INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 13:59 UTC

Dual AI Credential Theft Campaign: JetBrains Plugin Supply Chain and Chrome Extension Interception Target Developer Secrets and Chatbot Data

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0500
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	JetBrains IDE Marketplace (15 malicious plugins), Google Chrome (Smart Adblocker, Adblock for Browser extensions, 100,000+ combined users), OpenAI ChatGPT, Anthropic Claude, Google Gemini, Microsoft Copilot, Perplexity, DeepSeek, xAI Grok, Meta AI, SiliconFlow
Published	2026-06-17T05:38:46
Discovery Source	Rss

Executive Summary

Two coordinated supply chain campaigns are actively stealing AI credentials and sensitive business data from developer tools and Chrome browsers. Fifteen malicious JetBrains Marketplace plugins have been harvesting AI API keys since October 2025, using them to run unauthorized AI inference at victims' expense, while two fake ad-blocker extensions have silently copied conversation histories from ChatGPT, Claude, Gemini, Copilot, and five other AI platforms across more than 100,000 affected browsers. Organizations face direct financial exposure from API key abuse, and risk losing proprietary source code, business logic, and intellectual property that employees shared with AI assistants.

Technical Analysis

Two concurrent supply chain vectors target AI credential surfaces and AI workflow data. Vector 1: 15 malicious JetBrains Marketplace plugins, active since October 2025 and still publishing as of June 10, 2026, harvest AI provider API keys (OpenAI, Anthropic, Google, and others) from developer IDE environments. Stolen keys feed an LLMjacking operation, adversaries consume AI inference capacity billed to victim accounts. Weaknesses: CWE-494 (plugin distributed without integrity check), CWE-312 (credentials stored in cleartext), CWE-319 (cleartext transmission). MITRE techniques: T1195.001 (Compromise Software Supply Chain), T1552.001

(Credentials in Files), T1528 (Steal Application Access Token). Vector 2: Two Chrome extensions, Smart Adblocker and Adblock for Browser, with 100,000+ combined users deploy content scripts to intercept and exfiltrate full conversation histories (prompts, responses, metadata) from ChatGPT, Claude, Gemini, Copilot, Perplexity, DeepSeek, xAI Grok, and Meta AI. Weaknesses: CWE-200 (information exposure), CWE-494. MITRE techniques: T1176 (Browser Extensions), T1539 (Steal Web Session Cookie), T1114 (Email Collection analog, conversation exfiltration), T1041 (Exfiltration Over C2 Channel), T1567 (Exfiltration Over Web Service). No CVE IDs are assigned. No CISA KEV entry exists at time of analysis. Attribution is unconfirmed for both operators. Sources rated T1 (Microsoft Security Blog) and T3 (BleepingComputer, The Hacker News, WindowsForum).

Action Checklist

1. Containment: Immediately audit all JetBrains IDE installations across developer workstations; cross-reference installed plugins against vendor advisories and threat intelligence sources. Remove any flagged plugins and revoke all AI provider API keys (OpenAI, Anthropic, Google, and others) accessible from affected developer environments. For Chrome, push a managed browser policy to block or remove Smart Adblocker and Adblock for Browser extensions organization-wide. Apply NIST AC-6 (Least Privilege) by restricting which developer systems can store or access AI API keys.
2. Detection: Query endpoint logs and browser extension inventories for Smart Adblocker and Adblock for Browser extension IDs. Review JetBrains plugin installation logs and IDE telemetry for malicious plugin names from threat intelligence sources. Check AI provider billing dashboards (OpenAI, Anthropic, Google) for anomalous API consumption spikes inconsistent with legitimate usage, a primary LLMjacking indicator. Correlate outbound network traffic from developer workstations to unknown domains against T1041/T1567 patterns. Enable AU-2 (Event Logging) and AU-6 (Audit Record Review) on developer endpoints if not already active. Apply D3-LAM (Local Account Monitoring) to surface unauthorized credential access on affected machines.
3. Eradication: Rotate all AI API keys that were present in any JetBrains IDE environment where a malicious plugin was installed; do not simply remove the plugin without revoking keys, as exfiltration may have already occurred. Uninstall the two malicious Chrome extensions from all affected browsers and clear associated browser session data. Audit all ChatGPT, Claude, Gemini, and Copilot account sessions for unauthorized access and revoke active sessions on affected accounts. Apply D3-CRO (Credential Rotation) for all affected AI service credentials. Apply D3-CH (Credential Hardening) to prevent plaintext API key storage in IDE configuration files.
4. Recovery: Validate that no residual malicious plugins remain in JetBrains Marketplace-connected environments by re-running plugin audits post-removal. Confirm new API keys are functional and that billing anomalies have ceased. Monitor AI provider dashboards for at least 30 days post-remediation for resumed LLMjacking activity indicating additional credential exposure. Enable D3-UAP (User Account Permissions) controls to restrict which developer roles can create or access AI API keys. Apply NIST AU-9 (Protection of Audit Information) to ensure post-incident logs are preserved for forensic review.
5. Post-Incident: Review organizational controls for third-party plugin and browser extension vetting; implement an approved extension allowlist enforced via managed browser policy (CIS 2.3: Address Unauthorized Software). Establish a process for scanning developer IDE plugin inventories on a recurring basis (CIS 7.1: Establish and Maintain a Vulnerability Management Process). Assess whether AI API keys are stored in secrets management systems rather than IDE config files or environment variables, and enforce this via policy. Evaluate whether conversation content shared with AI platforms constitutes

