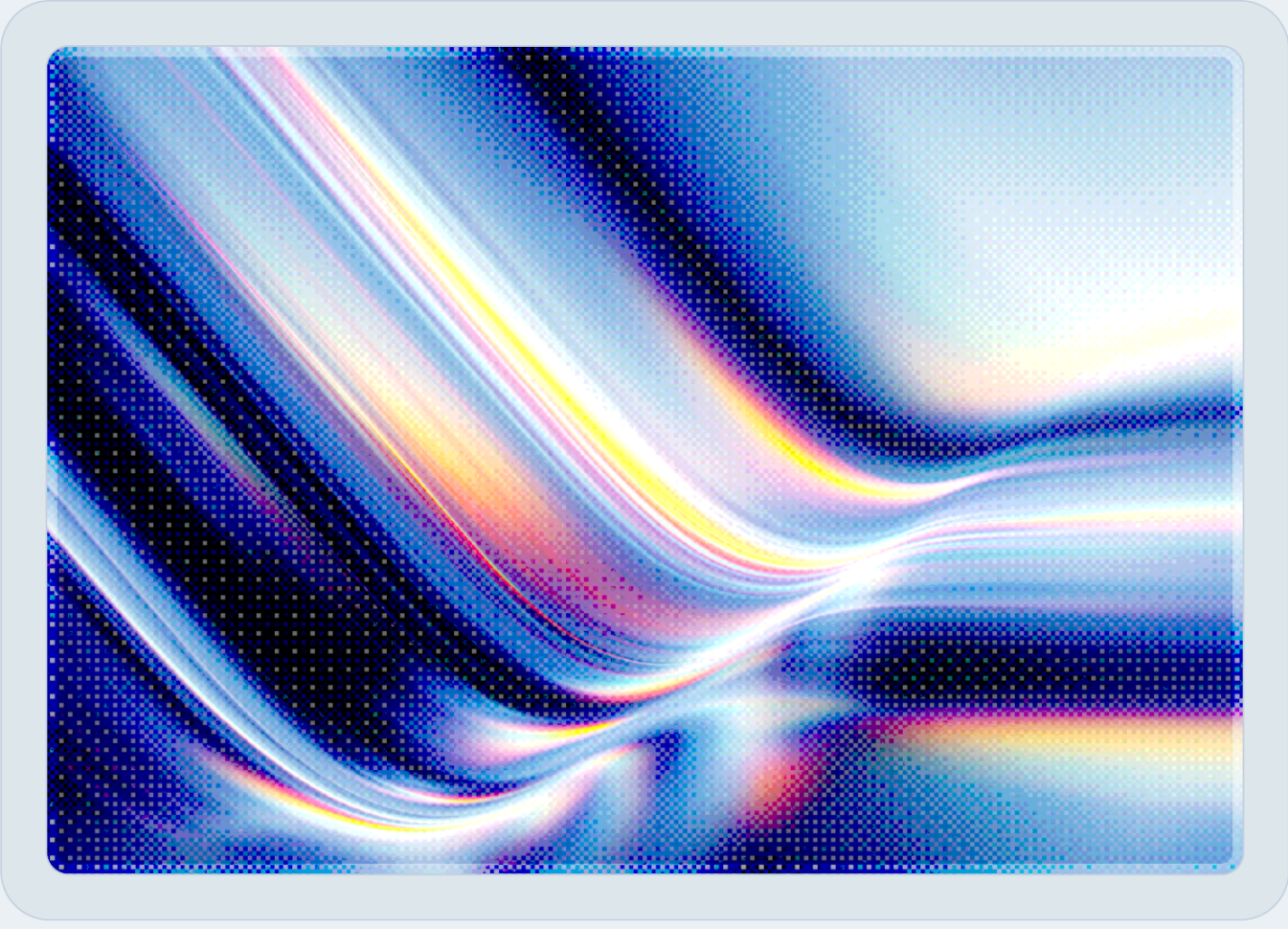
CSAI Foundation | Cloud Security Alliance

Linux Kernel Root Exploits: pedit COW and DirtyClone

Working Public Exploits for CVE-2026-46331 and CVE-2026-43503 Demand Immediate Action

2026-06-29

AI-assisted Rapid Research



CSAI

cloud security alliance®

Key Takeaways

The simultaneous availability of working public exploits for two distinct Linux kernel local privilege escalation vulnerabilities — CVE-2026-46331 ("pedit COW") and CVE-2026-43503 ("DirtyClone") — constitutes a critical and time-sensitive exposure for any organization operating unpatched Linux infrastructure. The following points summarize the operational posture organizations should adopt immediately.

