INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-22 13:41 UTC

Mozilla Firefox 150 Patches 41 Security Vulnerabilities Including High-Severity RCE Flaws

SECURITY ANALYSIS | HIGH | CVSS 8.8

SCC Item ID	SCC-STY-2026-0074
Type	Security Analysis
Severity	HIGH
CVSS Base Score	8.8
Affected Products	Mozilla Firefox < 150
Published	4 hours ago
Discovery Source	Serper

Executive Summary

Mozilla released Firefox 150 addressing 41 security vulnerabilities, several rated high-severity with remote code execution potential, representing a significant patch cycle for the browser. According to third-party reporting, Mozilla used Anthropic's Mythos AI-assisted bug-finding tool to identify approximately 271 potential vulnerabilities across the Firefox codebase, suggesting that additional remediation may be staged for subsequent releases. For organizations running Firefox at scale, this release signals both an immediate patching obligation and a broader shift in how AI-assisted tooling is changing the volume and velocity of vulnerability discovery.

Technical Analysis

Mozilla Security Advisory MFSA2026-30 documents 41 vulnerabilities addressed in Firefox 150, with multiple high-severity findings carrying remote code execution potential. The MITRE technique mapping to T1190 (Exploit Public-Facing Application) and T1203 (Exploitation for Client Execution) frames the attack surface clearly: a threat actor exploiting these flaws could compromise an unpatched browser through a malicious webpage or embedded content, executing arbitrary code in the context of the logged-in user without any additional authentication bypass required.

The more analytically significant dimension of this release is the AI tooling disclosure. According to secondary reporting, Mozilla's use of Anthropic's Mythos platform identified approximately 271 potential vulnerabilities in Firefox's codebase, a figure nearly seven times larger than the 41 addressed in the official advisory. This discrepancy is not unusual in large remediation cycles; vendors routinely batch, triage, and stage fixes across

multiple releases, but the scale here warrants attention. Security teams should not interpret 'Firefox 150 patches 41 flaws' as the complete picture; additional fixes may be staged for subsequent releases. This claim should be verified against official Mozilla announcements.

No CVSS base scores have been officially published by NVD at this time; full technical details including severity scoring should be retrieved directly from MFSA2026-30 at Mozilla's advisory portal. No CVE identifiers, CWE mappings, or EPSS scores are available in the provided source data. Exploitation status is currently unknown. No entries appear in the CISA Known Exploited Vulnerabilities catalog or VulnCheck KEV for these findings as of this writing, but the absence of KEV listing does not indicate low exploitation risk for high-severity RCE-class browser vulnerabilities. Historically, browser vulnerabilities of this severity class have seen weaponized exploitation within days to weeks of public advisory release.

For enterprise environments, browser vulnerabilities in this severity class are a persistent challenge. Endpoint management solutions vary in patch propagation speed, and users running unmanaged or personal Firefox installations represent a gap that centralized patching cannot close. The AI-assisted discovery angle also raises a forward-looking implication: if tooling like Mythos is being applied systematically, patch volumes across the ecosystem may increase materially, accelerating the cadence pressure on vulnerability management programs.

