INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-09 15:02 UTC

CVE-2026-46331: Linux Kernel pedit COW Mechanism Privilege Escalation, Corroboration Unresolved

CVE VULNERABILITY | HIGH

SCC Item ID	SCC-CVE-2026-0403
Type	CVE Vulnerability
CVE ID	CVE-2026-46331
Severity	HIGH
EPSS Score	0.0032 (24th percentile)
Affected Products	Linux kernel (pedit net/sched subsystem, Copy-on-Write mechanism), specific version range unconfirmed
Published	2026-07-09
Discovery Source	Gemini

Executive Summary

A reported Linux kernel privilege escalation vulnerability, CVE-2026-46331, involves a flaw in the pedit packet editor component's Copy-on-Write mechanism that may allow an attacker to corrupt page cache and gain root access. Confidence in this report is low: source descriptions are internally inconsistent, no NVD record or vendor advisory has been confirmed, and the 2026 CVE year prefix is anomalous. Organizations running Linux-based infrastructure should monitor for authoritative advisories before treating this as a confirmed, actionable threat.

Technical Analysis

CVE-2026-46331 is reported to affect the Linux kernel's pedit component within the net/sched subsystem. One source characterizes the issue as a privilege escalation via page cache poisoning of cached binaries (aligned with MITRE T1068, Exploitation for Privilege Escalation); another frames it as a bug fix preventing page cache corruption. These descriptions are materially inconsistent. No CVSS base score, CWE mapping, affected version range, patch identifier, or exploit code has been confirmed from primary sources. EPSS score is 0.00321 (23rd percentile), indicating low current exploitation probability. No CISA KEV listing exists. The CVE year prefix (2026) is forward-dated relative to standard CVE issuance norms and warrants scrutiny. Source URLs citing NVD, Red Hat, SentinelOne, and CVE.org are listed but their content has not been independently confirmed as of this analysis. Do not treat this as a verified vulnerability until a primary advisory is confirmed.

Action Checklist

1. Step 1: Containment, Do not apply patches or configuration changes based solely on this report. Identify all Linux systems running kernels with the net/sched pedit subsystem enabled (check kernel build config for CONFIG_NET_SCH_PEDIT; use 'lsmod | grep sch_pedit' to verify module is loaded, or review full 'lsmod' output for scheduler modules). Isolate high-value Linux hosts from untrusted local users pending advisory confirmation.
2. Step 2: Detection, Monitor for local privilege escalation indicators on Linux hosts: unexpected root process spawns from non-root users, anomalous writes to page cache-backed files (audit syscalls mmap, mprotect, write via auditd or eBPF), and modifications to system binaries. Reference NIST AU-2 (Event Logging) and CIS 8.2 (Collect Audit Logs) for baseline logging requirements. No confirmed IOC patterns are available from primary sources for this CVE.
3. Step 3: Eradication, No confirmed patch, patch ID, or remediation guidance is available from a primary source as of this analysis. Monitor NVD (nvd.nist.gov), Red Hat Security Advisories, and Ubuntu Security Notices for a confirmed fix. When a patch is confirmed, apply via your standard kernel update process per CIS 7.3 (Perform Automated Operating System Patch Management).
4. Step 4: Recovery, Once a confirmed patch is applied, verify kernel version with 'uname -r', validate net/sched module integrity, and review audit logs for any privilege escalation attempts that may have occurred during the exposure window. Align post-patch monitoring with NIST AU-6 (Audit Record Review, Analysis, and Reporting).
5. Step 5: Post-Incident, Review your process for vetting low-confidence CVE reports before escalating to operational response. Assess whether your logging posture (NIST AU-2, AU-12) would detect a local privilege escalation attempt on Linux hosts. Evaluate least-privilege enforcement (NIST AC-6, CIS 5.4) to reduce blast radius if a kernel privilege escalation is confirmed in a future verified advisory.

