INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-21 13:56 UTC

LiteSpeed cPanel Plugin Privilege Escalation Vulnerability (CVE-2026-48172), Actively Exploited

CVE VULNERABILITY | **CRITICAL** | CVSS 9.8 | **CISA KEV**

SCC Item ID	SCC-CVE-2026-0207
Type	CVE Vulnerability
CVE ID	CVE-2026-48172
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0004 (12th percentile)
KEV Status	Yes — CISA Known Exploited Vulnerability
Affected Products	LiteSpeed User-End cPanel Plugin before version 2.4.5
Published	2026-05-21T00:00:00Z
Discovery Source	Vulncheck Kev

Executive Summary

A critical privilege escalation vulnerability in the LiteSpeed User-End cPanel Plugin (versions prior to 2.4.5) is being actively exploited in the wild and is listed on the CISA Known Exploited Vulnerabilities catalog. Attackers can exploit the flaw to gain root-level access on affected web hosting servers, enabling full system compromise. Organizations running cPanel-based hosting environments with this plugin installed face immediate risk of complete server takeover, data theft, and service disruption. Confirmed active exploitation detected as of May 2026.

Technical Analysis

CVE-2026-48172 affects LiteSpeed User-End cPanel Plugin versions prior to 2.4.5. The vulnerability stems from incorrect privilege assignment (CWE-266, CWE-269) in the handling of the Redis enable/disable feature via the cpanel_jsonapi_func=redisAble API parameter. Improper privilege checks during this operation allow an authenticated attacker to escalate privileges to root on the host system. CVSS base score: 9.8 (Critical). MITRE ATT&CK techniques: T1068 (Exploitation for Privilege Escalation) and T1548 (Abuse Elevation Control Mechanism). Active exploitation has been confirmed as of May 2026; the vulnerability is listed in the CISA KEV catalog. The LiteSpeed WHM Plugin (parent plugin) is not affected. Patch: upgrade to version 2.4.5 or later.

Vendor detection guidance directs operators to grep cPanel log paths for the redisAble API call to identify exploitation attempts.

Action Checklist

1. Step 1: Patch Immediately. Upgrade all affected servers to LiteSpeed User-End cPanel Plugin version 2.4.5 or later via the cPanel plugin management interface or LiteSpeed vendor portal. Complete patching within 24 hours. If patching cannot be completed immediately, block the cpanel_jsonapi_func=redisAble API parameter at the WAF or firewall layer as interim mitigation. Reference: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management).
2. Step 2: Detect Prior Exploitation. Grep cPanel access and API logs for the string 'cpanel_jsonapi_func=redisAble' per vendor detection guidance. Log paths vary by cPanel configuration but typically reside under /usr/local/cpanel/logs/ and /var/log/. Also review cPanel Web Disk logs under /usr/local/cpanel/logs/ and FTP logs for unauthorized file operations. Any output indicates exploitation activity. Record all source IP addresses, timestamps, and associated cPanel user accounts found. Cross-reference against known-good IP lists. Reference: CIS 8.2 (Collect Audit Logs).
3. Step 3: Eradication. Block or remove unauthorized source IPs identified in Step 2 from firewall and WAF allow-lists. Review system logs for all activity originating from those IPs to assess the full damage scope. Confirm the WHM Plugin version is also current as a hygiene measure (note: it is not reported vulnerable). Reference: NIST AC-3 (Access Enforcement).
4. Step 4: Recovery and Forensics. After patching, re-grep cPanel logs to confirm no new redisAble calls appear. Audit all privileged accounts and SSH authorized_keys files on affected hosts for unauthorized additions (reference: NIST AC-2 Account Management, CIS 5.3 Disable Dormant Accounts). Rotate credentials for any accounts that may have been accessible from compromised hosts (reference: NIST IA-4 Cryptographic Mechanisms). Search for new user accounts with elevated privileges and unexpected cron jobs. Restore from known-good backups only if root-level compromise is confirmed. Monitor system file integrity post-recovery (reference: NIST SI-7 Software, Firmware, and Information Integrity).
5. Step 5: Post-Incident Review. Document the gap that allowed a third-party plugin to hold incorrect privilege assignments and review the plugin vetting process. Map control gaps to NIST AC-6 (Least Privilege) and AC-3 (Access Enforcement). Implement ongoing monitoring for privilege escalation attempts using NIST SI-4 (System Monitoring) and NIST AC-2 (Account Management). Add cpanel_jsonapi_func=redisAble to SIEM detection rules as a persistent indicator. Review plugin update cadence against CIS 7.2 (Establish and Maintain a Remediation Process).