intellectual property or regulated data, and update AI usage policies accordingly. No mapped control exists in the provided knowledge base for browser extension content script monitoring, a gap to address in the next control review cycle.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal counsel immediately if forensic review of the Chrome extension Local Extension Settings storage or AI provider session audit logs reveals that conversation content containing PII, PHI, source code with embedded credentials, or client-confidential data was intercepted by the malicious extensions — this triggers breach notification assessment obligations under GDPR (72-hour window), CCPA, or HIPAA as applicable, and if LLMjacking token consumption exceeds a threshold constituting material financial loss, escalate to finance and cyber insurance carrier.
Recovery Notes	Post-eradication recovery is validated only when three conditions are simultaneously true: all AI provider billing dashboards show token consumption returning to developer-baseline levels for 72+ consecutive hours after new key issuance, a full re-scan of all JetBrains plugin directories across the developer fleet returns zero matches against the 15-plugin malicious list, and Chrome policy enforcement is confirmed active on all endpoints via chrome://policy audit. Continue monitoring AI provider usage dashboards daily for a minimum of 30 days post-remediation, as LLMjacking activity resuming after key rotation indicates either an additional undiscovered malicious plugin, a developer re-installing a flagged plugin from a personal device, or a secondary credential exposure pathway (e.g., API keys committed to a git repository) not addressed in the initial containment sweep. If any developer workstation cannot be fully audited due to remote work or BYOD status, treat those environments as uncontained and require fresh API key issuance only after remote verification.

Forensic Artifacts	JetBrains IDE plugin directory JAR files and plugin.xml descriptors at %APPDATA%\JetBrains\\plugins\ (Windows) or ~/.config/JetBrains//plugins/ (Linux) — the malicious plugin JARs contain obfuscated code responsible for reading AI API keys from IDE credential stores and environment variables; static analysis with jadx or procyon will reveal C2 domains and exfiltration logic specific to each of the 15 flagged plugins | JetBrains IDE application log at %APPDATA%\JetBrains\\log\idea.log — records plugin initialization events with timestamps, network request exceptions thrown by malicious plugins during exfiltration attempts, and any plugin-triggered credential store access events, establishing a precise timeline of API key theft per workstation | Chrome extension Local Extension Settings and IndexedDB storage at %LOCALAPPDATA%\Google\Chrome\User Data\Default\Local Extension Settings\\ — contains the malicious Smart Adblocker and Adblock for Browser extensions' locally buffered copies of intercepted AI platform conversation content from ChatGPT, Claude, Gemini, Copilot, Perplexity, DeepSeek, Grok, Meta AI, and SiliconFlow sessions, recoverable even after extension removal if the Chrome profile directory was imaged before clearing | Outbound network connection records (Sysmon Event ID 3 or Windows Firewall logs) from developer workstations filtered on processes matching JetBrains IDE executables — C2 callback traffic from the malicious plugins to attacker-controlled infrastructure used for API key exfiltration will appear as periodic low-volume HTTPS POST requests from the IDE process to non-JetBrains domains, distinct from legitimate plugin update traffic to plugins.jetbrains.com | AI provider API usage export logs (OpenAI /v1/usage endpoint, Anthropic console usage export, Google Cloud AI usage reports) for all developer-issued keys — unauthorized LLMjacking requests will appear as high-volume inference calls (chat completions, embeddings) originating from IP addresses outside the organization's known developer egress ranges, often at off-hours, providing both evidence of compromise and a quantified scope of unauthorized AI resource consumption