IR / Forensic Enrichment

Triage Priority	DEFERRED
Escalation Criteria	Escalate to urgent if NVD publishes a confirmed record for CVE-2026-46331 with a CVSS score ≥7.8, a Red Hat or Ubuntu security advisory is issued, a public proof-of-concept targeting the Linux kernel pedit COW mechanism appears on GitHub or Exploit-DB, or if auditd/eBPF monitoring produces confirmed evidence of UID escalation from a non-root process on a monitored Linux host.
Recovery Notes	Recovery should not be declared complete until a vendor-confirmed kernel patch for CVE-2026-46331 is applied and verified via 'uname -r' against the advisory's fixed version, net/sched module integrity is validated, and the full exposure-window auditd log review (targeting mmap/mprotect/write syscalls from non-root users and unexpected SUID binary modifications) yields no evidence of exploitation. Given the low-confidence status of this advisory, maintain the Step 2 auditd monitoring rules and the daily advisory-check cadence for a minimum of 30 days post-patch, or until a confirmed authoritative advisory closes the uncertainty. If the CVE is ultimately unconfirmed or retracted, document the outcome and update the vetting checklist accordingly.

Forensic Artifacts	auditd logs from '/var/log/audit/audit.log' filtered for mmap, mprotect, and write syscalls from non-root UIDs — the pedit COW page-cache corruption mechanism would require these syscalls to manipulate file-backed memory mappings in the kernel net/sched path | '/proc//maps' and '/proc//smaps' snapshots for processes with unexpected capability sets — a successful COW exploit elevating to root would show anomalous transitions from anonymous to file-backed writable mappings in a previously unprivileged process | LiME RAM image of affected host — in-memory page cache state is the primary artifact of a COW corruption attack and is destroyed on reboot, reimage, or kernel patch application; must be captured before any remediation action | SUID binary hash comparison (pre- and post-exposure) using 'find / -perm -4000 -type f -exec sha256sum {} \;' diffed against a known-good baseline — a privilege escalation via kernel COW corruption targeting page cache may manifest as unauthorized modification of SUID binaries such as '/usr/bin/passwd' or '/bin/su' | '/boot/config-$(uname -r)' and 'lsmod' output — documents whether CONFIG_NET_SCH_PEDIT was compiled in and whether the sch_pedit module was loaded at the time of the potential exposure, establishing whether the vulnerable code path was reachable on the specific host

Per-Action IR Details

Step 1: Containment — Do not apply patches or configuration changes based solely on this report. Identify all Linux systems running kernels with the net/sched pedit subsystem enabled (check with 'lsmod | grep sch' and review kernel build config for CONFIG_NET_SCH_PEDIT). Isolate high-value Linux hosts from untrusted local users pending advisory confirmation.

NIST Phase: Containment

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

Controls: NIST AC-6 (Least Privilege), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run 'lsmod | grep sch_ingress\|sch_netem\|cls_u32' across all Linux hosts via a simple SSH loop or Ansible ad-hoc command to enumerate net/sched exposure. For local user isolation on high-value hosts, restrict interactive login via PAM ('pam_access' rules) or set shell to /sbin/nologin for non-essential accounts using 'usermod -s /sbin/nologin '. Catalog results in a spreadsheet tracking kernel version ('uname -r'), CONFIG_NET_SCH_PEDIT state from '/boot/config-$(uname -r)', and local user count per host.

Evidence: Before isolating any host or modifying user access: capture current process tree ('ps auxf > /tmp/proctree_$(hostname)_$(date +%s).txt'), active network connections ('ss -tulnp > /tmp/netconn_$(hostname)_$(date +%s).txt'), and loaded kernel modules ('lsmod > /tmp/lsmod_$(hostname)_$(date +%s).txt'). A COW-based page cache corruption exploit may leave anomalous anonymous memory mappings — capture '/proc//maps' for all non-root processes with unexpected elevated capabilities ('getpcaps ') before any access changes evict live process state.