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to senior IR leadership, legal counsel, and data protection officer immediately if forensic analysis confirms root-level access was achieved (uid=0 process spawn or unauthorized SSH key in any account), if customer-hosted data (PII, PHI, PCI-DSS cardholder data) was accessible on the compromised server, or if the team lacks capability to perform memory forensics or rootkit analysis on a confirmed-compromised host — all three conditions are plausible given the CVSS 9.8 score, active CISA KEV listing, and the root-access blast radius of this privilege escalation on a multi-tenant cPanel hosting environment.
Recovery Notes	Do not return any cPanel host to production until plugin version 2.4.5 or later is confirmed installed, chkrootkit and rkhunter scans return clean, all SSH authorized_keys files are audited against a known-good baseline, and a re-grep of /usr/local/cpanel/logs/access_log shows zero new redisAble API calls post-patch. For confirmed root-compromise hosts, prefer full OS rebuild from a clean image over in-place recovery, as attacker-installed rootkits operating at the kernel or bootloader level cannot be reliably detected by userspace tools alone. Monitor cPanel API logs and /var/log/secure continuously for at least 30 days post-recovery for signs of re-exploitation from attacker IPs not captured in initial Step 2 analysis, or lateral movement from compromised cPanel account credentials harvested during the breach window.
Forensic Artifacts	/usr/local/cpanel/logs/access_log and /usr/local/cpanel/logs/api_log — will contain the specific HTTP request(s) invoking cpanel_jsonapi_func=redisAble with authenticated cPanel username, source IP, timestamp, and HTTP 200 response code confirming successful exploitation of CVE-2026-48172 | /var/log/secure (RHEL/CentOS) or /var/log/auth.log (Debian/Ubuntu) — will show the privilege escalation event as a uid change to root (uid=0) or sudo invocation within seconds of the redisAble API hit, providing the precise exploitation timestamp for the incident timeline | SSH authorized_keys files across all accounts under /home/*/. ssh/authorized_keys and /root/.ssh/authorized_keys — attacker-planted public keys with file modification timestamps falling within the confirmed root-access window are primary evidence of persistence establishment following successful CVE-2026-48172 exploitation | /etc/passwd, /etc/shadow, /etc/sudoers, and /etc/sudoers.d/* — newly added system accounts or sudoers entries with creation/modification timestamps post-exploitation indicate the attacker leveraged root access to create durable backdoor accounts beyond SSH key persistence | /tmp/, /var/tmp/, /dev/shm/ directory listings with full timestamps — these world-writable directories are the standard staging locations for web shells, reverse shell binaries, and privilege escalation toolkits dropped by attackers after achieving root via the LiteSpeed cPanel plugin vulnerability

Per-Action IR Details

Step 1: Containment — Immediately identify all servers running LiteSpeed User-End cPanel Plugin versions prior to 2.4.5. Isolate affected hosts from internet-facing exposure where operationally feasible. Block inbound access to the cPanel API endpoint (cpanel_jsonapi_func=redisAble) at the WAF or firewall layer as an interim measure. Reference: CISA KEV advisory and NIST SI-4 (System Monitoring).

