INTELLIGENCE BRIEFING
Security Command Center

TLP:CLEAR
2026-06-25 18:50 UTC

Adblock for YouTube Chrome Extension (10M+ Installs) Contains Dormant Remote JavaScript Injection Capability

SECURITY ANALYSIS | CRITICAL | CVSS 7.5

SCC Item ID	SCC-STY-2026-0274
Type	Security Analysis
Severity	CRITICAL
CVSS Base Score	7.5
Affected Products	Google Chrome (all versions with extension installed); Adblock for YouTube extension (Chrome Extension ID: cmedhionkhpnakcndndgjdbohmhepckk); Chrome Web Store
Published	2026-06-25T10:12:52
Discovery Source	Rss

Executive Summary

A Chrome extension marketed as an ad blocker for YouTube, with over 10 million active installs, contains a hidden mechanism that allows its developers to push arbitrary JavaScript into any browser tab without issuing an update or triggering Chrome Web Store review. The capability has been dormant since at least February 2025 and is linked to a cluster of extensions already removed for malicious behavior, meaning the infrastructure and intent to weaponize it are established. This incident exposes a structural blind spot in enterprise browser security: trusted, widely installed extensions can serve as pre-positioned implants, activatable at the developer's discretion across the full install base simultaneously.

Technical Analysis

The Island.io 'BadBlocker' research documents a remote JavaScript injection architecture embedded in the Adblock for YouTube extension (Chrome Extension ID: cmedhionkhpnakcndndgjdbohmhepckk). The mechanism does not rely on a traditional software update pathway. Instead, the extension calls a remote server at runtime and receives executable JavaScript, which it injects into browser sessions. This design bypasses Chrome Web Store code review entirely, because the malicious payload never resides in the submitted extension package. The dormant capability maps directly to several MITRE ATT&CK techniques: T1059.007 (Command and Scripting Interpreter: JavaScript), T1539 (Steal Web Session Cookie), T1071.001 (Application Layer Protocol: Web Protocols for C2 communication), T1185 (Browser Session Hijacking), T1176 (Browser Extensions as an initial access or persistence vector), and T1657 (Financial Theft, representing one plausible

monetization path).

The CWE profile reflects the layered design of the threat. CWE-95 (Improper Neutralization of Directives in Dynamically Evaluated Code) captures the injection surface. CWE-116 (Improper Encoding/Escaping of Output) and CWE-602 (Client-Side Enforcement of Server-Side Security) describe the extension's reliance on a remotely controlled server as its trust boundary, with no client-side validation of injected content. CWE-693 (Protection Mechanism Failure) and CWE-284 (Improper Access Control) reflect the broader failure: the Chrome extension permission model was not designed to prevent this class of behavior, because the extension legitimately requests broad page-access permissions to perform its stated function.

The extension's linkage to other already-removed Chrome Web Store extensions indicates this is not an isolated developer error. It is consistent with a coordinated supply-chain positioning strategy: publish a high-utility, high-install-count extension, embed a remote activation capability, and wait. The Island.io report phrases the risk precisely: 'one server call away from compromise.' If activated, the threat actor could execute session hijacking, in-browser credential harvesting, and data exfiltration across millions of sessions simultaneously, without any user interaction, without a new install event, and without a detectable update to the extension itself.

The extension remained available in the Chrome Web Store as of the discovery reporting date. No CVE has been assigned. The CVSS base of 7.5 reflects remote code execution with high confidentiality and integrity impact; qualitative severity is assessed as Critical given the pre-positioned install base, zero-friction activation path, and demonstrated malicious intent by the associated extension cluster.