- Both vulnerabilities allow an unprivileged local user to obtain a root shell on unpatched systems where user namespaces are enabled, which is the default on Debian- and Ubuntu-family distributions and is enabled by default in RHEL 9.3+ and RHEL 10.
- Working proof-of-concept exploits are publicly available for both CVEs: the packet_edit_meme PoC for CVE-2026-46331 dropped June 17, 2026 [1]; JFrog Security Research published a detailed exploit walkthrough for CVE-2026-43503 on June 25, 2026 [2].
- DirtyClone is the fourth high-severity local-root flaw to emerge from the Linux kernel's page-cache and zero-copy networking subsystems in approximately six weeks, pointing to a structural audit gap that may yield additional variants [3].
- Kernel patches are available in mainline for both CVEs, but as of late June 2026, Ubuntu and older Debian releases have not shipped backported fixes for all supported versions — organizations running these distributions face a window of active exposure.
- Multi-tenant cloud environments, containerized AI inference pipelines, and shared GPU cluster nodes face elevated risk because any user with local shell access on a shared host can now escalate to root with a publicly available one-command exploit.

Background

The Linux kernel's networking stack has evolved around performance-critical paths, favoring zero-copy data movement, shared memory paths, and in-place cryptographic transformations to sustain the throughput demands of modern cloud and data center workloads. These optimizations introduce a structural tension: when kernel code moves data between network socket buffers and file-backed page cache memory without consistently propagating safety flags across all code paths, the boundary

between user-controlled data and privileged kernel memory becomes fragile. A single missed flag check in any of the dozens of subsystems that handle socket buffer cloning or fragment management can expose a write primitive into memory that holds the in-memory images of setuid binaries.

This structural weakness has produced what security researchers now call the "Dirty" vulnerability family [3][8] — a cluster of Linux local privilege escalation flaws that reach page-cache corruption through different paths in the same general kernel subsystem. DirtyClone (CVE-2026-43503) is the fourth such flaw to surface in roughly six weeks, following Copy Fail (CVE-2026-31431) in late April, the original DirtyFrag in early May, and Fragnesia in mid-May 2026 [3]. The naming convention reflects both the shared exploitation technique and the accumulating evidence that the SKBFL_SHARED_FRAG flag — introduced by the original DirtyFrag patch to mark fragments backed by file page cache memory — is not consistently propagated through all socket-buffer cloning paths in the kernel.

pedit COW (CVE-2026-46331) reaches the same class of exploitation outcome through the traffic-control (tc) subsystem rather than through the zero-copy networking paths, which makes it technically distinct from the DirtyFrag family while sharing the same dangerous consequence: page-cache corruption that allows an unprivileged user to overwrite the in-memory image of a privileged binary. The convergence of these two independently rooted vulnerabilities — both with working exploits, both publicly available within the same two-week window in June 2026 — makes them operationally urgent regardless of their distinct technical origins.

Security Analysis

CVE-2026-46331: pedit COW

Researcher Massimiliano Oldani published the "packet_edit_meme" proof-of-concept on June 17, 2026, one day after kernel.org's CNA formally assigned CVE-2026-46331 [1][4][15]. The vulnerability resides in `tcf_pedit_act()`, the kernel function that implements the `pedit` (packet edit) action within Linux's traffic-control filter framework. The flaw is a bounds-check omission in how the function computes the copy-on-write range it passes to `skb_ensure_writable()`.

Before entering the key processing loop, `tcf_pedit_act()` calculates the range using `tcfp_off_max_hint` — a precomputed value that does not account for the runtime header offset added by typed keys. When the loop executes, part of the write region therefore lies outside the COW protection boundary established by `skb_ensure_writable()`. An attacker who can reach `act_pedit` through unprivileged user namespaces crafts a packet whose edit key targets this

unprotected region, writing attacker-controlled bytes directly into file-backed page cache memory [5]. The published exploit poisons the cached in-memory image of the setuid-root binary `/bin/su` and obtains a root shell in a single command on an affected system [4].