NIST Phase: Containment

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

Controls: NIST SI-4 (System Monitoring), NIST AC-4 (Information Flow Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run 'grep -r "lscp" /var/cpanel/plugins/ && cat /var/cpanel/plugins/litespeed/version' on each cPanel host to confirm installed plugin version without an enterprise asset manager. For WAF-less environments, add an iptables rule blocking the cpanel_jsonapi_func=redisAble URI pattern: 'iptables -I INPUT -p tcp --dport 2082 -m string --string "redisAble" --algo bm -j DROP'. For cPanel environments using CSF (ConfigServer Security & Firewall), add

the string match to /etc/csf/csf.conf under CC_DENY or use the CSF UI port block for TCP 2082/2083 from untrusted source ranges.

Evidence: Before isolating, capture a full netstat/ss snapshot ('ss -tnp | grep -E "2082|2083|8080|7080"') to record active connections to the cPanel API and LiteSpeed web server ports. Preserve /usr/local/cpanel/logs/access_log and /usr/local/cpanel/logs/api_log in read-only form (use 'cp --preserve=timestamps') before any log rotation occurs. Document the exact installed version of the LiteSpeed User-End cPanel Plugin by reading /var/cpanel/plugins/litespeed/plugin.json or equivalent manifest file. Note all public IP addresses bound to the affected host — these will anchor the WAF block rule and appear in exploitation logs.

Step 2: Detection — Grep cPanel access and API logs for the string 'cpanel_jsonapi_func=redisAble' per vendor detection guidance. Log paths vary by cPanel configuration but typically reside under /usr/local/cpanel/logs/ and /var/log/. Any output indicates exploitation activity. Record all source IP addresses found. Cross-reference against known-good IP lists using NIST AU-6 (Audit Record Review, Analysis, and Reporting) procedures. Reference: CIS 8.2 (Collect Audit Logs).

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), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Run the following grep chain to extract all exploitation attempts with timestamps and source IPs: 'grep -rh "cpanel_jsonapi_func=redisAble" /usr/local/cpanel/logs/ /var/log/ 2>/dev/null | awk "{print $1, $2, $NF}" | sort | uniq -c | sort -rn > /tmp/litespeed_exploit_hits.txt'. For ongoing monitoring without a SIEM, deploy a one-line inotifywait watch: 'inotifywait -m /usr/local/cpanel/logs/access_log -e modify | while read; do grep --line-buffered "redisAble" /usr/local/cpanel/logs/access_log | tail -1 >> /tmp/litespeed_live_hits.txt; done &'. Use 'last -Fwx' and 'lastlog' to correlate source IPs against recent SSH logins. Write a Sigma rule targeting the cPanel access log source for the 'redisAble' keyword pattern for teams that add log shipping later.

Evidence: Collect /usr/local/cpanel/logs/access_log, /usr/local/cpanel/logs/api_log, and /usr/local/cpanel/logs/error_log — the exploitation attempt against cpanel_jsonapi_func=redisAble will appear as a POST or GET request in the cPanel API access log with the authenticated cPanel username, source IP, and HTTP response code (a 200 response is a strong indicator of successful exploitation). Also collect /var/log/secure (RHEL/CentOS) or /var/log/auth.log (Debian/Ubuntu) to identify privilege escalation to root following a successful API call — look for 'uid=0' or 'sudo' entries within seconds of the redisAble API hit. Capture 'ps auxf' and 'ls -la /proc/*/exe' output to identify any processes spawned by the LiteSpeed plugin process (lscpd or lsphp) running as root post-exploitation.

