[.]238 appear in logs.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), CIS 6.2 (Establish an Access Revoking Process), CIS 6.3 (Require MFA for Externally-Exposed Applications)

Compensating: Without a SIEM for 30-day monitoring, configure Google Workspace Alert Center (free with Workspace) to send email alerts for 'Suspicious login activity', 'Government-backed attack warning', and 'User granted Admin privilege' events on affected accounts. For Gmail forwarding rule monitoring without automation, run a weekly manual audit using 'gam user show filters' (free GAM CLI) to detect any forwarding rules created during or after the compromise window. Verify the C2 block persists by running a daily cron job: 'curl --max-time 5 -s -o /dev/null -w "%{http_code}" http://144.126.135.238' from a sacrificial host — a timeout or connection refused confirms the block is active.

Evidence: Before closing the incident, pull a full Gmail filter and forwarding rule export for each compromised account (Google Takeout or GAM CLI) — attackers with temporary OAuth2 access commonly create persistent forwarding rules to maintain intelligence access after token revocation. Review Google Workspace login audit logs (Admin Console > Reports > Audit > Login) for the 48-hour window following initial extension installation to identify any access from anomalous geolocations or user agents consistent with attacker-controlled infrastructure using the stolen tokens. Check Google Drive activity logs for bulk file access or sharing changes made under the compromised OAuth2 session.

Step 5: Post-Incident — Conduct a gap assessment against CIS Benchmark for Google Chrome (CIS Google Chrome Benchmark) and NIST SP 800-53 SC-18 (Mobile Code) controls. Implement a formal Chrome extension governance policy: periodic reviews, approved publisher lists, and automated alerting for new extension installations on managed devices. Evaluate whether Google Workspace Conditional Access and OAuth app whitelisting controls are enabled. Brief employees on the risk of installing unvetted browser extensions from the Chrome Web Store.

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 SC-18 (Mobile Code), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-7 (Least Functionality), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: For teams without enterprise extension management tooling, implement a free Sigma rule in any log pipeline targeting Chrome extension installation events derived from Windows Event Log or Sysmon Event ID 11 (FileCreate) monitoring the '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\' path for new subdirectories — new extension IDs not on the approved list trigger an alert. Publish an internal approved extension list as a static JSON file and schedule a weekly PowerShell script across endpoints that compares installed extension IDs against the approved list and emails a discrepancy report. For OAuth app governance without Workspace Enterprise tier, use the free Google Workspace Alert Center and configure periodic manual reviews of the token audit report exported via GAM 'gam report token' monthly.

Evidence: Document the full list of the 108 malicious extension IDs and the five fake developer account identities for inclusion in the lessons-learned report and internal threat intelligence feed — these serve as permanent IOCs for future detection rules. Preserve a copy of the Chrome Web Store pages (via web archive or screenshot) for each malicious extension before Google removes them, as they may be needed for insurance claims, regulatory notifications, or law enforcement referrals. Record the timeline from extension installation to C2 callback to token exfiltration (derived from Chrome Extension Activity DB and firewall logs) to quantify dwell time and improve detection SLAs in the updated IR plan.

Detection Guidance

IOC: Outbound network connections to 144.126.135[.]238 (TCP, likely port 443 or 80). Primary IOC sourced from T3-tier reporting; cross-reference against authoritative threat intelligence before broad-scale production blocking to avoid false positives. Query firewall, proxy, and SIEM logs for this IP. DNS queries resolving to this IP should also be flagged.

Behavioral indicators:

- CSP (Content-Security-Policy) headers missing or stripped on responses observed through SSL-inspecting proxy, compare baseline header presence before and after extension installation.

- Unexpected JavaScript execution or overlay content injected into known-clean web pages, particularly Google or Telegram Web sessions.

- New or unrecognized OAuth2 application authorizations appearing in Google Admin Console > Security > API controls > App access control.

- Active Telegram sessions appearing from IP addresses or device types inconsistent with user behavior.

- Chrome extension installations from developer accounts not on an approved list, particularly accounts with very few total published extensions or recent account creation dates.

Log sources to query: Chrome Enterprise reporting (extension inventory), endpoint EDR (browser process network connections), perimeter firewall/proxy (outbound to 144.126.135[.]238), Google Workspace Admin audit logs (OAuth grants, login events), Telegram session audit (user-level).

Note: Specific extension IDs and file hashes were not confirmed in available T3-tier source material. Obtain a verified IOC list from a primary threat intelligence source before signature-based detection deployment.

Indicators of Compromise

Type	Value	Context	Confidence
IP	`144.126.135[.]238`	Attacker-controlled C2 server shared across all 108 malicious Chrome extensions; destination for exfiltrated OAuth2 tokens and Telegram session cookies	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1056.001** — Keylogging
- **T1564.001** — Hidden Files and Directories
- **T1539** — Steal Web Session Cookie
- **T1071.001** — Web Protocols
- **T1583.004** — Server
- **T1550.001** — Application Access Token

- **T1176** — Software Extensions
- **T1185** — Browser Session Hijacking

OWASP-TOP10-2021

- **A03:2021** — Injection
- **A01:2021** — Broken Access Control

NIST-800-53R5

- **SI-10** — Information Input Validation
- **AC-3** — Access Enforcement
- **SR-2** — Supply Chain Risk Management Plan

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **6.3** — Require MFA for Externally-Exposed Applications
- **15.1** — Establish and Maintain an Inventory of Service Providers

ISO-27001-2022

- **A.8.28** — Secure coding
- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1056.001	Keylogging	Collection
T1564.001	Hidden Files and Directories	Defense-Evasion
T1539	Steal Web Session Cookie	Credential-Access

Technique ID	Technique Name	Tactic
T1071.001	Web Protocols	Command-And-Control
T1583.004	Server	Resource-Development
T1550.001	Application Access Token	Defense-Evasion
T1176	Software Extensions	Persistence
T1185	Browser Session Hijacking	Collection

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/04/108-malicious-chrome-extensions-s...	T3
Zero-click vulnerability afflicts Telegram, allows full device takeover ...	https://cybernews.com/security/telegram-zero-click-vulnerability-an...	T3
Secure Chrome Passwords: Stop AutoFill Vulnerabilities - TikTok	https://www.tiktok.com/@techinherstep/video/7626503348031720735	T3
Here Come the AI Browsers - Scareware Blockers - YouTube	https://www.youtube.com/watch?v=mDGSQLNB4q4	T3
Claude vs. ChatGPT: Chrome Extension Vulnerability Found - TikTok	https://www.tiktok.com/@infosecant/video/7621877434551586051	T3

Generated 2026-04-14 06:03 UTC by TJS Security Command Center