Action Checklist

1. Step 1: Assess exposure, identify all Firefox installations across managed endpoints, including ESR builds, development workstations, and any embedded or kiosk deployments running Firefox as a browser engine.
2. Step 2: Deploy the update, push Firefox 150 through your endpoint management platform (Intune, SCCM, Jamf, or equivalent) immediately; verify enforcement is not advisory-only and confirm version telemetry shows 150 across the fleet.
3. Step 3: Address unmanaged devices, notify employees running personal Firefox on corporate-connected devices and require confirmation of update; consider temporary enforcement of browser version checks via NAC or endpoint compliance policy.
4. Step 4: Review controls for T1203 (client-side exploitation), confirm EDR coverage is active on all endpoints where Firefox is deployed; verify that web content filtering or DNS-layer controls are tuned to block known malicious redirect chains that typically precede browser exploit delivery.
5. Step 5: Monitor the advisory for staged CVE disclosure, bookmark MFSA2026-30 and track subsequent Firefox releases; if additional vulnerabilities from the Mythos-assisted discovery are staged across multiple releases, expect additional high-severity patches in the near term.
6. Step 6: Communicate findings, brief relevant stakeholders on the patching timeline and the AI-assisted discovery context; frame this as both an immediate operational task and a signal to revisit browser patch SLA targets given accelerating discovery velocity.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to CISO and activate IR plan if any endpoint shows Firefox process spawning an unexpected child process (Sysmon Event ID 1 / Windows Security Event ID 4688 with firefox.exe as parent), if DNS or proxy logs show outbound connections to newly-registered or uncategorized domains from a Firefox process during the pre-patch exposure window, or if the affected environment processes PII, PHI, or PCI-scoped data — any of which triggers potential breach notification obligations under HIPAA, GDPR, or PCI DSS given a CVSS 8.8 RCE-class vulnerability with confirmed exposure.
Recovery Notes	After confirming Firefox 150 deployment across all endpoints, validate browser integrity by checking that no unauthorized extensions were silently installed during the exposure window — compare current extension inventory (`%APPDATA%\Mozilla\Firefox\Profiles\*.default\extensions.json`) against a known-good baseline, as browser RCE payloads frequently establish persistence via malicious extension injection or modification of Firefox user preferences (`user.js` or `prefs.js` in the profile directory). Monitor EDR and DNS telemetry for at least 14 days post-patch for delayed beacon activity, as exploit kit payloads delivered via T1203 browser exploitation sometimes include staged downloaders with delayed C2 check-in intervals designed to outlast short-window monitoring. If no EDR is present, run a weekly osquery sweep for the 30-day post-patch period querying for new browser extension installations and unexpected Firefox profile directory modifications.
Forensic Artifacts	Firefox crash reports and minidumps at %APPDATA%\Mozilla\Firefox\Crash Reports\submitted\ (Windows) or ~/Library/Application Support/Firefox/Crash Reports/ (macOS) — heap corruption and use-after-free exploits targeting the RCE-class vulnerabilities in MFSA2026-30 frequently generate .dmp files before achieving stable shellcode execution, providing evidence of exploit attempts even when exploitation failed | Firefox profile directory artifacts — specifically extensions.json, user.js, and prefs.js at %APPDATA%\Mozilla\Firefox\Profiles\\ — post-exploitation persistence via browser RCE commonly involves injecting a malicious extension or modifying security preferences (e.g., disabling certificate validation) to facilitate follow-on activity | Windows Security Event Log Event ID 4688 (Process Creation) and Sysmon Event ID 1 filtered for firefox.exe as ParentImage — successful RCE exploitation of a browser memory vulnerability via T1203 manifests as Firefox spawning cmd.exe, powershell.exe, rundll32.exe, or wscript.exe, which is anomalous and not produced by normal browser operation | Proxy and DNS logs covering the pre-patch exposure window — browser exploit kit delivery chains (the most common delivery mechanism for Firefox RCE exploitation) produce a characteristic pattern of HTTP 302 redirect hops through intermediary domains to the exploit host, followed by a payload download; filter proxy logs for requests with Referer chains exceeding 3 hops and DNS logs for queries to domains with registration age under 30 days from Firefox process context | Firefox HTTP cache and session restore files at %APPDATA%\Mozilla\Firefox\Profiles\\cache2\ and sessionstore-backups\ — these preserve the URL and content of pages visited during the exploitation window, enabling reconstruction of the malicious page or redirect chain that delivered the exploit payload, and are not cleared by a browser restart or update

Per-Action IR Details

Step 1: Assess exposure — identify all Firefox installations across managed endpoints, including ESR builds, development workstations, and any embedded or kiosk deployments running Firefox as a browser engine.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: asset inventory and exposure baseline before active incident

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-5 (Vulnerability Monitoring and Scanning), 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), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Run `wmic product where "name like 'Mozilla Firefox%'" get name,version /format:csv > firefox_inventory.csv` across Windows endpoints via PSExec or a scheduled task pushed through Group Policy. On macOS, use `sudo find /Applications -name 'Firefox.app' -maxdepth 2 -exec defaults read {}/Contents/Info.plist CFBundleShortVersionString \; 2>/dev/null`. For Linux, run `dpkg -l firefox* 2>/dev/null || rpm -qa firefox*` via SSH loop script. Pipe all results to a central share for manual triage. Cross-reference against AD-joined machine list to identify gap (unmanaged or BYOD). For kiosk/embedded: physically or remotely check about:version via Firefox CLI flag `firefox --version`.