Action Checklist

1. Step 1: Assess exposure, inventory browser extensions across managed endpoints immediately; query specifically for Chrome Extension ID cmedhionkhpnakcndndgjdbohmhepckk (Adblock for YouTube). Use endpoint management tooling, browser management policies, or EDR extension telemetry. Unmanaged BYOD devices and personal profiles on corporate browsers are a secondary but real exposure surface.
2. Step 2: Remove or block the extension, push a Chrome Group Policy or Chrome Browser Cloud Management policy to blocklist Extension ID cmedhionkhpnakcndndgjdbohmhepckk. Do not wait for a Chrome Web Store removal; the store has not removed it as of the reporting date. Reference CIS Safeguard 2.3 (Address Unauthorized Software) and CIS Safeguard 2.1 (Establish and Maintain a Software Inventory) for process framing.
3. Step 3: Review extension governance controls, audit your Chrome extension allow-list policy. If no policy exists, treat this incident as the trigger to implement one. Apply NIST AC-3 (Access Enforcement) and AC-6 (Least Privilege) principles: only extensions with a documented business need and verified publisher should be permitted on managed devices.
4. Step 4: Audit remote-call telemetry, hunt for outbound connections from Chrome browser processes to unknown or uncategorized domains during active browsing sessions. Extensions performing runtime server calls for executable content represent a behavioral anomaly distinct from CDN or ad-network traffic. Reference NIST AU-6 (Audit Record Review, Analysis, and Reporting) and NIST SI-4 (System Monitoring) for detecting anomalous outbound connections. Cross-reference D3FEND countermeasures for session-context anomaly detection and proxy-based web server access mediation for intercepting and inspecting extension-initiated outbound calls.
5. Step 5: Update threat model, add browser extension supply-chain positioning as an explicit attack vector in your threat register. Map to T1176 (Browser Extensions) and T1185 (Browser Session Hijacking). Note that this class of threat bypasses perimeter controls, endpoint AV, and store-side code review

simultaneously.

6. Step 6: Communicate findings, brief leadership on the specific risk: a widely trusted productivity tool with 10 million installs contained a hidden, remotely activatable compromise capability. Frame the business risk as session hijacking and credential theft across any employee who installed the extension on a work browser profile.

7. Step 7: Monitor developments, track for a Chrome Web Store removal notice, a Google security advisory, Island.io follow-up disclosures, and any CVE assignment. If the extension is removed without a public post-mortem, treat the associated extension cluster as an ongoing threat and review for related Extension IDs referenced in the Island.io BadBlocker report.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal/privacy counsel immediately if Step 4 telemetry confirms any outbound connection from Chrome processes to the extension's remote JavaScript delivery infrastructure, indicating potential activation and session or credential exposure requiring breach notification assessment under GDPR, CCPA, or applicable state law.
Recovery Notes	After the blocklist policy is confirmed deployed and enforced across all managed endpoints, verify removal by querying `chrome://extensions` state via endpoint management tooling and confirming Extension ID cmedhionkhpnakcndndgjdbohmhepckk is absent from all browser profiles. For any endpoint where Step 4 identified outbound calls to non-Google infrastructure attributable to the extension, treat authenticated sessions active during the installation window as potentially compromised and initiate forced re-authentication for associated SaaS and internal applications. Continue monitoring Chrome process network telemetry and DNS logs for the related extension cluster IDs from the Island.io BadBlocker report for a minimum of 30 days post-containment, as the established C2 infrastructure may pivot to a surviving extension in the cluster.

Forensic Artifacts	Chrome extension directory for cmedhionkhpnakcndndgjdbohmhepckk on disk — Windows: `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\cmedhionkhpnakcndndgjdbohmhepckk\`; macOS: `~/Library/Application Support/Google/Chrome/Default/Extensions/cmedhionkhpnakcndndgjdbohmhepckk/` — containing `manifest.json`, `background.js`, and any bundled scripts that reveal the remote fetch mechanism and target domains for JavaScript delivery | Chrome HTTP cache entries at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Cache` (Windows) filterable for MIME type `application/javascript` or `text/javascript` from non-Google, non-YouTube origin domains, which would represent remotely injected script payloads delivered to the browser prior to containment | Sysmon Event ID 3 (Network Connection Detected) and Event ID 22 (DNS Query) logs filtered on `chrome.exe` as the initiating process, covering the period from February 2025 (earliest known dormant capability date per advisory) to present, to identify C2 domain contacts made by the extension's service worker or background page | Chrome `History` SQLite database at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\History` — query the `urls` and `visits` tables for entries made by extension service workers rather than direct user navigation, and the `downloads` table for any script or payload fetched during extension execution | Chrome `Local Storage` and `IndexedDB` stores at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Local Storage\leveldb\` and `%LOCALAPPDATA%\Google\Chrome\User Data\Default\IndexedDB\` scoped to the extension's origin (`chrome-extension://cmedhionkhpnakcndndgjdbohmhepckk`), which may contain staged JavaScript payloads, exfiltrated session tokens, or C2 configuration persisted between browser sessions