Step 2: Detection — Monitor for local privilege escalation indicators on Linux hosts: unexpected root process spawns from non-root users, anomalous writes to page cache-backed files (audit syscalls mmap, mprotect, write via auditd or eBPF), and modifications to system binaries. Reference NIST AU-2 (Event Logging) and CIS 8.2 (Collect Audit Logs) for baseline logging requirements. No confirmed IOC patterns are available from primary sources for this CVE.

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy auditd rules targeting the syscalls most relevant to a pedit COW page-cache corruption exploit: '-a always,exit -F arch=b64 -S mmap -S mprotect -S write -F uid!=0 -k pedit_cow_watch'. Use 'ausearch -k pedit_cow_watch' to query results. Supplement with a Sigma rule detecting UID 0 processes whose parent PPID maps to a non-root user (parent_user != root AND process_user == root). If eBPF tooling is available, run 'bpftrace -e

"tracepoint:syscalls:sys_enter_mmap { if (uid != 0) { printf(\"%d %s\\n\", uid, comm); } }'" to observe mmap calls from non-root processes in real time. Compare '/usr/bin' and '/usr/sbin' hash baselines using 'sha256sum' against stored values.

Evidence: No pre-action volatile capture is required for passive monitoring; however, if a suspicious process is identified, capture before any intervention: RAM image using LiME kernel module ('insmod lime.ko path=/mnt/evidence/ram.lime format=lime') to preserve in-memory page cache state that a COW exploit would manipulate. Also collect '/proc//smaps' and '/proc//pagemap' for the suspect process to document anonymous vs. file-backed mappings, and auditd logs from '/var/log/audit/audit.log' filtered for the suspect UID.

Step 3: Eradication — No confirmed patch, patch ID, or remediation guidance is available from a primary source as of this analysis. Monitor NVD (nvd.nist.gov), Red Hat Security Advisories, and Ubuntu Security Notices for a confirmed fix. When a patch is confirmed, apply via your standard kernel update process per CIS 7.3 (Perform Automated Operating System Patch Management).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Until a confirmed kernel patch is published, consider disabling the pedit module if operationally feasible: 'rmmod cls_u32' and blacklist via '/etc/modprobe.d/pedit-block.conf' with 'blacklist sch_pedit'. Verify removal with 'lsmod | grep pedit'. For environments that cannot disable net/sched components, enforce strict local user access controls as a compensating measure. Track advisory sources manually: NVD at nvd.nist.gov for CVE-2026-46331, Red Hat CVE database at access.redhat.com/security/cve/CVE-2026-46331, and Ubuntu at ubuntu.com/security/CVE-2026-46331 — check each at a defined cadence (recommend daily) and document check timestamps.

Evidence: Before applying any confirmed kernel patch (which constitutes a host state change): acquire a full RAM image using LiME if not already captured during detection phase, snapshot '/proc/kallsyms' to baseline kernel symbol addresses, and record the pre-patch kernel version ('uname -a > /tmp/prepatch_kernel_$(hostname).txt'). A COW page cache exploit that achieved persistence may have modified SUID binaries — run 'find / -perm -4000 -type f -exec sha256sum {} \;' and diff against your baseline before patching, as reimaging after patching would destroy evidence of prior compromise.

Step 4: Recovery — Once a confirmed patch is applied, verify kernel version with 'uname -r', validate net/sched module integrity, and review audit logs for any privilege escalation attempts that may have occurred during the exposure window. Align post-patch monitoring with 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), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Post-patch, verify kernel integrity by comparing the running kernel's cryptographic signature against the vendor-signed package: 'rpm -V kernel' (RHEL/CentOS) or 'debsums -c linux-image-$(uname -r)' (Debian/Ubuntu). Rerun the auditd queries from Step 2 across the full exposure window (from earliest possible exploit date to patch application date) using 'ausearch -k pedit_cow_watch --start --end '. Audit SUID binary hashes against the pre-patch baseline to detect any files that a successful COW page-cache corruption attack may have modified. Document findings in a change record.