Step 3: Eradication — Upgrade LiteSpeed User-End cPanel Plugin to version 2.4.5 or later via the cPanel plugin management interface or LiteSpeed vendor portal. Confirm the WHM Plugin version is also current as a hygiene measure (note: it is not reported vulnerable). Block or remove unauthorized source IPs identified in Step 2. Review system logs for all activity originating from those IPs to assess the full damage scope. Reference: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-3 (Configuration Change Control), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Download the 2.4.5 package directly from the LiteSpeed vendor portal and verify its SHA-256 checksum against the vendor-published hash before installation: 'sha256sum litespeed-cpanel-plugin-2.4.5.tar.gz'. Install via 'cd /usr/local/cpanel/whostmgr/docroot/cgi/addon_lsws.cgi' or the cPanel plugin CLI. After upgrade, confirm the installed version with 'cat /var/cpanel/plugins/litespeed/plugin.json | grep version'. For IP blocking without a commercial firewall, add attacker IPs to CSF deny list: 'csf -d "CVE-2026-48172 exploitation attempt"' which persists across reboots. Run 'grep /var/log/secure /var/log/auth.log /usr/local/cpanel/logs/access_log' to enumerate all activity from each blocked IP before removing access.

Evidence: Before patching, preserve a full memory snapshot if root-level compromise is confirmed (use LiME kernel module or 'dd if=/dev/mem' on smaller systems) to capture any in-memory payloads or reverse shells spawned via the privilege escalation. Collect 'find / -newer /tmp/litespeed_exploit_hits.txt -type f -ls 2>/dev/null' scoped to the time window of confirmed exploitation to identify files written by the attacker after gaining root. Document the pre-patch plugin version from /var/cpanel/plugins/litespeed/plugin.json and retain as chain-of-custody evidence. Capture 'crontab -l' for all users and 'ls -la /etc/cron.*/' to identify persistence mechanisms installed by the attacker during the window of root access.

Step 4: Recovery — After patching, re-grep cPanel logs to confirm no new redisAble calls appear. Audit all privileged accounts and SSH authorized_keys files on affected hosts for unauthorized additions (reference: NIST AC-2 Account Management, CIS 5.3 Disable Dormant Accounts). Rotate credentials for any accounts that may have been accessible from compromised hosts (D3-CRO: Credential Rotation). Restore from known-good backups only if root-level compromise is confirmed. Monitor system file integrity post-recovery (D3-SFA: System File Analysis).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AC-2 (Account Management), NIST AC-6 (Least Privilege), NIST CP-10 (System Recovery and Reconstitution), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 5.3 (Disable Dormant Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Enumerate all SSH authorized_keys files across hosting accounts with: 'find /home /root /var/cpanel -name "authorized_keys" -exec ls -la {} \; -exec cat {} \;' — any key added after the earliest confirmed exploitation timestamp is suspect. Check for new system-level accounts with: 'awk -F: "($3 >= 0 && $3 password='. Use 'chkrootkit' (free) and 'rkhunter --check' to scan for rootkits or backdoors installed during the root-access window before declaring recovery complete. Run 'rpm -Va' (RHEL/CentOS) or 'debsums -c' (Debian/Ubuntu) to verify system binary integrity against package database checksums.

Evidence: Before restoring from backup, preserve the full state of /etc/passwd, /etc/shadow, /etc/sudoers, and all files in /etc/sudoers.d/ to document any unauthorized privilege grants made while the attacker held root. Collect all SSH authorized_keys files with timestamps ('find /home /root -name authorized_keys -printf "%T+ %p\n" | sort') as evidence of persistence mechanisms. Run 'last -Fwx' and retain output to document all successful logins — attacker logins via planted SSH keys will appear here. Capture 'ls -la /tmp/ /var/tmp/ /dev/shm/' for dropped payloads, web shells, or staging tools commonly placed in world-writable directories during post-exploitation.

Step 5: Post-Incident — Document the gap that allowed a third-party plugin to hold incorrect privilege assignments and review the plugin vetting process. Map control gaps to NIST AC-6 (Least Privilege) and AC-3 (Access Enforcement). Implement ongoing monitoring for privilege escalation attempts using NIST SI-4 (System Monitoring) and D3-LAM (Local Account Monitoring). Add cpanel_jsonapi_func=redisAble to SIEM detection rules as a persistent indicator. Review plugin update cadence against CIS 7.2 (Establish and Maintain a Remediation Process).

