INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-30 19:02 UTC

CVE-2026-31431 'Copy Fail': Nine Years of Linux Kernels Exposed to Root via Crypto Subsystem Bug

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0108
Type	CVE Vulnerability
CVE ID	CVE-2026-31431
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0001 (1th percentile)
Affected Products	Linux kernel 4.14 and later (2017-present); Ubuntu 24.04 LTS, Amazon Linux 2023, RHEL 10.1, SUSE 16, Fedora 42; Kubernetes/container clusters, CI runners, cloud SaaS environments
Published	2026-04-30T09:54:47
Discovery Source	Rss

Executive Summary

A critical privilege escalation vulnerability in the Linux kernel, tracked as CVE-2026-31431 ('Copy Fail'), allows any user with basic local access to gain full administrative (root) control over affected systems. The flaw exists in Linux kernel versions 4.14 and later, meaning virtually every major Linux distribution released since 2017 is affected, including Ubuntu 24.04 LTS, RHEL 10.1, and Amazon Linux 2023. A publicly disclosed exploit raises immediate risk for organizations running shared Linux infrastructure, including Kubernetes clusters, CI/CD pipelines, and multi-tenant cloud environments.

Technical Analysis

CVE-2026-31431 is a logic flaw in the Linux kernel crypto subsystem, affecting authenticated encryption functions within the AEAD (Authenticated Encryption with Associated Data) template. The vulnerability was introduced in kernel 4.14 (2017) and affects all subsequent releases through the present. Associated CWEs are CWE-787 (out-of-bounds write), CWE-269 (improper privilege management), and CWE-667 (improper locking). CVSS base score is reported at 9.5 (Critical); vector string is pending NVD publication. Security researchers have reported a working proof-of-concept exploit; independent verification of reliability claims is recommended. Attack vector is local, an unprivileged user with shell access can escalate to root. No authentication bypass is

required beyond initial local access. MITRE ATT&CK mappings include T1548.001 (Abuse Elevation Control Mechanism: Setuid/Setgid), T1611 (Escape to Host), T1068 (Exploitation for Privilege Escalation), and T1543 (Create or Modify System Process). Upstream kernel patches exist. As of publication, distribution-level advisories have been confirmed for RHEL and Ubuntu; check your vendor's security portal for specific availability. EPSS score is 9e-05 (0.00915th percentile) at time of data capture; this figure reflects pre-PoC disclosure scoring and should be treated as stale. Source quality score is 0.64 (T3 sources dominant); NVD entry (T1) should be consulted to confirm official scoring and vector. All source URLs in this item are unverified in this session; human validation is recommended before citing.

