INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-06 15:14 UTC

Opera GX Auto-Install Flaw Enabled Zero-Click CSS Exfiltration Across Every Site a Victim Visited

SECURITY ANALYSIS | HIGH | CVSS 5.0

SCC Item ID	SCC-STY-2026-0325
Type	Security Analysis
Severity	HIGH
CVSS Base Score	5.0
Affected Products	Opera GX (patched in v130.0.5847.89); Opera browser (crash variant affected)
Published	2026-07-06T03:27:50
Discovery Source	Rss

Executive Summary

A flaw in Opera GX's browser mod pipeline, reported by security news outlets, allowed malicious websites to silently install browser mods without any user interaction, then inject CSS across every site a victim visited to reconstruct sensitive identity data, including Gmail addresses, one character at a time. The attack required no clicks, no downloads, and no visible indicators of compromise, making it exceptionally difficult for users to detect. Opera has issued a patch in version 130.0.5847.89; however, the incident signals a broader architectural risk in browser extension and mod ecosystems where consent enforcement and cross-origin isolation assumptions remain weak.

Technical Analysis

According to reporting from The Hacker News, Infosecurity Magazine, and GBHackers, the Opera GX vulnerability combined three distinct weaknesses into a zero-click exfiltration chain. The first weakness (CWE-862) involved an unguarded auto-install mechanism in Opera GX's mod pipeline: a malicious website could trigger mod installation without meaningful user consent enforcement, mapping to MITRE ATT&CK T1176 (Browser Extensions) and T1189 (Drive-by Compromise). Once installed, the malicious mod exploited its privileged access to inject arbitrary CSS into every origin the victim visited, a cross-site scope violation corresponding to CWE-79 (Improper Neutralization of Input During Web Page Generation) and T1059.007 (JavaScript execution in the browser context). The third layer (CWE-116) involved encoding and escaping failures that allowed the injected CSS to carry exfiltration payloads, enabling a CSS-based cross-site leak (XS-Leak) technique. Using CSS attribute selectors combined with attacker-controlled image requests, each

request encoding one character of a target value, the attack reconstructed a victim's Gmail address character-by-character. This technique maps to T1185 (Man-in-the-Browser) in terms of information access scope. No user interaction was required at any stage after the initial drive-by visit. The source set for this story consists entirely of Tier 2 and Tier 3 secondary outlets; no first-party advisory from Opera, NVD entry, or CISA alert has been confirmed in the provided material. Exploitation scope and impact figures should be treated as reported but not independently corroborated at the authoritative tier. No CVE has been assigned as of the configuration date, and no confirmed wild exploitation has been reported.

Action Checklist

1. Step 1: Assess exposure, determine whether Opera GX or the standard Opera browser is deployed on any enterprise endpoints, managed or unmanaged (BYOD), and identify whether any installed mods are present; confirm browser version against the patched release v130.0.5847.89
2. Step 2: Review controls, verify that browser management policy enforces approved extension/mod allowlists and blocks unapproved installations (NIST AC-3: Access Enforcement; CIS 2.3: Address Unauthorized Software; CIS 2.1: Establish and Maintain a Software Inventory); confirm endpoint firewall rules restrict unauthorized outbound image or resource requests that could serve as exfiltration channels (CIS 4.4, CIS 4.5)
3. Step 3: Update threat model, add CSS-based XS-Leak exfiltration and browser mod abuse as attack patterns in your threat register; map to MITRE ATT&CK T1176, T1189, T1059.007, and T1185 (Man-in-the-Browser); assess whether other Chromium-based browsers or extension-enabled products in your environment have similar auto-install or privileged CSS injection risks
4. Step 4: Communicate findings, brief leadership on the specific risk: a browser used on endpoints, including personal devices accessing corporate web applications, could silently exfiltrate employee or customer identity data from any visited site without triggering conventional security controls; frame the risk in terms of corporate email account exposure and credential reconnaissance
5. Step 5: Monitor developments, track for a formal CVE assignment, an official Opera security advisory, and any CISA or NVD entry for this vulnerability; watch for follow-up researcher disclosure of proof-of-concept code that could lower the barrier for exploitation; revisit exposure assessment if a CVE or KEV listing is published

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent if: (1) any enterprise endpoint is confirmed running Opera GX prior to v130.0.5847.89 while authenticated to corporate Gmail, Microsoft 365, or any SaaS application containing PII/PHI — triggering potential breach notification obligations under GDPR, CCPA, or HIPAA given the identity data exfiltration capability; or (2) a formal CVE is assigned and added to the CISA Known Exploited Vulnerabilities catalog, indicating active in-the-wild exploitation.

