INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-30 15:32 UTC

Apple Patches 30+ Vulnerabilities Across iOS, macOS, and Safari Including AI-Discovered WebKit Flaws

CVE VULNERABILITY | HIGH | CVSS 7.5

SCC Item ID	SCC-CVE-2026-0374
Type	CVE Vulnerability
CVE ID	CVE-2026-43707, CVE-2026-43716, CVE-2026-43745, CVE-2026-43715, CVE-2026-43720, CVE-2026-43725, CVE-2026-43722, CVE-2026-43724, CVE-2026-39868
Severity	HIGH
CVSS Base Score	7.5
EPSS Score	0.0016 (6th percentile)
Affected Products	Apple iOS 26.5.2, iPadOS 26.5.2, macOS Tahoe 26.5.2, Safari 26.5.2, WebKit
Published	2026-06-30T03:15:07
Discovery Source	Rss

Executive Summary

Apple has released security updates for iOS, iPadOS, macOS, and Safari addressing more than 30 vulnerabilities, including kernel memory corruption bugs, use-after-free conditions, and a WebKit sandbox escape. No in-the-wild exploitation has been confirmed; however, the presence of a sandbox escape and the breadth of affected Apple platforms elevates urgency for enterprise fleet deployment.

Technical Analysis

Apple's update patches iOS, iPadOS, macOS, and Safari across a set of vulnerabilities including CVE-2026-43707, CVE-2026-43716, CVE-2026-43745, CVE-2026-43715, CVE-2026-43720, CVE-2026-43725, CVE-2026-43722, CVE-2026-43724, and CVE-2026-39868, among others in a 30+ vulnerability batch. Vulnerability classes include: memory safety issues in the kernel (CWE-787 out-of-bounds write, CWE-119 improper restriction of operations within memory bounds), use-after-free conditions in WebKit (CWE-416), out-of-bounds reads (CWE-125), and information disclosure (CWE-200). A WebKit sandbox escape is included in the patch set. No in-the-wild exploitation is confirmed in the provided sources.

Action Checklist

1. Step 1: Containment. Identify all managed Apple devices (iPhones, iPads, Macs, Safari deployments) running versions prior to iOS/iPadOS 26.5.2 and macOS Tahoe 26.5.2. Prioritize internet-facing or remote-access devices. Consider restricting Safari on unpatched endpoints from accessing untrusted web content until patching is complete. Reference Apple Security Advisory at https://support.apple.com/en-us/100100 for authoritative affected version scope (verify that URL resolves to the relevant advisory before relying on it).
2. Step 2: Detection. Query MDM/EDR platforms for device OS versions below 26.5.2 across iOS, iPadOS, and macOS Tahoe fleets (NIST AU-2, CIS 8.2). Review endpoint logs for anomalous WebKit/Safari process behavior, unexpected child process spawning from browser processes, or sandbox policy violations. MITRE T1189 (drive-by) activity may surface as unusual outbound connections from browser processes to unfamiliar domains. No public IOCs were present in the source material; monitor for indicators as they emerge from Apple PSIRT or threat intelligence feeds.
3. Step 3: Eradication. Deploy iOS 26.5.2, iPadOS 26.5.2, macOS Tahoe 26.5.2, and Safari 26.5.2 to all affected devices via MDM push or supervised update channels (CIS 7.3, CIS 7.4, Automated OS and Application Patch Management). Validate update completion through MDM compliance reports. For devices that cannot be immediately updated, restrict Safari usage and enforce content filtering to limit WebKit exposure (NIST AC-4, Information Flow Enforcement).
4. Step 4: Recovery. After patching, run MDM compliance queries to confirm all devices report the target OS/app version. Validate that Safari and WebKit processes are operating normally with no residual crash telemetry or unexpected sandbox policy logs. Re-enable any access restrictions that were applied as interim containment. Monitor CISA KEV and Apple PSIRT for any post-patch exploitation confirmation that would require escalated response (NIST AU-6, Audit Record Review, Analysis, and Reporting).
5. Step 5: Post-Incident. Review the patch deployment timeline against your SLA for high-severity CVEs. The AI-accelerated discovery cadence Apple cited suggests future patch batches may arrive with compressed lead times; validate that your MDM pipeline can sustain rapid fleet updates. Assess whether WebKit-based enterprise applications (internal portals rendered in Safari/WKWebView) require additional WAF or content security policy controls (NIST AC-4, CIS 4.4). Document control gaps exposed by this event, particularly around browser patch velocity and MDM coverage of BYOD devices.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to full incident response and legal/privacy counsel if macOS Unified Log or Safari crash diagnostics on any device reveal successful WebKit sandbox escape indicators (e.g., `com.apple.WebKit.WebContent` spawning a shell or accessing paths outside its sandbox container), particularly on devices with access to PII or PHI, which may trigger breach notification obligations under HIPAA or applicable state privacy laws.

