INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-27 18:55 UTC

Chrome 148 Emergency Patch: Actively Exploited RCE Flaw Demands Immediate Enterprise Action

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0234
Type	CVE Vulnerability
CVE ID	CVE-2026-2441
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.2313 (96th percentile)
Affected Products	Google Chrome versions prior to 148.0.7778.96 on Windows, macOS, and Linux
Published	2026-05-27T17:52:30+00:00
Discovery Source	Rss:T1 Psirt

Executive Summary

Google issued an emergency out-of-band patch for Chrome 148 on May 27, 2026, addressing CVE-2026-2441, a use-after-free flaw in Chrome's CSS engine that enables remote code execution and is confirmed actively exploited in the wild. Every enterprise running Chrome prior to version 148.0.7778.96 on Windows, macOS, or Linux is exposed. With a CVSS score of 9.5, this vulnerability represents an immediate, high-probability risk of endpoint compromise across the organization.

Technical Analysis

CVE-2026-2441 is a use-after-free vulnerability (CWE-416) in Google Chrome's CSS rendering engine, enabling remote code execution (CWE-94). Affected versions: all Chrome releases prior to 148.0.7778.96 on Windows, macOS, and Linux. The vulnerability is exploitable via a maliciously crafted web page (T1189, Drive-by Compromise), requiring no user interaction beyond page load. Successful exploitation can yield arbitrary code execution in the renderer process, with documented escalation paths to sandbox escape (T1068) and process injection (T1055). CVSS base score: 9.5. EPSS score: 0.231 (96th percentile), indicating high exploitation probability relative to the CVE population. Active exploitation confirmed by Google. The patch was released as an out-of-band emergency update, consistent with Google's zero-day response posture. Relevant MITRE ATT&CK techniques: T1189 (Drive-by Compromise), T1055 (Process Injection), T1203 (Exploitation for Client Execution), T1059 (Command and Scripting Interpreter), T1068 (Exploitation for Privilege Escalation).

CWE references: CWE-416 (Use After Free), CWE-94 (Improper Control of Code Generation). Official patch advisory: https://chromereleases.googleblog.com/

Action Checklist

1. Step 1: Containment. Immediately identify all enterprise endpoints running Chrome prior to 148.0.7778.96 via your asset inventory (CIS 1.1). Restrict internet browsing on unpatched endpoints using host-based firewall rules (CIS 4.5, NIST AC-4) until patching is complete. Consider blocking known malicious URLs at the web proxy or DNS layer as an interim control (NIST SC-7).
2. Step 2: Detection. Query EDR and endpoint logs for Chrome process anomalies: unexpected child process spawning from chrome.exe or chrome (Linux/macOS), memory injection events (T1055), or lateral movement originating from browser processes. Review proxy and DNS logs for drive-by download indicators (T1189): high-entropy domains, newly registered domains, or unusual JS/WASM payload downloads. Check SIEM for NIST AU-6 audit review triggers on affected hosts. Apply D3-SFA (System File Analysis) to monitor for unexpected Chrome binary modifications.
3. Step 3: Eradication. Push Chrome 148.0.7778.96 or later to all endpoints immediately via automated patch management (CIS 7.3, CIS 7.4). Verify update via Chrome's built-in version check (chrome://settings/help) or endpoint management console. For managed fleets, push the update through your MDM or software deployment platform without waiting for the standard patch cycle. Validate that no instances of pre-148.0.7778.96 Chrome remain in the software inventory (CIS 2.1).
4. Step 4: Recovery. After patching, validate Chrome version across all endpoints using your configuration management tool or EDR fleet query. Monitor process and network telemetry on recently patched endpoints for 48-72 hours for residual indicators of prior compromise (T1055, T1068). Review AU-6 audit records for any anomalous activity during the exposure window. Confirm host-based firewall rules applied in Step 1 are reverted or made permanent as appropriate (NIST AC-4).
5. Step 5: Post-Incident. Evaluate your patch cycle SLA against this timeline: Google released an out-of-band emergency update, your process should have a documented emergency patching track separate from monthly cadence (CIS 7.2, NIST IR-4). Audit browser management controls: are all Chrome instances centrally managed and enrolled in auto-update policies? Review CIS 2.2 compliance, confirm only supported, current browser versions are authorized. Document any gap between patch release and full fleet deployment as a remediation metric.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to full incident response (engage CISO, legal, and if applicable breach notification counsel) if any endpoint shows Sysmon Event ID 1 or 8 evidence of a child process spawned from Chrome's renderer process, registry persistence created during the exposure window, or proxy/DNS logs confirm connection to known CVE-2026-2441 exploit delivery infrastructure — any of these conditions indicates the vulnerability was successfully exploited and a compromised host may be in scope for regulatory breach notification if it processed PII or PHI.