Recovery Notes	Ensure all Opera GX instances across managed and BYOD endpoints are verified at v130.0.5847.89 or later before considering the exposure window closed; unmanaged BYOD devices should be considered unverifiable without self-attestation and treated as potentially exposed. For any endpoint confirmed to have run a vulnerable Opera GX version while authenticated to corporate web applications, conduct a 30-day retrospective review of outbound network logs for the browser process — specifically looking for sequences of rapid GET requests to a single external domain with incrementing single-character query parameters, which is the network signature of CSS character-by-character exfiltration. Monitor the NVD, CISA KEV catalog, and Opera's official security advisory channel for a minimum of 90 days for PoC publication or active exploitation confirmation that would warrant escalating this from a patching event to a full incident investigation.
Forensic Artifacts	Opera GX Extensions directory snapshot: `%APPDATA%\Opera Software\Opera GX Stable\Extensions\` (Windows) or `~/Library/Application Support/com.operasoftware.OperaGX/Extensions/` (macOS) — records installed mod IDs, permissions, and install timestamps; a mod installed without user interaction during the vulnerability window is the primary indicator of silent auto-install exploitation | Opera GX Preferences and Secure Preferences files in the browser profile root — contain the extension/mod registry including install source, enabled state, and granted permissions; compare against known-good baselines to identify mods not present before the vulnerability window | Outbound network traffic pcap from the vulnerable endpoint: filter for the browser process making sequential GET requests to an external domain with single-character query string values (e.g., `GET /beacon?c=g`, `GET /beacon?c=m`, `GET /beacon?c=a`...) — this is the CSS XS-Leak exfiltration channel reconstructing a Gmail address one character at a time | Browser DNS cache and connection history: on Windows, `ipconfig /displaydns` output and browser's `%APPDATA%\Opera Software\Opera GX Stable\Network\` directory — identifies attacker-controlled exfiltration domains contacted by the malicious mod's CSS injection | Windows Security Event Log Event ID 4688 (Process Creation) entries for `opera.exe` and any child processes, correlated with the timestamps of mod installation artifacts — establishes the timeline of browser activity during the exposure window and may reveal unusual subprocess invocations associated with mod execution

Per-Action IR Details

Step 1: Assess exposure — determine whether Opera GX or the standard Opera browser is deployed on any enterprise endpoints, managed or unmanaged (BYOD), and identify whether any installed mods are present; confirm browser version against the patched release v130.0.5847.89

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: scope and characterize the affected asset population before any further action

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Run `wmic product where 'name like "%Opera%"' get name,version` (Windows) or `find /Applications -name 'Opera*' -maxdepth 2` (macOS) across managed endpoints via a scheduled osquery query: `SELECT name, version FROM apps WHERE name LIKE '%Opera%';`. For BYOD, issue a self-attestation form requesting browser version and cross-reference against v130.0.5847.89. Enumerate installed Opera GX mods by inspecting the profile directory at `%APPDATA%\Opera Software\Opera GX Stable\Extensions\` (Windows) or `~/Library/Application Support/com.operasoftware.OperaGX/Extensions/` (macOS).

Evidence: This step is inventory-only and does not alter live state. However, before any remediation or mod removal occurs in later steps, capture the current mod/extension directory listings and browser profile metadata intact: snapshot `%APPDATA%\Opera Software\Opera GX Stable\` including `Preferences`, `Secure Preferences`, and `Extensions\`

subdirectories. These files record installed mod IDs, permissions, and install timestamps — the primary artifact trail for determining whether a malicious mod was silently installed via the auto-install flaw.