Recovery Notes	After confirming patch deployment via MDM compliance reports, maintain elevated WebKit process monitoring through macOS Unified Log and any available EDR telemetry for a minimum of 14 days, as the presence of a sandbox escape CVE (CVE-2026-43745) means pre-patch exploitation may have deposited persistence mechanisms (e.g., malicious Launch Agents in `~/Library/LaunchAgents/` or modified Safari extensions) that survive the OS update. Validate integrity of Safari extension state post-patch by auditing `~/Library/Safari/Extensions/` and comparing installed extension identifiers against an approved baseline. Re-enable any Safari content restrictions only after all devices in the fleet confirm the target version and no anomalous crash telemetry persists.
Forensic Artifacts	Safari and WebKit crash reports at `~/Library/Logs/DiagnosticReports/com.apple.WebKit.WebContent-*.ips` — direct evidence of memory corruption exploitation attempts targeting CVE-2026-43707 (kernel memory corruption) or the use-after-free CVEs; heap state in these dumps may reveal shellcode or ROP chain fragments. | macOS Unified Log entries for `com.apple.WebKit` and `com.apple.sandbox` subsystems — sandbox policy denial events referencing `com.apple.WebKit.WebContent` indicate attempted sandbox escape activity consistent with CVE-2026-43745. | Safari browsing history and cache database at `~/Library/Safari/History.db` and `~/Library/Caches/com.apple.Safari/` — identify the origin URL of a potential drive-by delivery; WebKit vulnerabilities in this advisory are browser-attack-surface CVEs triggered by visiting a malicious page. | Process spawn records from macOS Endpoint Security or Sysmon showing `com.apple.WebKit.WebContent` as a parent process for any shell, scripting interpreter, or network tool — post-sandbox-escape execution would appear here as an anomalous child process lineage. | Launch Agent and Launch Daemon plist files in `~/Library/LaunchAgents/`, `/Library/LaunchAgents/`, and `/Library/LaunchDaemons/` with modification timestamps falling within the exposure window — a successful sandbox escape followed by privilege escalation via the kernel memory corruption CVEs could install persistence at these locations.

Per-Action IR Details

Step 1: Containment — Identify all managed Apple devices (iPhones, iPads, Macs, Safari deployments) running versions prior to iOS/iPadOS 26.5.2 and macOS Tahoe 26.5.2. Prioritize internet-facing or remote-access devices. Consider restricting Safari on unpatched endpoints from accessing untrusted web content until patching is complete. Reference Apple Security Advisory at https://support.apple.com/en-us/100100 for authoritative affected version scope (verify that URL resolves to the relevant advisory before relying on it).

NIST Phase: Containment

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

Controls: NIST AC-4 (Information Flow Enforcement), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Without MDM, run `system_profiler SPSoftwareDataType` on each Mac via SSH or a looped bash script across the fleet to extract OS build versions. On iOS/iPadOS without MDM enrollment, query Apple Configurator 2 device summary exports. Restrict Safari at the network layer using pfSense or an open-source DNS sinkhole (Pi-hole) to block known malvertising and untrusted TLDs for unpatched devices until the patch window closes.

Evidence: Before restricting Safari or enforcing content filtering on a potentially compromised host, capture: (1) a full Safari browsing history and cache snapshot from `~/Library/Safari/History.db` and `~/Library/Caches/com.apple.Safari/`; (2) active network connections from the Safari/WebKit process using `lsof -i -n -P | grep -i safari` and `netstat -anv`; (3) running process tree with `ps auxww` to detect any unexpected child processes spawned under `com.apple.WebKit.WebContent`. These volatile artifacts will be destroyed when Safari is killed or the host is network-isolated.

Step 2: Detection — Query MDM/EDR platforms for device OS versions below 26.5.2 across iOS, iPadOS, and macOS Tahoe fleets (NIST AU-2, CIS 8.2). Review endpoint logs for anomalous WebKit/Safari process behavior, unexpected child process spawning from browser processes, or sandbox policy violations. MITRE T1189 (drive-by) activity may surface as unusual outbound connections from browser processes to unfamiliar domains. No public IOCs were present in the source material; monitor for indicators as they emerge from Apple PSIRT or threat intelligence feeds.