Recovery Notes	After confirming 100% patch coverage to Chrome 148.0.7778.96 or later, maintain elevated monitoring on all endpoints that ran a vulnerable Chrome version during the May 27, 2026 exposure window for a minimum of 72 hours — focus Sysmon and network telemetry specifically on post-exploitation persistence (new scheduled tasks, autorun registry keys, unexpected outbound connections from user-context processes) that would survive the browser patch. Any host that shows anomalous indicators during this window should be treated as potentially compromised and escalated to full forensic triage rather than considered remediated by patching alone, since patching closes the vulnerability but does not evict an attacker who achieved RCE prior to the patch.
Forensic Artifacts	Chrome SQLite History database at '%LOCALAPPDATA%\Google\Chrome\User Data\Default\History' (Windows) or '~/.config/google-chrome/Default/History' (Linux) — query the 'urls' and 'visits' tables for the exposure window to identify exploit delivery URLs; a CSS-engine use-after-free like CVE-2026-2441 is delivered via a crafted webpage, so the delivery URL will appear here | Chrome HTTP cache directory at '%LOCALAPPDATA%\Google\Chrome\User Data\Default\Cache\' — may contain cached copies of the malicious JavaScript or WebAssembly payload used to trigger the CVE-2026-2441 CSS use-after-free, recoverable with tools like ChromeCacheView or manually via cache entry parsing | Windows Security Event Log Event ID 4688 (Process Creation) and Sysmon Event ID 1 filtered on ParentImage matching chrome.exe with RendererCommandLine containing '--type=renderer' — a successful sandbox escape from this RCE flaw would manifest as the renderer process spawning cmd.exe, powershell.exe, or a dropper executable, which is definitively anomalous and exploit-specific | Memory dump of chrome.exe and its renderer child processes captured pre-patch using ProcDump ('procdump -ma chrome.exe') — the use-after-free heap corruption artifact from CVE-2026-2441 CSS engine exploitation may be recoverable in a live or post-mortem memory image and can confirm exploitation occurred on a specific host | Windows Prefetch files at 'C:\Windows\Prefetch\' and Sysmon Event ID 7 (Image Loaded) logs — if the CVE-2026-2441 RCE resulted in a dropper execution, prefetch entries and loaded DLL records will show the first execution timestamp and path of any payload binary dropped post-exploitation, establishing the timeline of compromise relative to the patch release date

Per-Action IR Details

Step 1: Containment — Immediately identify all enterprise endpoints running Chrome prior to 148.0.7778.96 via your asset inventory (CIS 1.1). Restrict internet browsing on unpatched endpoints using host-based firewall rules (CIS 4.5, NIST AC-4) until patching is complete. Consider blocking known malicious URLs at the web proxy or DNS layer as an interim control (NIST SC-7).

NIST Phase: Containment

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

Controls: NIST AC-4 (Information Flow Enforcement), NIST SC-7 (Boundary Protection), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Use osquery to enumerate Chrome versions fleet-wide: `SELECT name, version, install_location FROM programs WHERE name LIKE '%Chrome%'` (Windows) or `SELECT name, version FROM deb_packages WHERE name = 'google-chrome-stable'` (Linux). On Windows, apply a host-based firewall rule via PowerShell to block outbound HTTP/HTTPS on unpatched hosts: `New-NetFirewallRule -DisplayName 'Block Chrome HTTP' -Program 'C:\Program Files\Google\Chrome\Application\chrome.exe' -Action Block -Direction Outbound`. At the DNS layer, implement RPZ (Response Policy Zone) blocks on your recursive resolver for known malicious domains if you have logs showing suspicious pre-patch browsing activity.