Per-Action IR Details

Containment — Immediately audit all JetBrains IDE installations across developer workstations; cross-reference installed plugins against the 15 malicious plugin names identified in BleepingComputer and The Hacker News reporting. Remove any flagged plugins and revoke all AI provider API keys (OpenAI, Anthropic, Google, and others) accessible from affected developer environments. For Chrome, push a managed browser policy to block or remove Smart Adblocker and Adblock for Browser extensions organization-wide. Apply NIST AC-6 (Least Privilege) by restricting which developer systems can store or access AI API keys.

NIST Phase: Containment

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

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), CIS 2.3 (Address Unauthorized Software)

Compensating: On Windows developer workstations, enumerate all installed JetBrains plugins via the IDE's plugin directory (typically %APPDATA%\JetBrains\\plugins\ or ~/.config/JetBrains//plugins/ on Linux) using: `Get-ChildItem -Path "$env:APPDATA\JetBrains" -Recurse -Filter "plugin.xml" | Select-String -Pattern ""` to extract plugin names, then diff against the malicious plugin list. For Chrome extensions, enumerate installed extension IDs via `Get-ChildItem "$env:LOCALAPPDATA\Google\Chrome\User Data\Default\Extensions"` and cross-reference Smart Adblocker and Adblock for Browser extension IDs. Without an MDM, distribute a PowerShell script via shared network drive to all developer machines and collect output centrally. Immediately rotate API keys via each provider's web console — OpenAI at platform.openai.com/api-keys, Anthropic at console.anthropic.com, Google at aistudio.google.com/app/apikey.

Evidence: Before removing any plugin or revoking API keys, capture volatile state that documents active exfiltration channels: (1) Run `netstat -ano` or `Get-NetTCPConnection` on each affected workstation to capture established outbound connections from JetBrains IDE processes (idea.exe, pycharm.exe, etc.) to external IP addresses — these

connections disappear upon plugin removal. (2) Collect a live memory dump of the IDE process using ProcDump (`procdump -ma ide_memdump.dmp`) to preserve in-memory API key strings and C2 URLs before the plugin is unloaded. (3) Export the JetBrains IDE local plugin cache and any plugin-specific configuration files (e.g., idea.properties, custom.properties, workspace.xml) that may contain exfiltrated or cached API key values before modification. (4) Capture Chrome extension local storage for Smart Adblocker and Adblock for Browser at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Local Extension Settings\\` before policy removal wipes it — this may contain buffered conversation history awaiting exfiltration.

Detection — Query endpoint logs and browser extension inventories for Smart Adblocker and Adblock for Browser extension IDs. Review JetBrains plugin installation logs and IDE telemetry for the 15 flagged plugin names. Check AI provider billing dashboards (OpenAI, Anthropic, Google) for anomalous API consumption spikes inconsistent with legitimate usage — a primary LLMjacking indicator. Correlate outbound network traffic from developer workstations to unknown domains against T1041/T1567 patterns. Enable AU-2 (Event Logging) and AU-6 (Audit Record Review) on developer endpoints if not already active. Apply D3-LAM (Local Account Monitoring) to surface unauthorized credential access on affected machines.

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, deploy Sysmon with a configuration that logs ProcessCreate (Event ID 1), NetworkConnect (Event ID 3), and FileCreate (Event ID 11) events on developer workstations. Write a Sigma rule targeting Sysmon Event ID 3 where the initiating image matches JetBrains IDE executables (idea.exe, pycharm.exe, webstorm.exe, goland.exe) and the DestinationHostname does not match known JetBrains update servers (plugins.jetbrains.com, downloads.jetbrains.com). For LLMjacking detection without SIEM: pull OpenAI usage logs via `curl https://api.openai.com/v1/usage -H "Authorization: Bearer "` for each potentially compromised key and compare daily token consumption against a developer's normal coding-session baseline. For Chrome extension detection without MDM, use osquery: `SELECT * FROM chrome_extensions WHERE identifier IN ('', '');` run against all developer endpoints via osquery's distributed query interface.