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs)

Compensating: Without EDR, enable macOS Unified Log collection and filter for WebKit sandbox violations using: `log stream --predicate 'subsystem == "com.apple.WebKit"' --level debug`. Deploy Sysmon for macOS (or use the built-in `EndpointSecurity` framework via a lightweight free agent such as Santa by Google) to detect unexpected process spawns from `com.apple.WebKit.WebContent`. Write a Sigma rule targeting parent-process `WebContent` spawning shells or scripting interpreters (e.g., `sh`, `python3`, `osascript`). Use `osquery` with the query `SELECT pid, name, parent, path FROM processes WHERE parent IN (SELECT pid FROM processes WHERE name LIKE '%WebKit%');` to enumerate suspicious child processes.

Evidence: Before any process termination or host isolation, capture: (1) macOS Unified Log entries for `com.apple.WebKit.WebContent` and `com.apple.sandbox` subsystems covering the 72-hour window preceding detection — export with `log collect --last 72h --output /tmp/webkit_unified.logarchive`; (2) a list of open file handles for the WebKit process using `lsof -p `; (3) kernel extension and sandbox policy denial logs from `/var/log/system.log` and the `kernel` subsystem in Unified Log; (4) a full memory image of the WebKit WebContent process if a use-after-free condition is suspected, using `osxpmem` or `iDump` before killing the process — this is the only opportunity to capture heap state evidence of CVE-2026-43707 or related memory corruption bugs.

Step 3: Eradication — Deploy iOS 26.5.2, iPadOS 26.5.2, macOS Tahoe 26.5.2, and Safari 26.5.2 to all affected devices via MDM push or supervised update channels (CIS 7.3, CIS 7.4 — Automated OS and Application Patch Management). Validate update completion through MDM compliance reports. For devices that cannot be immediately updated, restrict Safari usage and enforce content filtering to limit WebKit exposure (NIST AC-4 — Information Flow Enforcement).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), NIST AC-4 (Information Flow Enforcement)

Compensating: For organizations without enterprise MDM, use Apple Configurator 2 in supervised mode to push the update to physically accessible devices. For remote unmanaged Macs, deploy a lightweight shell script via SSH: `softwareupdate --install --all --restart` and validate with `sw_vers -productVersion` post-reboot. Track completion in a shared spreadsheet against the asset inventory built in Step 1. For BYOD iOS devices not enrolled in MDM, communicate a mandatory self-update deadline with verification via a conditional access check at the VPN or corporate Wi-Fi authentication layer (e.g., block access if the device user-agent reports a Safari version below 26.5.2).

Evidence: Before pushing the patch or reimaging any device suspected of prior compromise (particularly those showing WebKit sandbox violations from Step 2), capture: (1) a full filesystem snapshot or Time Machine backup of the Mac to preserve potential post-exploitation artifacts in `~/Library/WebKit/`, `~/Library/Safari/`, and `/private/var/db/` prior to the OS overwrite; (2) a copy of the Safari crash reporter logs from `~/Library/Logs/DiagnosticReports/` — crash dumps of `com.apple.WebKit.WebContent` are direct forensic evidence of memory corruption exploitation attempts against CVE-2026-43707 or the use-after-free CVEs; (3) `ioreg -l` and `system_profiler SPHardwareDataType` output to establish device identity for chain-of-custody documentation before any state change.

Step 4: Recovery — After patching, run MDM compliance queries to confirm all devices report the target OS/app version. Validate that Safari and WebKit processes are operating normally with no residual crash telemetry or unexpected sandbox policy logs. Re-enable any access restrictions that were applied as interim

containment. Monitor CISA KEV and Apple PSIRT for any post-patch exploitation confirmation that would require escalated response (NIST AU-6 — Audit Record Review, Analysis, and Reporting).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting)

Compensating: Without automated MDM compliance dashboards, build a simple osquery pack that runs `SELECT name, version FROM apps WHERE name = 'Safari';` and `SELECT * FROM os_version;` across enrolled Macs on a scheduled basis, exporting results to a CSV for manual review. Set a cron job to re-run daily for 14 days post-patch to catch any devices that missed the update due to being offline. Subscribe to Apple PSIRT RSS (`https://support.apple.com/en-us/111900` — verify this resolves before use) and CISA KEV JSON feed (`https://www.cisa.gov/sites/default/files/feeds/known_exploited_vulnerabilities.json`) for automated alerting on CVE status changes.