Action Checklist

1. Step 1: Containment, Identify all Linux systems running kernel 4.14 or later across production, staging, and CI/CD environments. Prioritize multi-tenant systems, Kubernetes nodes, and CI runners where unprivileged user access is shared. Restrict interactive shell access to only essential, trusted accounts until patches are applied. On Kubernetes clusters, audit pod security policies and node access controls to limit lateral movement risk.
2. Step 2: Detection, Query kernel versions across your fleet using 'uname -r' output collected via your configuration management tool (Ansible, Chef, Puppet) or cloud provider inventory APIs. On RHEL/CentOS systems, check the Red Hat Customer Portal advisory for CVE-2026-31431 for affected package versions (verify portal URL before use). Review audit logs (auditd) for unexpected privilege escalation events: look for setuid/setgid execution anomalies, unexpected process ownership changes, and kernel subsystem errors in /var/log/kern.log or journalctl -k. No confirmed IOC signatures are available at this time; behavioral detection is the primary indicator.
3. Step 3: Eradication, Apply upstream kernel patches as distributed by each vendor: Check the Red Hat Customer Portal (https://access.redhat.com/security/cve/cve-2026-31431, verify this URL before visiting), Ubuntu Security Notices, SUSE Security Advisories, and Amazon Linux Security Center for distribution-specific packages. If vendor patches are unavailable for your distribution, consult your kernel documentation and security team on temporary mitigation options (e.g., user namespace restrictions, mandatory access control profiles). Upstream patches are preferred. Do not modify crypto subsystem modules without vendor guidance.
4. Step 4: Recovery, After patching, reboot affected systems to load the updated kernel (kernel patches require a reboot to take effect). Confirm the updated kernel version is running via 'uname -r'. Re-validate that privileged account access logs show no anomalous root sessions during the exposure window. Monitor auditd output and SIEM alerts for continued privilege escalation attempts post-patch, which may indicate active exploitation or persistence mechanisms already in place.
5. Step 5: Post-Incident, Conduct a control gap review against NIST SP 800-53 AC-6 (Least Privilege) and SI-2 (Flaw Remediation). Evaluate whether kernel patch cadence SLAs are defined and enforced for your fleet. Assess whether multi-tenant and shared-compute environments have compensating controls (namespace isolation, pod security admission, user namespace restrictions) that would limit the blast radius of a local privilege escalation. Update your vulnerability management program to prioritize local privilege escalation CVEs affecting shared infrastructure at the same urgency level as remote code execution vulnerabilities.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal/compliance immediately if auditd or wtmp evidence shows any successful privilege escalation event (euid=0 from non-root auid) on any multi-tenant system, Kubernetes node, or CI runner during the exposure window, as this may constitute a data breach triggering regulatory notification obligations under GDPR, HIPAA, or applicable state breach notification laws.
Recovery Notes	After patching and rebooting all affected systems, maintain elevated auditd monitoring (specifically the priv_esc_suspicious rules targeting euid=0 from non-root auid) for a minimum of 30 days post-remediation to detect any attacker who established persistence before the patch was applied. On Kubernetes clusters and CI runner environments, re-run pod security admission policy audits weekly for 60 days to confirm no new workloads are introduced that relax namespace or securityContext restrictions that compensated for the vulnerability during the exposure window. Any system where full patch-plus-reboot was not achievable and only compensating controls (AF_ALG module restriction, SSH access lockdown) were applied must remain on a watch list with a hard deadline for full kernel remediation, tracked in your vulnerability management program under CIS 7.2 (Establish and Maintain a Remediation Process).
Forensic Artifacts	auditd SYSCALL records filtered for euid=0 events with non-root auid values (/var/log/audit/audit.log) — the primary behavioral artifact of a successful CVE-2026-31431 local privilege escalation completing, extractable with 'ausearch -m SYSCALL -i | grep -E "euid=root" | grep -v "auid=root|auid=unset"' | Kernel ring buffer output from 'journalctl -k -p err..emerg' covering the exposure window, filtered for crypto subsystem errors (af_alg, skcipher, AEAD, algif) — failed or probing exploitation attempts against the kernel crypto API will surface as kernel errors or warnings before a successful exploit lands | Snapshot of loaded kernel modules via 'lsmod' and /proc/modules captured at time of incident, specifically documenting presence of af_alg, algif_hash, algif_skcipher, algif_aead modules which represent the attack surface exposed by CVE-2026-31431's crypto subsystem flaw | SUID/SGID binary inventory ('find / -perm /6000 -type f -ls') with file modification timestamps compared against patch application date — a post-exploitation persistence mechanism on Linux commonly involves dropping a new SUID binary or modifying an existing one, and any SUID binary with a mtime during the exposure window is a high-priority forensic artifact | wtmp/btmp binary login records exported via 'utmpdump /var/log/wtmp' and 'utmpdump /var/log/btmp', and SSH auth log (/var/log/auth.log or /var/log/secure) covering the full exposure window — these establish a timeline of all root and non-root interactive sessions to correlate against the privilege escalation detection signals and identify which accounts may have leveraged the vulnerability