Evidence: Before completing recovery sign-off, retain the following from the exposure window for potential future forensic use: all auditd logs from '/var/log/audit/' covering the exposure period (compress and archive with timestamps), the pre-patch '/proc/kallsyms' snapshot, any LiME RAM images captured during detection, and the lsmod/process-tree captures from Step 1. These artifacts document whether exploitation of the pedit COW mechanism occurred during the unpatched window and are necessary if a later confirmed advisory changes the severity or exploitation status.

Step 5: Post-Incident — Review your process for vetting low-confidence CVE reports before escalating to operational response. Assess whether your logging posture (NIST AU-2, AU-12) would detect a local privilege escalation attempt on Linux hosts. Evaluate least-privilege enforcement (NIST AC-6, CIS 5.4) to reduce blast radius if a kernel privilege escalation is confirmed in a future verified advisory.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity (Lessons Learned)

Controls: NIST AU-2 (Event Logging), NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Document a one-page vetting checklist for low-confidence kernel CVE reports that includes: NVD record present (yes/no), vendor advisory published (RHEL/Ubuntu/SUSE), PoC code publicly available, and CVE year consistency check (a 2026-prefixed CVE in 2026 warrants NVD cross-check). For logging posture assessment, run 'auditctl -l' on representative Linux hosts to verify mmap/mprotect/write syscall rules are active; if absent, deploy the rules from Step 2 as a permanent baseline. Review '/etc/passwd' and 'getent group sudo wheel' to enumerate accounts with local privilege paths that a pedit COW escalation would exploit.

Evidence: No live volatile state is at risk in the post-incident phase. Retain and archive: the complete timeline of advisory checks performed (NVD, Red Hat, Ubuntu, with timestamps and results), the asset inventory of net/sched-enabled hosts produced in Step 1, and the auditd log review findings from Step 4. These records constitute the evidentiary basis for a lessons-learned report and support any future regulatory inquiry if CVE-2026-46331 is later confirmed as exploited in the wild against Linux kernel net/sched pedit.

Detection Guidance

No confirmed IOC patterns, exploit signatures, or behavioral indicators are available from primary sources for CVE-2026-46331. General detection guidance for local Linux kernel privilege escalation (T1068): (1) Enable auditd syscall monitoring for mmap, mprotect, write, and execve calls from non-root users on sensitive hosts. (2) Monitor for unexpected setuid/setgid process creation or UID transitions to 0 in system logs (/var/log/auth.log, journalctl). (3) Use eBPF-based tools (e.g., Falco, Tetragon) to detect anomalous kernel-level behavior in the net/sched subsystem. (4) Alert on unexpected modifications to system binaries in page-cache-backed paths. Reference NIST AU-2 (Event Logging) and CIS 8.2 (Collect Audit Logs). These are general kernel privilege escalation detection measures, no CVE-specific signatures are confirmed.

Framework Mappings

MITRE-ATTACK

- **T1068** — Exploitation for Privilege Escalation

NIST-800-53R5

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

CIS-V8

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

- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1068	Exploitation for Privilege Escalation	Privilege-Escalation

Sources

Source	URL	Tier
CVE-2026-46331: Linux Kernel Privilege Escalation Flaw	https://www.sentinelone.com/vulnerability-database/cve-2026-46331/	T1
CVE-2026-46331 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-46331	T1
CVE-2026-46331 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-46331	T1
CVE-2026-46331 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-46331	T1
CVE-2026-46331: Linux pedit net/sched Bug Fix Prevents Page ...	https://windowsforum.com/threads/cve-2026-46331-linux-pedit-net-sch...	T3

Generated 2026-07-09 15:02 UTC by TJS Security Command Center