Evidence: Before re-enabling Safari for previously restricted devices, verify: (1) absence of new crash reports in `~/Library/Logs/DiagnosticReports/` for `com.apple.WebKit.WebContent` post-patch; (2) macOS Unified Log shows no sandbox policy denials from `com.apple.WebKit` subsystem in the post-patch window — run `log show --predicate 'subsystem == "com.apple.sandbox" AND eventMessage CONTAINS "WebContent"' --last 24h`; (3) MDM compliance report confirms reported build version matches the expected iOS/iPadOS/macOS Tahoe 26.5.2 build string.

Step 5: Post-Incident — Review the patch deployment timeline against your SLA for high-severity CVEs. The AI-accelerated discovery cadence Apple cited suggests future patch batches may arrive with compressed lead times; validate that your MDM pipeline can sustain rapid fleet updates. Assess whether WebKit-based enterprise applications (internal portals rendered in Safari/WKWebView) require additional WAF or content security policy controls (NIST AC-4, CIS 4.4). Document control gaps exposed by this event, particularly around browser patch velocity and MDM coverage of BYOD devices.

NIST Phase: Post Incident

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

Controls: NIST AC-4 (Information Flow Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For teams without a WAF budget, implement Content Security Policy (CSP) headers on internal WKWebView-rendered portals using Apache `mod_headers` or nginx `add_header` directives to restrict script sources and block inline execution — this directly limits WebKit sandbox escape post-exploitation lateral movement. Document MDM BYOD coverage gaps using the osquery asset query from Step 4 cross-referenced against your Active Directory or IdP device registration list to identify unmanaged Apple devices authenticating to corporate resources.

Evidence: For the lessons-learned record, preserve: (1) the full MDM compliance timeline report showing per-device patch completion timestamps against the advisory release date, to measure actual vs. SLA-defined patch velocity; (2) all WebKit crash reports (`~/Library/Logs/DiagnosticReports/com.apple.WebKit.WebContent-*.ips`) collected during the exposure window — even absent confirmed exploitation, these establish a baseline for future anomaly comparison; (3) the interim Safari restriction access logs from your DNS sinkhole or firewall, documenting which unpatched devices attempted to access external web content during the containment window.

Detection Guidance

No public IOCs (IPs, domains, hashes) were present in the provided source material for this vulnerability set. Detection should focus on behavioral and version-based signals. (1) MDM version compliance: query all managed Apple devices for OS/app versions below iOS/iPadOS 26.5.2 and macOS Tahoe 26.5.2; any unpatched device is an exposure condition, not a confirmed compromise indicator. (2) WebKit/Safari process anomalies: look for Safari or WebKit-based processes spawning unexpected child processes, making outbound

network connections outside normal browsing patterns, or triggering macOS sandbox denial events in system logs (unified log: 'subsystem:com.apple.sandbox' with deny actions). (3) Kernel memory anomalies: monitor for unexpected kernel panics, KEXT crash reports, or kernel_task anomalies that may indicate CWE-787/CWE-119 exploitation attempts. (4) Privilege escalation signals: MITRE T1068 activity may surface as processes running at elevated privilege levels without corresponding user authorization events in audit logs. (5) NIST AU-2 and CIS 8.2 compliance: ensure audit logging is enabled on macOS endpoints to capture the events above. If Apple PSIRT releases IOCs or exploitation signatures following this advisory, ingest them into SIEM/EDR immediately.

Framework Mappings

MITRE-ATTACK

- **T1189** — Drive-by Compromise
- **T1082** — System Information Discovery
- **T1083** — File and Directory Discovery
- **T1211** — Exploitation for Defense Evasion
- **T1068** — Exploitation for Privilege Escalation
- **T1203** — Exploitation for Client Execution

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-16** — Memory Protection
- **SI-10** — Information Input Validation
- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest

OWASP-TOP10-2021

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

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

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control

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
T1082	System Information Discovery	Discovery
T1083	File and Directory Discovery	Discovery
T1211	Exploitation for Defense Evasion	Defense-Evasion
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1203	Exploitation for Client Execution	Execution

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/06/apple-patches-30-ios-macos-safari...	T2
macOS 26.x < 26.5.2 Multiple Vulnerabilities (127595) | Tenable®	https://www.tenable.com/plugins/nessus/323673	T1
May 2026 CVE Landscape - Recorded Future	https://www.recordedfuture.com/blog/may-2026-cve-landscape	T1
Apple Security Advisory	https://support.apple.com/en-us/100100	T1
CVE-2026-43707 - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-43707	T1

Generated 2026-06-30 15:32 UTC by TJS Security Command Center