Per-Action IR Details

Step 1: Containment — Identify all Linux systems running kernel 4.14 or later across production, staging, and CI/CD environments. Prioritize multi-tenant systems, Kubernetes nodes, and CI runners where unprivileged user access is shared. Restrict interactive shell access to only essential, trusted accounts until patches are applied. On Kubernetes clusters, audit pod security policies and node access controls to limit lateral movement risk.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment, Eradication, and Recovery: Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST AC-6 (Least Privilege), NIST CM-8 (System Component Inventory), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Run 'ansible all -m command -a "uname -r" -o' or 'for h in $(cat hosts.txt); do ssh $h uname -r; done' to enumerate kernel versions fleet-wide without a CMDB. Restrict interactive SSH by setting 'AllowUsers' or 'AllowGroups' to a minimal trusted list in /etc/ssh/sshd_config and reloading sshd. For Kubernetes nodes without a policy engine, immediately apply 'kubectl cordon ' on unpatched nodes to prevent new pod scheduling, and review existing pod specs with 'kubectl get pods -A -o jsonpath="{range .items[*]}{.metadata.name} {.spec.securityContext}{\n}{end}"' to identify pods running without securityContext restrictions.

Evidence: Before restricting access, snapshot /etc/passwd, /etc/shadow, /etc/sudoers, and /etc/sudoers.d/* to establish the baseline of privileged accounts existing prior to any exploitation. Capture 'last -F' and 'lastlog' output to record all recent interactive logins. On Kubernetes nodes, export 'kubectl get rolebindings,clusterrolebindings -A -o yaml' to document current RBAC state before any changes. Preserve /proc/kallsyms and the output of 'lsmod' on each node to document loaded kernel modules at time of containment, since CVE-2026-31431 exploitation targets the kernel crypto subsystem and module state is forensically relevant.

Step 2: Detection — Query kernel versions across your fleet using 'uname -r' output collected via your configuration management tool (Ansible, Chef, Puppet) or cloud provider inventory APIs. On RHEL/CentOS systems, check the Red Hat Customer Portal advisory for CVE-2026-31431 for affected package versions. Review audit logs (auditd) for unexpected privilege escalation events: look for setuid/setgid execution anomalies, unexpected process ownership changes, and kernel crypto subsystem errors in /var/log/kern.log or journalctl -k. No confirmed IOC signatures are available at this time — behavioral detection is the primary indicator.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Signs of an Incident and Incident Analysis

Controls: NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy auditd rules targeting privilege escalation patterns specific to a crypto subsystem exploit: 'auditctl -a always,exit -F arch=b64 -S execve -F euid=0 -F auid>=1000 -k priv_esc_suspicious' to catch unprivileged users executing processes that land with euid=0. Search kern.log for crypto subsystem anomalies with 'journalctl -k --since "72 hours ago" | grep -iE "(crypto|af_alg|skcipher|aead|FAULT|BUG:|Call Trace)"'. Use osquery to identify unexpected root-owned processes spawned from non-root parents: 'SELECT pid, parent, name, uid, euid FROM processes WHERE euid=0 AND uid != 0;'. Cross-reference with 'ausearch -m SYSCALL -sv no -i | grep -E "(setuid|setgid|capset)"' for denied or anomalous privilege calls.

Evidence: Collect the full auditd log (/var/log/audit/audit.log) covering the disclosure window (treat kernel 4.14 introduction in 2017 as theoretical exposure start; focus triage on post-public-exploit period). Extract auditd records with 'ausearch -m SYSCALL,EXECVE,PROCTITLE -ts recent' and filter for euid=0 events initiated by non-root auid values, which is the behavioral signature of a local privilege escalation completing successfully. Capture 'journalctl -k -p err..emerg' output to surface kernel-level errors in the crypto subsystem (af_alg socket, skcipher, or AEAD interfaces) that may indicate exploit attempts, even unsuccessful ones. Preserve /proc//maps and /proc//status for any suspicious processes identified, as heap/stack layout artifacts may indicate exploitation technique used.