Evidence: Before restricting network access, capture full proxy/DNS query logs for the 72-hour window prior to containment — specifically look for HTTP 200 responses serving .js or .wasm payloads from newly registered or

high-entropy domains, which are characteristic of drive-by delivery exploiting browser rendering engine flaws like this CSS use-after-free. Snapshot Chrome's running process list (`chrome.exe` PID tree on Windows via `wmic process where name='chrome.exe' get ProcessId,ParentProcessId,CommandLine`) and capture any open network connections from Chrome processes (`netstat -b` or `ss -tp`) before firewall rules terminate them.

Step 2: Detection — Query EDR and endpoint logs for Chrome process anomalies: unexpected child process spawning from chrome.exe or chrome (Linux/macOS), memory injection events (T1055), or lateral movement originating from browser processes. Review proxy and DNS logs for drive-by download indicators (T1189): high-entropy domains, newly registered domains, or unusual JS/WASM payload downloads. Check SIEM for NIST AU-6 audit review triggers on affected hosts. Apply D3-SFA (System File Analysis) to monitor for unexpected Chrome binary modifications.

NIST Phase: Detection Analysis

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

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

Compensating: Deploy Sysmon with SwiftOnSecurity's config and add a targeted rule to catch suspicious child processes of Chrome: look for Event ID 1 (Process Create) where ParentImage ends in `chrome.exe` and Image is `cmd.exe`, `powershell.exe`, `wscript.exe`, `mshta.exe`, or `rundll32.exe`. CVE-2026-2441 exploitation via the CSS engine use-after-free would manifest as chrome's renderer process (running with `--type=renderer`) spawning unexpected executables — filter Sysmon Event ID 1 on ParentCommandLine containing `--type=renderer`. For memory injection (T1055), monitor Sysmon Event ID 8 (CreateRemoteThread) and Event ID 10 (ProcessAccess) targeting non-Chrome processes from Chrome PIDs. On Linux/macOS, use auditd rules: `auditctl -a always,exit -F arch=b64 -S execve -F ppid=$(pgrep -x chrome)` to catch child process execution. For network indicators, run Zeek or Wireshark capture filtered on `http.request.uri matches \.wasm` or TLS SNI to domains registered within 30 days (cross-reference with WhoisXML API free tier or CIRCL passive DNS).

Evidence: Capture Windows Security Event Log Event ID 4688 (Process Creation with command-line auditing enabled) filtered on processes spawned by any `chrome.exe` PID. Collect Sysmon Event ID 3 (Network Connection) records from Chrome renderer processes connecting to external IPs — post-exploitation lateral movement from a compromised Chrome sandbox escape (T1055, T1068) would show chrome's renderer process initiating SMB (445/TCP) or WinRM (5985/TCP) connections, which is anomalous. Preserve Chrome's user profile directory (`%LOCALAPPDATA%\Google\Chrome\User Data\Default\`) including Cache, History, and Network Persistent State files — the exploit delivery URL will appear in Chrome's SQLite History database and cached response headers may contain the malicious JS/WASM payload.