Evidence: This step is analytical and does not alter live system state; however, before enabling new logging (AU-2/AU-6), preserve existing log snapshots to establish a pre-detection baseline: (1) Export Windows Event Log — Security (Event ID 4663: object access on credential files; Event ID 4688: process creation showing IDE child processes) and Application logs from all developer workstations before Sysmon is deployed, as Sysmon installation rewrites kernel callbacks and may displace existing telemetry. (2) Capture JetBrains IDE log files at %APPDATA%\JetBrains\\log\idea.log (Windows) or ~/.cache/JetBrains//log/idea.log (Linux) — these logs record plugin load events, exceptions, and network requests made by plugins, providing a timeline of malicious plugin activity. (3) Export Chrome browser history and extension background page console logs from chrome://extensions/ (Developer mode) for Smart Adblocker and Adblock for Browser before any removal action — background page errors or XHR calls to non-ad-network domains are indicators of conversation exfiltration.

Eradication — Rotate all AI API keys that were present in any JetBrains IDE environment where a malicious plugin was installed; do not simply remove the plugin without revoking keys, as exfiltration may have already occurred. Uninstall the two malicious Chrome extensions from all affected browsers and clear associated browser session data. Audit all ChatGPT, Claude, Gemini, and Copilot account sessions for unauthorized access and revoke active sessions on affected accounts. Apply D3-CRO (Credential Rotation) for all affected AI service credentials. Apply D3-CH (Credential Hardening) to prevent plaintext API key storage in IDE configuration files.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-2 (Account Management), NIST IA-5 (Authenticator Management), CIS 5.2 (Use Unique Passwords), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Without enterprise PAM or secrets management, use a two-person verification procedure: one analyst revokes the compromised key at the provider console while a second confirms via the provider's audit log (OpenAI: Settings → API Keys → Last Used timestamp; Anthropic: console.anthropic.com/settings/keys) that the key shows no further usage after revocation time. To harden against plaintext API key storage in JetBrains IDE config files, grep all developer workstation JetBrains configuration directories for API key patterns before new keys are issued: `grep -r "sk-" ~/.config/JetBrains/ ~/.local/share/JetBrains/` (Linux) or `Select-String -Path "$env:APPDATA\JetBrains\*" -Pattern "sk-|AIza|ant-api" -Recurse` (Windows) — treat any matches as confirmed exfiltration candidates. For Chrome session revocation on ChatGPT, navigate to chatgpt.com → Settings → Security → Manage Sessions and revoke all sessions except the current verified one; repeat for Claude at claude.ai/settings, Gemini at myaccount.google.com/security.

Evidence: Before revoking API keys or clearing browser session data — both of which destroy live authentication state — capture the following: (1) Export the full API key usage history for each compromised key from each provider's dashboard (OpenAI usage export, Anthropic usage logs) — this documents the scope of LLMjacking: total tokens consumed, models called, and timestamps of unauthorized inference requests, which constitutes evidence of financial harm and unauthorized access. (2) Before clearing Chrome browser session data, image the Chrome profile directory at %LOCALAPPDATA%\Google\Chrome\User Data\Default\ (Windows) or ~/.config/google-chrome/Default/ (Linux) — the Cookies, Local Storage, and Session Storage files within this directory contain authentication tokens for ChatGPT, Claude, Gemini, and Copilot sessions that the malicious extension may have been reading or copying. (3) Capture the extension's IndexedDB and Local Extension Settings storage (as noted in Step 1) if not already collected — these may contain a local buffer of intercepted AI conversation fragments from all 10 targeted platforms awaiting exfiltration.