Step 3: Eradication — Apply upstream kernel patches as distributed by each vendor: check the Red Hat Customer Portal (https://access.redhat.com/security/cve/cve-2026-31431 — validate this URL before use), Ubuntu Security Notices, SUSE Security Advisories, and Amazon Linux Security Center for distribution-specific packages. If vendor patches are not yet available for your distribution, apply the upstream kernel fix directly if your environment supports custom kernel builds, or implement the mitigation of restricting authencesn module loading where operationally feasible (consult your kernel documentation before making module changes in production).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication and Recovery: Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-8 (System Component Inventory), 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: For RHEL 10.1/CentOS, run 'dnf check-update kernel && dnf update kernel' and confirm the patched package version matches the advisory. For Ubuntu 24.04 LTS, run 'apt-get update && apt-get install --only-upgrade linux-image-$(uname -r | cut -d- -f1-2)'. For Amazon Linux 2023, run 'dnf update kernel'. If vendor patches are not yet available, restrict access to the AF_ALG socket interface (the kernel crypto API surface implicated in this vulnerability class) by adding 'install af_alg /bin/true' to /etc/modprobe.d/cve-2026-31431-mitigate.conf and running 'modprobe -r af_alg' if the module is loaded — verify operational impact before applying in production. Verify patch integrity with 'rpm -V kernel' (RHEL) or 'debsums -c linux-image-$(uname -r)' (Ubuntu) after installation.

Evidence: Before applying patches, collect 'rpm -qa kernel' or 'dpkg -l linux-image*' to document the pre-patch kernel package version as a forensic baseline. Capture 'lsmod | grep -E "(af_alg|algif_hash|algif_skcipher|algif_aead|algif_rng)"' to record which crypto subsystem modules were loaded at eradication time — this documents the attack surface that was present. If any system shows evidence of prior exploitation (unexpected root processes, modified binaries), image the disk before patching: 'dd if=/dev/sda | gzip > /forensics/hostname_pre_patch_$(date +%Y%m%d).img.gz' and preserve to offline storage before proceeding with remediation.

Step 4: Recovery — After patching, reboot affected systems to load the updated kernel (kernel patches require a reboot to take effect). Confirm the updated kernel version is running via 'uname -r'. Re-validate that privileged account access logs show no anomalous root sessions during the exposure window. Monitor auditd output and SIEM alerts for continued privilege escalation attempts post-patch, which may indicate active exploitation attempts or persistence mechanisms already in place.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Eradication and Recovery: Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-6 (Security and Privacy Function Verification), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CP-10 (System Recovery and Reconstitution), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 8.2 (Collect Audit Logs)

Compensating: After reboot, run 'uname -r' and compare output against the vendor advisory's fixed version string — document this verification per system. Hunt for persistence mechanisms that a successful pre-patch exploit of CVE-2026-31431 could have installed: check for new SUID/SGID binaries with 'find / -perm /6000 -type f -newer /tmp/patch_timestamp -ls 2>/dev/null', scan cron directories ('ls -la /etc/cron* /var/spool/cron/crontabs/') for entries added during the exposure window, and inspect /etc/ld.so.preload and /etc/ld.so.conf.d/ for injected shared libraries, which are common post-exploitation persistence mechanisms on Linux. Run 'chkrootkit' or 'rkhunter --check --sk' as a lightweight rootkit sweep on recovered systems.

Evidence: Capture 'last -F root' and 'ausearch -ua 0 -ts ' to enumerate all root sessions that occurred between patch announcement and successful remediation — any session not attributable to a known administrative action should be treated as a potential exploitation event. Preserve /var/log/wtmp and /var/log/btmp (binary login records) before and after reboot using 'utmpdump /var/log/wtmp > wtmp_post_patch.txt' for timeline correlation. On Kubernetes nodes, re-run 'kubectl get events -A --field-selector reason=Escalating,reason=Failed' to identify any post-patch policy violations that may indicate an attacker retrying exploitation against the patched kernel.