Step 3: Eradication — Deploy Chrome 148.0.7778.96 or later to all endpoints immediately via automated patch management (CIS 7.3, CIS 7.4). Verify update via Chrome's built-in version check (chrome://settings/help) or endpoint management console. For managed fleets, push the update through your MDM or software deployment platform without waiting for the standard patch cycle. Validate that no instances of pre-148.0.7778.96 Chrome remain in the software inventory (CIS 2.1).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-6 (Configuration Settings), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: For unmanaged or BYOD endpoints without MDM, script a forced Chrome update using the Google Update service on Windows: `reg add HKLM\SOFTWARE\Policies\Google\Update /v UpdateDefault /t REG_DWORD /d 1 /f` followed by `"C:\Program Files (x86)\Google\Update\GoogleUpdate.exe" /ua /installsource scheduler`. On Linux, run `apt-get install --only-upgrade google-chrome-stable` or `yum update google-chrome-stable` via a pushed shell script. Post-update, verify version with: PowerShell — `(Get-Item 'C:\Program Files\Google\Chrome\Application\chrome.exe').VersionInfo.FileVersion`; Linux/macOS — `google-chrome --version`. Flag any host still returning a version below `148.0.7778.96` for manual intervention and maintain network restriction

from Step 1 until confirmed patched.

Evidence: Before deploying the patch, preserve a forensic image or at minimum a memory dump (`procdump -ma chrome.exe chrome_preupdate.dmp` on Windows using Sysinternals ProcDump) of Chrome on any host showing anomalous indicators from Step 2 — the use-after-free in the CSS engine may leave heap corruption artifacts in a live memory capture that can confirm exploitation. Document the exact Chrome version string and binary hash (`Get-FileHash 'C:\Program Files\Google\Chrome\Application\chrome.exe' -Algorithm SHA256`) from pre-patch state on each affected host for the incident record.

Step 4: Recovery — After patching, validate Chrome version across all endpoints using your configuration management tool or EDR fleet query. Monitor process and network telemetry on recently patched endpoints for 48-72 hours for residual indicators of prior compromise (T1055, T1068). Review AU-6 audit records for any anomalous activity during the exposure window. Confirm host-based firewall rules applied in Step 1 are reverted or made permanent as appropriate (NIST AC-4).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AC-4 (Information Flow Enforcement), NIST CM-6 (Configuration Settings), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Run a post-patch osquery sweep to confirm version compliance: `SELECT name, version FROM programs WHERE name LIKE '%Chrome%' AND version < '148.0.7778.96'` — any non-empty result set identifies hosts requiring re-remediation. For 48-72 hour residual monitoring without EDR, configure Sysmon Event ID 1 alerts for unusual processes running under user accounts that browsed the web during the exposure window (compare process baseline from before the incident). Write a Sigma rule targeting post-exploitation persistence: detect new scheduled tasks (`schtasks /query`) or autorun registry key modifications (`HKCU\Software\Microsoft\Windows\CurrentVersion\Run`) created by any process in the Chrome PID tree during the exposure window. Use ClamAV with a YARA rule for known Chrome exploit shellcode patterns as a low-cost alternative to commercial malware scanning.

Evidence: Audit Windows Security Event Log for Event ID 4698 (Scheduled Task Created) and Event ID 4657 (Registry Value Modified) on affected hosts during the exposure window — a successful sandbox escape from CVE-2026-2441 exploitation would likely establish persistence via one of these mechanisms before lateral movement. Pull Chrome's `%LOCALAPPDATA%\Google\Chrome\User Data\Default\History` SQLite database and query `SELECT url, last_visit_time FROM urls WHERE last_visit_time > [exposure_window_start]` to reconstruct which URLs were visited on hosts that showed anomalous indicators, identifying potential exploit delivery infrastructure.

Step 5: Post-Incident — Evaluate your patch cycle SLA against this timeline: Google released an out-of-band emergency update — your process should have a documented emergency patching track separate from monthly cadence (CIS 7.2, NIST IR-4). Audit browser management controls: are all Chrome instances centrally managed and enrolled in auto-update policies? Review CIS 2.2 compliance — confirm only supported, current browser versions are authorized. Document any gap between patch release and full fleet deployment as a remediation metric.