Per-Action IR Details

Step 1: Assess exposure — inventory browser extensions across managed endpoints immediately; query specifically for Chrome Extension ID cmedhionkhpnakcndndgjdbohmhepckk (Adblock for YouTube). Use endpoint management tooling, browser management policies, or EDR extension telemetry. Unmanaged BYOD devices and personal profiles on corporate browsers are a secondary but real exposure surface.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection & Analysis: scoping adverse event to identify affected population before containment actions are taken

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: On Windows endpoints without EDR, run: `reg query 'HKLM\SOFTWARE\Google\Chrome\Extensions' /s | findstr cmedhionkhpnakcndndgjdbohmhepckk` and `reg query 'HKCU\SOFTWARE\Google\Chrome\Extensions' /s | findstr cmedhionkhpnakcndndgjdbohmhepckk` across hosts via PSExec or a simple PowerShell remoting loop. On macOS, run: `find ~/Library/Application\ Support/Google/Chrome -name 'manifest.json' | xargs grep -l cmedhionkhpnakcndndgjdbohmhepckk`. For BYOD triage, distribute a one-liner self-assessment script to employees via email with instructions to report presence.

Evidence: This step is read-only inventory and does not alter live state. However, before any subsequent action, capture: the full list of installed extension IDs and versions from each affected browser profile (`%LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\` on Windows; `~/Library/Application Support/Google/Chrome/Default/Extensions/` on macOS); the extension's `manifest.json` and `background.js` files from disk to preserve the version present at discovery; and any cached remote resources fetched by the extension stored in Chrome's cache directory (`%LOCALAPPDATA%\Google\Chrome\User Data\Default\Cache`).

Step 2: Remove or block the extension — push a Chrome Group Policy or Chrome Browser Cloud Management policy to blocklist Extension ID cmedhionkhpnakcndndgjdbohmhepckk. Do not wait for a Chrome Web Store removal; the store has not removed it as of the reporting date. Reference CIS Safeguard

2.3 (Address Unauthorized Software) and CIS Safeguard 2.1 (Establish and Maintain a Software Inventory) for process framing.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: blocking further execution of the dormant remote JavaScript injection capability before it is activated by the extension's C2 infrastructure

Controls: CIS 2.3 (Address Unauthorized Software), CIS 2.1 (Establish and Maintain a Software Inventory), NIST AC-3 (Access Enforcement)

Compensating: Without centralized Chrome policy management, deploy a Windows GPO `ExtensionInstallBlocklist` registry key under `HKLM\SOFTWARE\Policies\Google\Chrome\ExtensionInstallBlocklist\1 = cmedhionkhpnakcndndgjdbohmhepckk` via Group Policy or a PowerShell push script. On macOS without MDM, distribute a `.plist` configuration profile to `~/Library/Managed Preferences/com.google.Chrome.plist` setting `ExtensionInstallBlocklist`. As an immediate manual fallback, instruct users to navigate to `chrome://extensions`, locate 'Adblock for YouTube', and click Remove — document confirmation per user.

Evidence: BEFORE pushing the blocklist policy or instructing manual removal, capture: a full memory dump or at minimum `Get-Process chrome` output showing all active Chrome renderer and extension service worker PIDs; active network connections from Chrome processes via `netstat -ano | findstr ` or `Get-NetTCPConnection -OwningProcess ` to identify any live outbound call to the extension's remote JavaScript delivery infrastructure; Chrome's `Network` log from `chrome://net-export/` if still active, preserving any in-flight or recent requests made by extension ID cmedhionkhpnakcndndgjdbohmhepckk; and the extension's IndexedDB and localStorage contents at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\IndexedDB` which may contain tokens, session state, or injected script fragments staged for execution.