Recovery — Validate that no residual malicious plugins remain in JetBrains Marketplace-connected environments by re-running plugin audits post-removal. Confirm new API keys are functional and that billing anomalies have ceased. Monitor AI provider dashboards for at least 30 days post-remediation for resumed LLMjacking activity indicating additional credential exposure. Enable D3-UAP (User Account Permissions) controls to restrict which developer roles can create or access AI API keys. Apply NIST AU-9 (Protection of Audit Information) to ensure post-incident logs are preserved for forensic review.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-9 (Protection Of Audit Information), NIST AU-11 (Audit Record Retention), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a commercial plugin governance tool, establish a verified-clean plugin baseline using a free osquery query that hashes the contents of each installed JetBrains plugin directory: `SELECT name, identifier, version, hash FROM file WHERE path LIKE '/home/%/.config/JetBrains/%/plugins/%/plugin.xml';` — store the output as a signed baseline and schedule weekly osquery distributed queries to diff against it, alerting on any new entries. For 30-day LLMjacking monitoring without a SIEM, write a cron job (Linux) or scheduled PowerShell task (Windows) that calls each provider's usage API daily and writes token counts to a local CSV; flag any day exceeding 110% of the pre-incident 30-day rolling average. Protect collected incident logs by copying them to a write-once network share (WORM policy via Windows Server File Server Resource Manager or Linux `chattr +i`) to satisfy AU-9 without a dedicated log management platform.

Evidence: Recovery validation requires confirming that the attack surface has been fully closed — the primary risk here is an undetected second set of compromised keys or a missed plugin instance. Before declaring recovery complete: (1) Re-run the JetBrains plugin directory enumeration script from Step 1 on all developer workstations and diff against the clean baseline to confirm zero malicious plugin artifacts remain — specifically check the plugin cache directory (%APPDATA%\JetBrains\\system\plugins-sandbox\ on Windows) in addition to the active plugins folder, as cached plugin JARs may persist after UI-level removal. (2) Confirm via each AI provider's key management console that all pre-incident keys show a revoked/deleted status with a revocation timestamp that precedes any new key issuance — this validates that no window existed during which both old and new keys were simultaneously active and potentially exfiltrable. (3) Verify Chrome extension policy enforcement by checking chrome://policy on a sample of developer workstations to confirm ExtensionInstallBlocklist entries for both malicious extension IDs are applied and that chrome://extensions shows no installed instances.

Post-Incident — Review organizational controls for third-party plugin and browser extension vetting; implement an approved extension allowlist enforced via managed browser policy (CIS 2.3: Address Unauthorized Software). Establish a process for scanning developer IDE plugin inventories on a recurring basis (CIS 7.1: Establish and Maintain a Vulnerability Management Process). Assess whether AI API keys are stored in secrets management systems rather than IDE config files or environment variables, and enforce this via policy. Evaluate whether conversation content shared with AI platforms constitutes intellectual property or regulated data, and update AI usage policies accordingly. No mapped control exists in the provided knowledge base for browser extension content script monitoring — a gap to address in the next control review cycle.

NIST Phase: Post Incident

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

Controls: CIS 2.3 (Address Unauthorized Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: Without enterprise MDM for browser policy enforcement, distribute a Chrome Group Policy ADMX template (available free from Google at dl.google.com/dl/edgedl/chrome/policy/chrome_policy_list.pdf) configured with ExtensionInstallAllowlist containing only vetted extension IDs and ExtensionInstallBlocklist set to wildcard (*) — deploy via Windows Group Policy Object to all developer OUs at no cost. For recurring JetBrains plugin inventory scanning without a commercial tool, create a weekly scheduled task running the PowerShell plugin enumeration script from Step 1 and emailing the diff output to the security team alias. For API key secrets management at no cost, adopt HashiCorp Vault Community Edition or Mozilla SOPS with AWS KMS/GCP KMS free tier — document this as a policy requirement in the updated AI tool usage policy, with a 90-day migration deadline for all developer teams.

Evidence: Post-incident activity does not alter live system state and therefore does not have order-of-volatility constraints; however, the lessons-learned process should draw on preserved forensic artifacts: (1) Retrieve and analyze the JetBrains IDE idea.log entries collected in Step 2, specifically timestamped plugin load events for the 15 malicious plugins — these establish earliest-known-compromise dates per workstation, which informs the breach notification window assessment if regulated data (source code containing PII, proprietary algorithms, or credentials) was accessible in the IDE during the exposure period. (2) Review the API usage export data collected in Step 3 to quantify financial impact from LLMjacking (unauthorized token consumption billed to the organization) and document this for potential insurance claims or vendor reimbursement requests. (3) Compile the Chrome Local Extension Settings storage images from Step 1 and perform keyword search against known project names, client names, or data classification terms to determine whether AI conversation content captured by the malicious extensions constituted regulated data under applicable frameworks (GDPR, CCPA, HIPAA), which governs breach notification obligations.