Step 5: Post-Incident — Conduct a control gap review against NIST SP 800-53 AC-6 (Least Privilege) and SI-2 (Flaw Remediation). Evaluate whether kernel patch cadence SLAs are defined and enforced for your fleet. Assess whether multi-tenant and shared-compute environments have compensating controls (namespace isolation, pod security admission, user namespace restrictions) that would limit the blast radius of a local privilege escalation. Update your vulnerability management program to prioritize local privilege escalation CVEs affecting shared infrastructure at the same urgency level as remote code execution vulnerabilities.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons Learned and Using Collected Incident Data

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST AC-6 (Least Privilege), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Document patch-to-reboot lag per system in a simple spreadsheet (hostname, kernel version pre-patch, patch applied timestamp, reboot timestamp, kernel version post-reboot) — this becomes your SLA baseline for future critical kernel CVEs. For multi-tenant gap assessment, use 'sysctl kernel.unprivileged_userns_clone' and 'sysctl kernel.perf_event_paranoid' to verify that user namespace and performance event restrictions are configured, as these are compensating controls that raise the exploitation bar for local privilege escalation classes like CVE-2026-31431. Draft a one-page lessons-learned memo covering: mean time from CVE disclosure to patch completion, systems that required manual intervention, and which compensating controls (if any) would have prevented exploitation of the crypto subsystem attack surface.

Evidence: Compile a final timeline from auditd, kern.log, wtmp, and SSH auth logs covering the full exposure window from kernel 4.14 deployment date on each system through confirmed patch+reboot, to determine whether any privilege escalation events occurred before remediation was complete. Retain all collected forensic artifacts (disk images, log exports, lsmod snapshots, process trees) for a minimum of 90 days per NIST AU-11 (Audit Record Retention) requirements, or longer if a breach or regulatory notification obligation is triggered. Produce a per-environment remediation completion report noting which systems in Kubernetes clusters, CI runners, and cloud SaaS environments were remediated, which received compensating controls only, and which remain at risk pending vendor patch availability.

Detection Guidance

No confirmed IOC signatures (hashes, IPs, domains) have been publicly attributed to active exploitation of CVE-2026-31431; this assessment may be stale, check CISA KEV and VulnCheck for updates. Detection relies on behavioral and configuration indicators. Query your asset inventory for all Linux systems running kernel versions 4.14 through current. Use auditd rules to flag unexpected privilege escalations: monitor for execve calls resulting in UID/GID changes to 0, unexpected setuid binary execution, and process lineage anomalies where an unprivileged shell spawns a root process. In Kubernetes environments, alert on unexpected host namespace access or container escape indicators (T1611). Review journalctl -k and /var/log/kern.log for kernel subsystem errors or unexpected module load events. SIEM correlation: alert on sequences of unprivileged login followed by root-level process creation within the same session on kernel-affected systems. If your EDR supports kernel-level telemetry, review for anomalous kernel interactions related to cryptographic operations.

Framework Mappings

MITRE-ATTACK

- **T1548.001** — Setuid and Setgid
- **T1611** — Escape to Host
- **T1068** — Exploitation for Privilege Escalation
- **T1543** — Create or Modify System Process

NIST-800-53R5

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

- **SI-16** — Memory Protection
- **SC-13** — Cryptographic Protection

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
- **16.10** — Apply Secure Design Principles in Application Architectures
- **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.5.34** — Privacy and protection of personal information
- **A.5.21** — Managing information security in the ICT supply chain
- **A.8.24** — Use of cryptography
- **A.5.23** — Information security for use of cloud services

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

- **164.312(e)(1)** — Transmission Security

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1548.001	Setuid and Setgid	Privilege-Escalation
T1611	Escape to Host	Privilege-Escalation
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1543	Create or Modify System Process	Persistence

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/new-linux-copy-fail-...	T3
CVE-2026-31431 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-31431	T3

Source	URL	Tier
What we know about Copy Fail (CVE-2026-31431) - Bugcrowd	https://www.bugcrowd.com/blog/what-we-know-about-copy-fail-cve-2026...	T3
New Linux 'Copy Fail' Vulnerability Enables Root Access on Major ...	https://thehackernews.com/2026/04/new-linux-copy-fail-vulnerability...	T3
Cve-2026-31431 medium unpriv to root : r/sysadmin - Reddit	https://www.reddit.com/r/sysadmin/comments/1szaif3/cve202631431_med...	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31431	T1

Generated 2026-04-30 19:02 UTC by TJS Security Command Center