Evidence: Before remediating, capture a software inventory snapshot showing current Firefox version per host (hostname, OS, version, last-seen timestamp) — this establishes the pre-patch exposure window for any post-incident timeline reconstruction. If breach investigation later surfaces a client-side exploit (T1203) delivered during this window, the version snapshot proves which hosts were vulnerable and for how long. Export from Intune/SCCM device compliance reports or osquery: `SELECT name, version, install_date FROM programs WHERE name LIKE '%Firefox%';`

Step 2: Deploy the update — push Firefox 150 through your endpoint management platform (Intune, SCCM, Jamf, or equivalent) immediately; verify enforcement is not advisory-only and confirm version telemetry shows 150 across the fleet.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: eliminating the vulnerability from the environment through verified patching

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 7.2 (Establish and Maintain a Remediation Process)

Compensating: Without enterprise patch management, use Mozilla's MSI installer with silent deployment: `msiexec /i Firefox_Setup_150.0.msi /quiet /norestart`. Script a post-install version check: `"C:\Program Files\Mozilla Firefox\firefox.exe" --version | findstr "150"` and log result to a network share with hostname prepended. On macOS, use `brew upgrade --cask firefox` or deploy the PKG via `installer -pkg Firefox\ 150.pkg -target /`. Verify with `osquery` query: `SELECT version FROM programs WHERE name = 'Mozilla Firefox' AND version < '150.0';` — any result indicates a non-compliant host. Schedule this query as a pack to run every 6 hours for 72 hours post-deployment.

Evidence: Before pushing the patch, capture Firefox process telemetry on a representative sample of endpoints: running processes, open network connections from the Firefox process (netstat -b or `ss -tp`), and any active browser extension IDs from the Firefox profile directory (`%APPDATA%\Mozilla\Firefox\Profiles\*.default\extensions.json` on Windows, `~/Library/Application Support/Firefox/Profiles/` on macOS). If an exploit was already delivered pre-patch, this preserves runtime state. Also snapshot Firefox crash reports at `%APPDATA%\Mozilla\Firefox\Crash Reports\submitted\` — RCE exploit attempts against memory corruption vulnerabilities (the mechanism for the high-severity flaws in MFSA2026-30) frequently generate crash telemetry before successful exploitation.

Step 3: Address unmanaged devices — notify employees running personal Firefox on corporate-connected devices and require confirmation of update; consider temporary enforcement of browser version checks via NAC or endpoint compliance policy.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: isolating or restricting vulnerable systems to limit attack surface while remediation proceeds

Controls: NIST AC-17 (Remote Access), NIST AC-20 (Use of External Systems), NIST IR-4 (Incident Handling), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.4 (Require MFA for Remote Network Access), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Without a NAC appliance, use an interim DNS-layer block: configure your internal DNS resolver (e.g., Pi-hole or Windows DNS) to return NXDOMAIN for known malicious redirect domains associated with browser exploit kit delivery (reference ET OPEN or abuse.ch URLhaus feeds filtered for browser exploits). Alternatively, push a browser-agnostic GPO that restricts outbound access from endpoints not meeting a compliance tag. For BYOD attestation without tooling, deploy a lightweight osquery agent (`osquery` is free, cross-platform) with the query `SELECT version FROM programs WHERE name LIKE '%Firefox%'` and collect results via `osqueryd` log shipping to a shared folder — flag any result below 150.0 as non-compliant and block network segment access via VLAN reassignment.

Evidence: Before enforcing NAC/compliance checks, capture the list of unmanaged devices currently on the corporate network segment (ARP table exports, DHCP lease logs, switch MAC address tables) alongside their last-seen timestamps. For any device that connected to the corporate network while running Firefox < 150, preserve network flow data (NetFlow/sFlow exports or Windows Firewall log at `%SystemRoot%\System32\LogFiles\Firewall\pfirewall.log`) covering the exposure window. Browser exploit delivery via T1203 typically involves an outbound HTTP/S request to an attacker-controlled domain followed by a payload download — these flows are your primary forensic indicator if exploitation occurred before containment.