Detection Guidance

JetBrains vector: Enumerate all installed plugins across developer IDEs using JetBrains toolbox or endpoint management tooling; compare against malicious plugin indicators from threat intelligence sources and vendor advisories. Query endpoint EDR telemetry for suspicious outbound connections from JetBrains IDE processes, particularly POST requests to unknown domains carrying API key-shaped strings. Review AI provider billing APIs or dashboards for consumption spikes: OpenAI usage dashboard at platform.openai.com/usage, Anthropic console, Google AI Studio, anomalous token consumption during off-hours is a primary LLMjacking behavioral indicator (T1528, T1041). Chrome extension vector: Query managed browser telemetry or Chrome Enterprise reports for extension IDs associated with Smart Adblocker and Adblock for Browser. Review outbound network logs for HTTPS POST traffic from browser processes to domains not matching the eight targeted AI platforms, content script exfiltration will appear as browser-originated traffic to adversary-controlled infrastructure (T1176, T1567). Behavioral indicator: users reporting unexpected AI session activity, unfamiliar prompts in conversation history, or AI accounts showing logins from unknown locations (T1539). Apply D3-SFA (System File Analysis) to

scan developer workstation file systems and IDE configuration directories for plaintext API key strings in config files, .env files, or IDE settings exports.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://plugins.jetbrains.com/`	JetBrains Marketplace — source distribution channel for 15 malicious plugins; specific plugin names available in BleepingComputer reporting. Validate against vendor advisory before acting.	MEDIUM
URL	`Smart Adblocker (Chrome Web Store extension name)`	Malicious Chrome extension exfiltrating AI chatbot conversation histories from eight platforms. Remove from all managed Chrome instances.	HIGH
URL	`Adblock for Browser (Chrome Web Store extension name)`	Second malicious Chrome extension in the PromptSnatcher campaign; over 100,000 combined installs with Smart Adblocker.	HIGH

Framework Mappings

MITRE-ATTACK

- **T1567** — Exfiltration Over Web Service
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1041** — Exfiltration Over C2 Channel
- **T1176** — Software Extensions
- **T1557** — Adversary-in-the-Middle
- **T1552.001** — Credentials In Files
- **T1528** — Steal Application Access Token
- **T1539** — Steal Web Session Cookie
- **T1114** — Email Collection

NIST-800-53R5

- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-4** — System Monitoring
- **SC-8** — Transmission Confidentiality and Integrity
- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest
- **SI-7** — Software, Firmware, and Information Integrity

- **CM-3** — Configuration Change Control
- **SR-2** — Supply Chain Risk Management Plan
- **SC-13** — Cryptographic Protection

OWASP-TOP10-2021

- **A02:2021** — Cryptographic Failures
- **A01:2021** — Broken Access Control
- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **3.10** — Encrypt Sensitive Data in Transit
- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **6.3** — Require MFA for Externally-Exposed Applications
- **15.1** — Establish and Maintain an Inventory of Service Providers

HIPAA-SECURITY

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

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC9.2** — Manages risks associated with vendors and business partners

ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information
- **A.5.21** — Managing information security in the ICT supply chain
- **A.8.24** — Use of cryptography

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1567	Exfiltration Over Web Service	Exfiltration
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1041	Exfiltration Over C2 Channel	Exfiltration

Technique ID	Technique Name	Tactic
T1176	Software Extensions	Persistence
T1557	Adversary-in-the-Middle	Credential-Access
T1552.001	Credentials In Files	Credential-Access
T1528	Steal Application Access Token	Credential-Access
T1539	Steal Web Session Cookie	Credential-Access
T1114	Email Collection	Collection

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/06/malicious-jetbrains-plugins-steal...	T3
Malicious Ad Blockers Stole AI Prompts and Metadata	https://windowsforum.com/threads/promptsnatcher-malicious-ad-blocke...	T3
Malicious JetBrains Marketplace plugins steal AI API keys from ...	https://www.bleepingcomputer.com/news/security/malicious-jetbrains-...	T3
Malicious AI Assistant Extensions Harvest LLM Chat Histories	https://www.microsoft.com/en-us/security/blog/2026/03/05/malicious-...	T1
30 Malicious Chrome Extensions - YouTube	https://www.youtube.com/watch?v=_gsDYhGHdnc	T3

Generated 2026-06-17 13:59 UTC by TJS Security Command Center