NIST Phase: Post Incident

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

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), NIST SI-4 (System Monitoring), NIST IR-4 (Incident Handling), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For teams without a SIEM, deploy a persistent Sigma-compatible log watch using 'auditd': add a rule 'auditctl -w /usr/local/cpanel/logs/access_log -p wa -k litespeed_cve_2026_48172' to alert on any write to the cPanel access log, then pipe auditd output through 'ausearch -k litespeed_cve_2026_48172' in a cron job checking hourly for the redisAble pattern. Publish a Sigma rule targeting cPanel access logs with detection condition 'cpanel_jsonapi_func=redisAble' for community sharing. Establish a weekly cron job: 'cat /var/cpanel/plugins/litespeed/plugin.json | grep version | mail -s "LiteSpeed Plugin Version Check" security@yourdomain.com' to enforce version awareness without an automated patch management platform. Subscribe to LiteSpeed security advisories via RSS or vendor mailing list as a free early-warning mechanism.

Evidence: Compile a final incident timeline correlating the earliest redisAble log entry, the timestamp of confirmed privilege escalation (from /var/log/secure or auth.log), the duration of attacker root access, and all files and accounts modified during that window — this timeline is required for breach notification assessment if hosted customer data was accessible. Retain all collected log files, memory images, and file system snapshots in a tamper-evident archive (SHA-256 hash each file and store hashes separately) per NIST AU-11 (Audit Record Retention) for a minimum retention period consistent with your organization's incident recordkeeping policy. Document the pre-2.4.5 plugin privilege model as a case study for the plugin vetting process review.

Detection Guidance

Vendor-confirmed detection method: grep cPanel log directories for the string 'cpanel_jsonapi_func=redisAble'. Typical log paths include /usr/local/cpanel/logs/access_log and /var/log/. Also review cPanel Web Disk logs under /usr/local/cpanel/logs/ and FTP logs for unauthorized file operations concurrent with exploitation timestamps. Any match is a positive indicator of exploitation activity; record the timestamp, source IP, and associated cPanel user account. In your SIEM, create a detection rule against cPanel API logs matching the parameter string 'redisAble' with attention to non-administrative source IPs. Correlate source IPs against threat intelligence feeds for known malicious infrastructure. Post-exploitation indicators to hunt: new SSH authorized_keys entries, unexpected cron jobs, new user accounts with elevated privileges, and anomalous outbound connections from the affected host (MITRE T1548, T1068). Reference CIS 8.2 for log collection and NIST SI-7 (Software, Firmware, and Information Integrity) and NIST AC-2 (Account Management) for host-level countermeasures.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`cpanel_jsonapi_func=redisAble`	API parameter string found in cPanel access logs indicating exploitation of CVE-2026-48172; presence of this string in logs is vendor-confirmed evidence of exploitation activity	HIGH

Framework Mappings

MITRE-ATTACK

- **T1068** — Exploitation for Privilege Escalation
- **T1548** — Abuse Elevation Control Mechanism

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **CM-6** — Configuration Settings
- **SI-4** — System Monitoring

OWASP-TOP10-2021

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

CIS-V8

- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **8.2** — Collect Audit Logs

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners
- **CC6.3** — Authorizes, modifies, or removes access

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored
- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1548	Abuse Elevation Control Mechanism	Privilege-Escalation

Sources

Source	URL	Tier
vulncheck_kev	https://nvd.nist.gov/vuln/detail/CVE-2026-48172	T1
Newest CVEs | Tenable®	https://www.tenable.com/cve/newest	T3
CVE-2026-48172 - Info Vulnerability - TheHackerWire	https://www.thehackerwire.com/vulnerability/CVE-2026-48172/	T3
CVE-2026-22172: Openclaw Auth Bypass Vulnerability - SentinelOne	https://www.sentinelone.com/vulnerability-database/cve-2026-22172/	T3

Source	URL	Tier
CVE-2026-27172 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-27172	**T1**
CISA KEV	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	**T1**

Generated 2026-05-21 13:56 UTC by TJS Security Command Center