Step 3: Review extension governance controls — audit your Chrome extension allow-list policy. If no policy exists, treat this incident as the trigger to implement one. Apply NIST AC-3 (Access Enforcement) and AC-6 (Least Privilege) principles: only extensions with a documented business need and verified publisher should be permitted on managed devices.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: updating preventive controls and policies based on lessons learned from this extension supply-chain compromise

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Without a Chrome Browser Cloud Management subscription, implement extension governance via Windows Group Policy using the Chrome ADMX templates (free from Google). Configure `ExtensionInstallAllowlist` with only explicitly approved extension IDs, combined with `ExtensionInstallBlocklist = *` as a wildcard deny-all baseline. Maintain the approved list in a version-controlled text file (Git repository or shared drive) reviewed quarterly. Document each approved extension with publisher, business justification, and last-reviewed date.

Evidence: This is a policy review step that does not alter live system state. No volatile capture is required before execution. Document the current policy state as baseline evidence: export existing GPO settings via `Get-GPOReport` or screenshot `chrome://policy` on a representative managed endpoint before changes are applied, preserving the pre-remediation configuration for post-incident review and audit trail.

Step 4: Audit remote-call telemetry — hunt for outbound connections from Chrome browser processes to unknown or uncategorized domains during active browsing sessions. Extensions performing runtime server calls for executable content represent a behavioral anomaly distinct from CDN or ad-network traffic. Reference NIST AU-6 (Audit Record Review, Analysis, and Reporting) and NIST SI-4 (no mapped control — SI-4 is not in the verified knowledge base control set; omit citation). Cross-reference D3FEND countermeasure D3-LAM (Local Account Monitoring) for session-context anomaly detection and D3-PBWSAM (Proxy-based Web Server Access Mediation) for intercepting and inspecting extension-initiated outbound calls.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection & Analysis: hunting for evidence that the dormant remote JavaScript injection capability was activated prior to containment, establishing whether this is a precursor or an active compromise

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM or proxy with SSL inspection, run Wireshark or `tshark` on a representative endpoint during an active Chrome session with the extension loaded: `tshark -i -f 'tcp port 443' -w extension_capture.pcap`. Separately, enable Chrome's built-in net-log via `chrome://net-export/` to capture extension-initiated requests including HTTPS hostnames. Parse DNS query logs from your local DNS resolver (Windows DNS debug log or Pi-hole query log) filtering for domains resolved by the Chrome process. Use Sysmon Event ID 22 (DNS Query) filtered on `chrome.exe` as the initiating process to correlate domains contacted specifically during extension activity windows.

Evidence: BEFORE concluding this hunting step, preserve as forensic evidence: DNS resolution history from Windows DNS Client cache via `ipconfig /displaydns > dns_cache__.txt`; Sysmon Event ID 3 (Network Connection) logs filtered on `chrome.exe` to identify outbound calls from the browser process to non-Google, non-YouTube infrastructure during the extension's installation period (from February 2025 onward per the advisory); Chrome browsing history and extension activity log at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\History` (SQLite) queryable for URLs fetched by the extension's service worker; and any JavaScript files retrieved remotely and cached, identifiable in the Chrome cache by examining cache entries for MIME type `application/javascript` or `text/javascript` originating from non-Google domains.

Step 5: Update threat model — add browser extension supply-chain positioning as an explicit attack vector in your threat register. Map to T1176 (Browser Extensions) and T1185 (Browser Session Hijacking). Note that this class of threat bypasses perimeter controls, endpoint AV, and store-side code review simultaneously.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: incorporating lessons learned from this specific extension cluster compromise into threat modeling and detection engineering to improve future identification

Compensating: Without a formal threat modeling tool, document the updated threat register entry in a structured markdown or spreadsheet format: record the attack vector (Chrome extension supply chain), the bypass mechanism (remote JavaScript injection post-install, bypassing Web Store review), affected asset class (all managed endpoints with Chrome and unvetted extensions), and link to the Island.io BadBlocker report as the threat intelligence source. Schedule a quarterly review of installed extensions against the CRXcavator (crxcavator.io) or Spin.AI risk scoring database as a free vetting mechanism.