Step 2: Review controls — verify that browser management policy enforces approved extension/mod allowlists and blocks unapproved installations (NIST AC-3: Access Enforcement; CIS 2.3: Address Unauthorized Software; CIS 2.1: Establish and Maintain a Software Inventory); confirm endpoint firewall rules restrict unauthorized outbound image or resource requests that could serve as exfiltration channels (CIS 4.4, CIS 4.5)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: verify that preventive and detective controls are in place to reduce the impact of this class of browser mod and CSS exfiltration attack

Controls: NIST AC-3 (Access Enforcement), NIST AC-4 (Information Flow Enforcement), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.3 (Address Unauthorized Software), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Enforce Opera GX mod policy via Group Policy or Intune by setting `ExtensionInstallBlocklist` to `*` and `ExtensionInstallAllowlist` to only approved mod IDs (Opera GX supports Chromium enterprise policies). For outbound exfiltration channel blocking without a SIEM, deploy Windows Firewall rules via PowerShell: `New-NetFirewallRule -DisplayName 'Block Opera GX Outbound Unknown' -Direction Outbound -Program '%LOCALAPPDATA%\Programs\Opera GX\opera.exe' -RemotePort 80,443 -Action Block` as a temporary measure pending allowlist review. Use Wireshark with display filter `http.request.method == 'GET' && http.request.uri contains '?'` on a monitored endpoint to spot anomalous single-character CSS-driven image beacon requests.

Evidence: This step reviews policy configuration and does not alter live browser state. Before modifying any firewall rules, export existing outbound firewall rule sets: `netsh advfirewall export 'C:\IR\firewall_baseline_pre_change.wfw'`. If reviewing live traffic to assess whether exfiltration channels were already active, capture a Wireshark pcap on the suspect endpoint prior to any network policy changes — the CSS XS-Leak exfiltration technique generates sequential single-pixel or zero-byte image requests to an attacker-controlled domain, one per character of the exfiltrated value (e.g., Gmail address characters), which will appear as a long series of rapid GET requests with single-character query parameters.

Step 3: Update threat model — add CSS-based XS-Leak exfiltration and browser mod abuse as attack patterns in your threat register; map to MITRE ATT&CK T1176, T1189, T1059.007, and T1185; assess whether other Chromium-based browsers or extension-enabled products in your environment have similar auto-install or privileged CSS injection risks

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: use lessons learned from this threat to update detection capabilities, threat models, and organizational awareness

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Document the threat model update in a shared markdown or wiki page. Enumerate all Chromium-based browsers in the software inventory (Edge, Chrome, Brave, Vivaldi, Opera, Opera GX) using osquery: `SELECT name, version, path FROM apps WHERE name LIKE '%Chrome%' OR name LIKE '%Brave%' OR name LIKE '%Edge%' OR name LIKE '%Opera%' OR name LIKE '%Vivaldi%';`. For each, verify whether the vendor supports enterprise extension management policies and document gaps. Create a Sigma rule to alert on mass rapid outbound GET requests from browser processes to a single external domain with incrementing single-character query strings — the behavioral fingerprint of CSS character-by-character exfiltration.

Evidence: This is a threat model and documentation step with no live-system changes. No volatile evidence capture is required. Retain the Wireshark pcaps, browser profile snapshots, and mod directory listings from Steps 1 and 2 as reference artifacts for the threat model update. If a Sigma or YARA rule is being authored, use these artifacts to validate the rule against actual artifacts from a test environment rather than theoretical patterns.

Step 4: Communicate findings — brief leadership on the specific risk: a browser used on endpoints, including personal devices accessing corporate web applications, could silently exfiltrate employee or customer identity data from any visited site without triggering conventional security controls; frame the risk in terms of corporate email account exposure and credential reconnaissance

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: communicate incident findings and risk implications to organizational leadership to support informed decision-making and policy updates

Compensating: Prepare a one-page executive summary that quantifies blast radius: identify how many endpoints run Opera GX, how many BYOD devices access corporate webmail or SaaS applications (e.g., Google Workspace, Microsoft 365), and whether any employees used Opera GX on systems where corporate Gmail or Outlook is authenticated. The zero-click, no-indicator nature of this attack means absence of alerts does not indicate absence of compromise — communicate this clearly. Use the browser version inventory from Step 1 as the data source.

Evidence: No live-system changes occur in this step. For the leadership brief, reference the mod directory snapshots and software inventory data collected in Steps 1 and 2 to support factual claims about enterprise exposure. If any endpoints were running Opera GX versions prior to v130.0.5847.89 during the vulnerability window, note that silent exfiltration of authenticated Gmail addresses (or other identity data from visited sites) cannot be ruled out without network traffic forensics from that period.