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 SI-2 (Flaw Remediation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Document the time delta between Google's May 27, 2026 advisory publication and your fleet reaching 100% patch coverage — this metric directly tests your emergency patching SLA. If no emergency track exists, draft a one-page policy addendum defining criteria for out-of-band patching (CVSS ≥ 9.0 + active exploitation confirmed = 24-hour patch SLA). Use Google's Chrome Browser Cloud Management (free tier) or GPO-based `TargetVersionPrefix` registry policy (`HKLM\SOFTWARE\Policies\Google\Chrome\TargetVersionPrefix`) to enforce minimum version floors and prevent Chrome from running below a defined version. Schedule a lessons-learned meeting within 5 business days per NIST 800-61r3 §4 guidance and record action items against named owners.

Evidence: Compile the full exposure window timeline: earliest confirmed Chrome version below 148.0.7778.96 in asset inventory → patch release timestamp (May 27, 2026) → last endpoint confirmed patched. Cross-reference proxy logs from the exposure window against threat intelligence feeds (CISA KEV, VirusTotal, URLhaus) to determine whether any enterprise endpoints contacted known exploit delivery infrastructure associated with CVE-2026-2441 campaigns — this determines whether post-incident is a lessons-learned exercise or an active breach investigation requiring escalation.

Detection Guidance

Primary detection focus: identify exploitation attempts (T1189) and post-exploitation activity (T1055, T1068, T1059) on endpoints running pre-patch Chrome. Log sources to query: EDR process telemetry, Windows Security Event Log (Event IDs 4688 for process creation, 4663 for object access), Sysmon (Event ID 1 for process creation, Event ID 8 for CreateRemoteThread), proxy/web gateway logs, and DNS query logs. Behavioral indicators: (1) Chrome renderer process (chrome.exe or chrome) spawning unexpected child processes such as cmd.exe, powershell.exe, wscript.exe, or mshta.exe, flag immediately as high-confidence exploitation indicator. (2) Renderer process opening handles to other processes (cross-process injection, T1055). (3) Network connections from Chrome to non-CDN, newly registered, or high-entropy domains delivering JavaScript, WebAssembly, or binary payloads. (4) Browser process writing executables or scripts to disk outside standard Chrome profile/cache directories. (5) Privilege escalation events (T1068) originating from a browser process. Apply D3-LAM (Local Account Monitoring) for any account activity initiated from browser process chains. Apply D3-SFA (System File Analysis) to monitor Chrome binary and profile directories for unexpected modification. For threat hunting: search for Chrome processes with unusual parent-child relationships, memory anomalies (reflective loading patterns), or outbound connections to low-reputation infrastructure during the active exploitation window (prior to patch deployment). NIST AU-6 mandates regular review of audit records, prioritize endpoints in the unpatched window for immediate log review.

Framework Mappings

MITRE-ATTACK

- **T1189** — Drive-by Compromise
- **T1055** — Process Injection
- **T1203** — Exploitation for Client Execution
- **T1059** — Command and Scripting Interpreter
- **T1068** — Exploitation for Privilege Escalation

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-2** — Flaw Remediation
- **CM-7** — Least Functionality
- **SI-7** — Software, Firmware, and Information Integrity

- **SI-16** — Memory Protection
- **SI-10** — Information Input Validation
- **IR-5** — Incident Monitoring

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

OWASP-TOP10-2021

- **A03:2021** — Injection

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1189	Drive-by Compromise	Initial-Access
T1055	Process Injection	Defense-Evasion
T1203	Exploitation for Client Execution	Execution
T1059	Command and Scripting Interpreter	Execution
T1068	Exploitation for Privilege Escalation	Privilege-Escalation

Sources

Source	URL	Tier
Google Chrome Releases	http://chromereleases.googleblog.com/2026/05/stable-channel-update-...	T3
	https://cybersecuritynews.com/chrome148-vulnerabilities-patched/	T3
	https://thecyberexpress.com/cve-2026-2441-google-chrome/	T3
	https://www.forbes.com/sites/zakdoffman/2025/12/10/google-issues-em...	T3
CVE-2026-2441 Detail - NVD	https://nvd.nist.gov/vuln/detail/cve-2026-2441	T1

Source	URL	Tier
Google Security Advisory	https://chromereleases.googleblog.com/	**T1**

Generated 2026-05-27 18:55 UTC by TJS Security Command Center