Evidence: No live system state is altered by this step. As supporting documentation for the threat model update, preserve: the original Island.io disclosure report and any associated IOC lists referencing the broader extension cluster linked to cmedhionkhpnakcndndgjdbohmhepckk; the list of related Extension IDs identified in the BadBlocker report that were previously removed for malicious behavior, which establishes the threat actor's infrastructure pattern; and internal telemetry exports from Steps 1 and 4 showing the blast radius (number of affected endpoints, duration of installation) to inform risk-rating the updated threat register entry.

Step 6: Communicate findings — brief leadership on the specific risk: a widely trusted productivity tool with 10 million installs contained a hidden, remotely activatable compromise capability. Frame the business risk as session hijacking and credential theft across any employee who installed the extension on a work browser profile.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: internal communications to decision-makers are part of coordinated containment, ensuring authorization for blocklist deployment and awareness of residual risk during the active containment window

Controls: NIST AC-1 (Policy And Procedures)

Compensating: For teams without a formal incident communications template, use a structured three-section brief: (1) What happened — the Adblock for YouTube extension (ID: cmedhionkhpnakcndndgjdbohmhepckk) contained a hidden capability to push arbitrary JavaScript into any open browser tab, active since at least February 2025; (2) What is at risk — authenticated sessions, credentials, and sensitive data visible in any browser tab on affected profiles; (3)

What we did — blocklist deployed, X endpoints confirmed clear, Y endpoints pending, monitoring active. Deliver via email with a read receipt to create an auditable notification record.

Evidence: No live system state is altered by this communication step. Include in the leadership brief the artifact inventory from Step 1 (affected endpoint count and user list) and the telemetry findings from Step 4 (whether any outbound calls to the remote JavaScript delivery infrastructure were observed) so leadership can accurately assess whether this is a confirmed active compromise or a contained precursor. If any evidence of activation was found in Step 4, escalate the brief to include potential breach notification requirements under applicable data protection regulations.

Step 7: Monitor developments — track for a Chrome Web Store removal notice, a Google security advisory, Island.io follow-up disclosures, and any CVE assignment. If the extension is removed without a public post-mortem, treat the associated extension cluster as an ongoing threat and review for related Extension IDs referenced in the Island.io BadBlocker report.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: sustained monitoring and threat intelligence tracking following initial containment, consistent with the advisory's finding that the remote activation infrastructure is established and the cluster has a history of weaponization

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

Compensating: Without a commercial threat intelligence feed, configure free monitoring using: Google Alerts for 'Adblock for YouTube extension' and 'cmedhionkhpnakcndndgjdbohmhepckk'; an RSS feed subscription to the CISA Known Exploited Vulnerabilities catalog and NVD for any CVE assignment referencing this extension or the broader cluster; and a recurring weekly manual check of the Chrome Web Store listing for extension ID cmedhionkhpnakcndndgjdbohmhepckk to detect removal or policy changes. Maintain a watchlist of the related Extension IDs from the Island.io BadBlocker report and query your endpoint inventory (Step 1 methodology) for any of those IDs on a biweekly basis.

Evidence: This is an ongoing monitoring step. Preserve a timestamped snapshot of the Chrome Web Store listing for cmedhionkhpnakcndndgjdbohmhepckk at the time of initial discovery (screenshot or archived HTML) as a baseline for detecting future changes, including silent removal or version updates that could indicate the operator attempting to cleanse or re-arm the extension. Retain all telemetry collected in Steps 1 and 4 for a minimum of 90 days to support retrospective analysis if the extension cluster is subsequently attributed to a threat actor or if activation evidence surfaces in the environment.