Step 5: Monitor developments — track for a formal CVE assignment, an official Opera security advisory, and any CISA or NVD entry for this vulnerability; watch for follow-up researcher disclosure of proof-of-concept code that could lower the barrier for exploitation; revisit exposure assessment if a CVE or KEV listing is published

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: maintain situational awareness and update the incident response posture as new threat intelligence becomes available for this vulnerability

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Subscribe to NVD new CVE notifications filtered for 'Opera' at https://nvd.nist.gov/vuln/search and to CISA Known Exploited Vulnerabilities catalog RSS feed at https://www.cisa.gov/known-exploited-vulnerabilities-catalog. Set a Google Alert for 'Opera GX mod auto-install CVE' and 'CSS XS-Leak exfiltration PoC'. Schedule a 30-day re-assessment checkpoint: if a CVE is assigned or PoC code is published, immediately re-run the exposure inventory from Step 1 and escalate triage priority from standard to urgent or immediate based on exploitation evidence. If a KEV listing is published, treat remediation as immediate regardless of current patch status.

Evidence: This is a monitoring and intelligence-tracking step with no live-system changes. No volatile capture is required. Retain the software inventory snapshots and browser version records from Step 1 as a baseline for the 30-day re-assessment comparison. If PoC code is published, collect and preserve a copy in a sandboxed environment for detection rule development — specifically to generate real network traffic samples of the CSS character-exfiltration beacon pattern for Sigma/Wireshark rule validation.

Detection Guidance

No verified IOC values (C2 domains, payload hashes, malicious mod identifiers) were present in the provided source material. Detection should focus on behavioral and log-based signals. Review browser process logs and endpoint telemetry for unexpected mod or extension installation events in Opera GX, particularly those not initiated through the official mod store workflow or IT-managed allowlist (CIS 2.3; NIST AU-2: Event Logging). Hunt for anomalous outbound HTTP/HTTPS requests to unfamiliar domains triggered by image resource loads, specifically high-frequency, sequenced single-character-parameter requests that are consistent with CSS

attribute selector enumeration (AU-6: Audit Record Review, Analysis, and Reporting). If web proxy or DNS logging is available, look for repeated low-payload GET requests to the same external domain from the same endpoint within a short time window, which is a behavioral signature of character-by-character CSS exfiltration. Audit browser policy enforcement to confirm that mod auto-install is disabled or requires explicit IT approval (NIST AC-3; CIS 4.6: Securely Manage Enterprise Assets and Software). Apply the principle of least privilege (NIST AC-6) to browser mod permissions; mods should not carry cross-origin CSS injection capability by default. Endpoint detection and response (EDR) or endpoint logging may surface unexpected privilege use if mods operate under user context with broad site access. For organizations running enterprise web applications, review Content Security Policy (CSP) headers on internal apps to determine whether CSS injection from a browser-level context could bypass server-side controls.

Framework Mappings

MITRE-ATTACK

- **T1204.001** — Malicious Link
- **T1557** — Adversary-in-the-Middle
- **T1176** — Software Extensions
- **T1189** — Drive-by Compromise
- **T1059.007** — JavaScript
- **T1185** — Browser Session Hijacking
- **T1565.002** — Transmitted Data Manipulation

OWASP-TOP10-2021

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

NIST-800-53R5

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

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.1** — Establish an Access Granting Process
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

ISO-27001-2022

- **A.8.28** — Secure coding
- **A.8.8** — Management of technical vulnerabilities

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1204.001	Malicious Link	Execution
T1557	Adversary-in-the-Middle	Credential-Access
T1176	Software Extensions	Persistence
T1189	Drive-by Compromise	Initial-Access
T1059.007	JavaScript	Execution
T1185	Browser Session Hijacking	Collection
T1565.002	Transmitted Data Manipulation	Impact

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/07/opera-gx-flaw-let-malicious-sites...	T2
Opera GX Flaw Let Sites Auto-Install Mods to Steal Data	https://www.infosecurity-magazine.com/news/opera-gx-flaw-gx-mods-css/	T2
Opera GX Universal CSS Injection Flaw Enabled Zero-Click XSLeak ...	https://cyberpress.org/opera-gx-universal-css-injection-flaw/	T3
Opera GX flaw allowed silent mod installs and data theft from visited ...	https://www.mallory.ai/stories/019f3679-3ead-7d8f-8d9a-72c8f7e9d65a	T3
Critical Opera GX Vulnerability Lets Attackers Inject CSS Across ...	https://gbhackers.com/critical-opera-gx-vulnerability/	T2

Generated 2026-07-06 15:14 UTC by TJS Security Command Center