Step 4: Review controls for T1203 (client-side exploitation) — confirm EDR coverage is active on all endpoints where Firefox is deployed; verify that web content filtering or DNS-layer controls are tuned to block known malicious redirect chains that typically precede browser exploit delivery.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: validating detection coverage and tuning controls to identify exploitation attempts against vulnerable Firefox versions

Controls: NIST SI-4 (System Monitoring), NIST SI-3 (Malicious Code Protection), NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 8.2 (Collect Audit Logs), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Without EDR, deploy Sysmon (free, Microsoft Sysinternals) with a configuration that captures Event ID 1 (Process Creation) filtering on `firefox.exe` as parent process spawning unexpected child processes (cmd.exe, powershell.exe, wscript.exe, mshta.exe) — this is the canonical post-exploitation signal for a successful browser RCE via T1203. Use the SwiftOnSecurity Sysmon config as a baseline and add a specific rule: `firefox.exe`. For DNS-layer coverage without a commercial tool, configure Pi-hole with the URLhaus malicious domains list (updated daily via blocklist URL). For network-layer detection, write a Snort/Suricata rule matching HTTP responses containing known browser exploit kit URI patterns (e.g., heavily obfuscated JavaScript with `eval(unescape(` or `String.fromCharCode(` patterns in content). Reference Sigma rule `proc_creation_win_susp_browser_child_process.yml` from the SigmaHQ repository for SIEM-free log hunting via PowerShell: `Get-WinEvent -LogName Security | Where-Object {$_.Id -eq 4688 -and $_.Message -match 'firefox' -and $_.Message -match 'cmd.exe'}`.

Evidence: Query Windows Security Event Log for Event ID 4688 (Process Creation) filtering on processes with `firefox.exe` as the parent process — any child process spawn (especially cmd.exe, powershell.exe, rundll32.exe, or regsvr32.exe) is a high-fidelity indicator of successful RCE exploitation of one of MFSA2026-30's memory corruption or use-after-free vulnerabilities. Also collect: proxy/web filter logs for HTTP 302 redirect chains to domains not in Alexa/Cisco Umbrella top-1M (characteristic of exploit kit redirect chains preceding browser exploit delivery); DNS query logs for newly-registered domains queried from Firefox process context; and Firefox crash reports at `%APPDATA%\Mozilla\Firefox\Crash Reports\` — heap corruption exploits targeting browser memory vulnerabilities often generate minidump files (`.dmp`) before achieving stable code execution.

Step 5: Monitor the advisory for staged CVE disclosure — bookmark MFSA2026-30 and track subsequent Firefox releases; if Mozilla's Mythos-surfaced 271 findings are being staged across multiple releases, additional high-severity patches may follow in the near term.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: integrating threat intelligence from this advisory cycle into ongoing detection and patch SLA processes

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST IR-5 (Incident Monitoring), NIST RA-5 (Vulnerability Monitoring and Scanning), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Subscribe to Mozilla's security advisories RSS feed at `https://www.mozilla.org/en-US/security/advisories/` and route new MFSA entries to a team Slack/Teams channel or shared email alias via an RSS-to-email bridge (free options: IFTTT, rss2email). Set a calendar reminder to check for Firefox 151+ release notes within 72 hours of each Mozilla release, specifically filtering advisory text for memory safety, use-after-free, or heap corruption language — the vulnerability classes most likely to carry RCE potential from the Mythos-identified backlog. Maintain a local tracking spreadsheet correlating MFSA ID, CVE (when assigned), CVSS, and patch deployment date to measure SLA compliance as the staged disclosure unfolds.

Evidence: Preserve the current MFSA2026-30 advisory text and any associated CVE records as they exist today — Mozilla sometimes amends severity ratings or adds CVE assignments after initial publication, and the original snapshot establishes your baseline risk acceptance decision. Also retain the Mythos/271-vulnerabilities disclosure source for your GRC team: if subsequent releases reveal that high-severity RCEs were among the staged findings, this documentation supports any regulatory notification timeline analysis (e.g., demonstrating reasonable response velocity relative to the disclosure timeline).