Detection Guidance

Detection for this threat operates across three layers: extension inventory, network telemetry, and browser session behavior.

Extension inventory: Query endpoint management or EDR for the presence of Chrome Extension ID cmedhionkhpnakcndndgjdbohmhepckk across all managed devices. Chrome Browser Cloud Management, Microsoft Intune (for Chromium-based browsers), or EDR tools with browser extension visibility can surface this. Flag any device where the extension is installed and active. Reference CIS 8.2 (Collect Audit Logs) for ensuring extension installation events are captured in audit logs.

Network telemetry: The injection mechanism operates via a runtime server call from the extension to a remote backend. Hunt for outbound HTTP/S connections originating from Chrome renderer processes to domains not associated with Google, YouTube, or established CDNs. Proxy logs and DNS query logs are the primary sources. Apply D3FEND proxy-based web server access mediation to inspect and log extension-initiated requests. Look specifically for connections that return JavaScript content-type responses to a browser extension context outside of normal page navigation.

Browser session anomalies: If the payload were activated, behavioral indicators would include: unexpected DOM manipulation events, outbound POST requests containing session cookie data or form field content, and connections to unfamiliar domains immediately following page load events. Web proxy or CASB solutions with SSL inspection enabled are best positioned to detect this. Reference D3FEND system file analysis for monitoring browser profile data and session storage for unexpected modifications.

Post-activation forensics: If compromise is suspected, examine browser local storage, session cookies, and saved credentials for signs of exfiltration. Apply NIST AU-11 (Audit Record Retention) to ensure sufficient log history exists to reconstruct the session timeline. The Island.io BadBlocker report (island.io/blog/badblocker-11-million-users-one-server-call-away-from-compromise) should be reviewed for any published indicators of compromise associated with the backend infrastructure.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`Chrome Extension ID: cmedhionkhpnakcndndgjdbohmhepckk`	Adblock for YouTube extension leveraged via Chrome's legitimate extension permission model to perform remote JavaScript injection into browser sessions via a runtime server call, enabling potential session hijacking and credential theft	HIGH
URL	`Pending — refer to Island.io BadBlocker report (island.io/blog/badblocker-11-million-users-one-server-call-away-from-compromise) for published backend infrastructure indicators`	Backend server domains and associated infrastructure used to deliver the remote JavaScript payload; specific domains and IPs not extracted in available source material	LOW

Framework Mappings

MITRE-ATTACK

- **T1059.007** — JavaScript
- **T1539** — Steal Web Session Cookie
- **T1071.001** — Web Protocols
- **T1657** — Financial Theft
- **T1176** — Software Extensions
- **T1185** — Browser Session Hijacking

OWASP-TOP10-2021

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

NIST-800-53R5

- **AC-3** — Access Enforcement
- **SI-10** — Information Input Validation
- **SI-4** — System Monitoring

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC7.4** — Responds to identified security incidents

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(6)(ii)** — Response and Reporting

ISO-27001-2022

- **A.8.28** — Secure coding
- **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
T1059.007	JavaScript	Execution
T1539	Steal Web Session Cookie	Credential-Access
T1071.001	Web Protocols	Command-And-Control
T1657	Financial Theft	Impact
T1176	Software Extensions	Persistence
T1185	Browser Session Hijacking	Collection

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/06/chrome-ad-blocker-with-10m-instal...	T3
Adblock for Youtube™ - Chrome Web Store	https://chromewebstore.google.com/detail/adblock-for-youtube/cmedhi...	T3
BadBlocker: 11 Million Users, One Server Call Away from Compromise	https://www.island.io/blog/badblocker-11-million-users-one-server-c...	T3
If youtube hates Adblockers, why does chrome have them in the store	https://www.reddit.com/r/youtube/comments/1r7se5y/if_youtube_hates_...	T3
The End of Ad Blockers In Chrome - YouTube	https://www.youtube.com/watch?v=TO48wxdjWMg	T3

Generated 2026-06-25 18:50 UTC by TJS Security Command Center