Step 6: Communicate findings — brief relevant stakeholders on the patching timeline and the AI-assisted discovery context; frame this as both an immediate operational task and a signal to revisit browser patch SLA targets given accelerating discovery velocity.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: lessons learned, policy updates, and stakeholder communication to improve organizational response posture

Controls: NIST IR-6 (Incident Reporting), NIST IR-8 (Incident Response Plan), NIST IR-4 (Incident Handling), NIST PM-15 (Security and Privacy Groups and Associations), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Without a formal GRC platform, draft a one-page stakeholder brief using the following structure: (1) Exposure summary — how many endpoints ran Firefox < 150 and for how long after MFSA2026-30 publication; (2) Patch deployment status — percentage remediated, outstanding exceptions, and deadline; (3) AI-assisted discovery context — note that Mozilla's use of Mythos to surface 271 vulnerabilities signals that traditional patch cadence assumptions (monthly, quarterly) may be insufficient for Firefox going forward; (4) SLA recommendation — propose reducing Firefox patch SLA from current standard (e.g., 30 days for High) to 7 days for any Firefox advisory containing RCE-rated findings. Route to CISO/IT leadership via email with the MFSA2026-30 advisory linked directly.

Evidence: Before closing this patching cycle, compile a post-action metrics package: total endpoints exposed (from Step 1 inventory), time-to-patch per device cohort (managed vs. unmanaged), any exceptions granted and their risk justification, and EDR/DNS-layer alert counts triggered during the exposure window (from Step 4 monitoring). This package serves dual purpose — it satisfies NIST IR-5 (Incident Monitoring) documentation requirements and provides the empirical baseline for the SLA revision argument in the stakeholder brief. Retain for a minimum of one year per NIST AU-11 (Audit Record Retention) guidance.

Detection Guidance

Because no confirmed exploitation activity is documented for these specific vulnerabilities, detection guidance focuses on behavioral patterns consistent with T1203 (client execution via browser exploit) and post-exploitation activity. Review the following:

Endpoint telemetry: Look for Firefox spawning unexpected child processes, particularly cmd.exe, powershell.exe, wscript.exe, or mshta.exe. A browser process should not be a direct parent of shell or scripting processes under normal operation.

Network logs: Monitor for unusual outbound connections originating from firefox.exe, particularly to newly registered domains, IP ranges with no prior organizational history, or connections using non-standard ports. Browser exploit delivery chains frequently involve a redirect sequence ending at a staging server.

Process injection indicators: Watch for cross-process memory writes where firefox.exe is the source process. This pattern can indicate exploit-to-shellcode-to-injection chains.

Patch version telemetry: Query your EDR or asset inventory for any Firefox instances below version 150. Any endpoint still running Firefox < 150 after your patch deployment window closes should be treated as elevated risk and flagged for manual verification.

[UPDATE: Revisit this section once CVE identifiers and CWE mappings are published in MFSA2026-30.] Specific CVE identifiers and associated CWE classes (e.g., memory corruption, use-after-free, type confusion) are not available in the provided source data. Once CVE details are published, refine detection logic to target the specific vulnerability class; heap spray patterns differ meaningfully from use-after-free or integer overflow exploitation paths.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`Pending — refer to Mozilla Security Advisory MFSA2026-30 for published indicators`	CVE identifiers, CWE mappings, and any associated technical indicators for the 41 vulnerabilities addressed in Firefox 150 are published at the Mozilla advisory; specific values were not available in the source data provided for this story.	LOW

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1203** — Exploitation for Client Execution

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring

CIS-V8

- **7.3** — Perform Automated Operating System Patch Management

- **7.4** — Perform Automated Application Patch Management

ISO-27001-2022

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1203	Exploitation for Client Execution	Execution

Sources

Source	URL	Tier
	https://gbhackers.com/mozilla-firefox-150-released/	T3
Security Vulnerabilities fixed in Firefox 150 - Mozilla	https://www.mozilla.org/en-US/security/advisories/mfsa2026-30/	T3
Mozilla Products Multiple Vulnerabilities	https://www.hkcert.org/security-bulletin/mozilla-products-multiple-...	T3
Security Vulnerabilities fixed in Firefox 150 - Update your install!	https://www.reddit.com/r/firefox/comments/1srnhuo/security_vulnerab...	T3
Mozilla Used Anthropic's Mythos to Find and Fix 271 Bugs in Firefox	https://www.facebook.com/Techmeme/posts/mozilla-says-its-firefox-15...	T3

Generated 2026-04-22 13:41 UTC by TJS Security Command Center