CVE-2026-46331 affects Linux kernel versions v5.18 through v7.1-rc6; the fix is present in v7.1-rc7 [6] [14]. Red Hat has shipped kernel errata for RHEL 8, 9, and 10, and AlmaLinux 8 users have an update available. Debian 13 (Trixie) is patched; Debian 11 (Bullseye) and 12 (Bookworm) remained vulnerable as of this writing. Ubuntu's security tracker listed all supported releases from 18.04 through 26.04 as unpatched as of June 25, 2026 [7]. Red Hat has rated the flaw "Important." Given the availability of a turnkey public exploit, organizations should treat remediation with the same urgency reserved for Critical-rated vulnerabilities, regardless of the vendor's formal severity designation.

CVE-2026-43503: DirtyClone

JFrog Security Research published a detailed exploit walkthrough for CVE-2026-43503 on June 25, 2026 [2]. The vulnerability carries a CVSS score of 8.8 and centers on a flag-propagation failure in `__pskb_copy_fclone()`, the kernel function that clones socket buffer structures during packet duplication. During this cloning operation, `__pskb_copy_fclone()` drops the `SKBFL_SHARED_FRAG` safety flag — the critical marker introduced by the original DirtyFrag patch to identify fragments backed by file page cache memory [8]. The Linux netfilter TEE target, which internally duplicates packets using `__pskb_copy_fclone()`, provides the unprivileged trigger for this chain.

The attack proceeds as follows: the attacker arranges for file-backed memory pages belonging to a privileged binary to be referenced as packet fragment data, then forces the kernel to clone that packet via the TEE target. The cloned packet passes through an IPsec tunnel under the attacker's control. Because the clone no longer carries the `SKBFL_SHARED_FRAG` flag, the decryption operation proceeding in-place overwrites the privileged binary's in-memory authentication checks with attacker-selected bytes [9][10]. The next execution of that binary completes the escalation. Critically, this exploitation chain does not generate kernel audit log entries or on-disk file-integrity artifacts that traditional host-based monitors would detect [2][11]. SIEM-based detection that depends solely on these sources will miss an active DirtyClone exploitation; behavioral detection approaches are required to close this gap.

The fix for CVE-2026-43503 was merged into Linux mainline on May 21, 2026 (commit `48f6a5356a33`) [16], with the first tagged release incorporating the fix being v7.1-rc5 on May 24, 2026 [12]. Distribution backports have reached varying stages: Ubuntu 24.04 LTS is addressed at kernel version 6.8.0-124.124; Ubuntu 22.04 LTS at 5.15.0-181.191 [13]. Debian users should verify their kernel

against the following fixed builds: Bullseye 5.10.257-1, Bookworm 6.1.174-1, and Trixie 6.12.94-1 [13][17]. Red Hat Enterprise Linux maintains a separate advisory track, and RHEL users should consult the Red Hat Customer Portal for their specific kernel errata.

Cloud and Container Exposure

Both vulnerabilities require the attacker to already hold a local user account on the target system — neither is remotely exploitable in the conventional sense. In many traditional single-tenant environments, this requirement limits the practical attack surface. In multi-tenant cloud infrastructure, containerized workload platforms, and shared AI training clusters, however, "local access" describes the baseline condition for any tenant or pipeline stage running as an unprivileged user on a shared host node.

Kubernetes clusters where pods run with user namespace support enabled and without restrictive seccomp profiles meet all prerequisites for exploitation — any pod tenant with a shell becomes a potential attacker against the host kernel. An initial container escape that delivers an unprivileged shell on the underlying host node becomes a full root compromise on any unpatched kernel with these exploits available. Similarly, shared GPU nodes that run inference or training workloads from multiple users or organizational tenants — a common architecture in enterprise machine learning platforms — expose each co-located user as a potential internal attacker once a working exploit is public [9][10]. This means the population of entities that can exploit these CVEs in cloud AI infrastructure is substantially larger than the term "local user" might initially suggest.

The broader structural concern is that available evidence suggests the "Dirty" vulnerability family may not be exhausted [3][8]. The recurring pattern — a missed flag propagation in one of the many code paths that clone or splice socket buffer fragments — suggests a comprehensive audit of SKBFL_SHARED_FRAG propagation has not yet been completed across the full kernel networking stack. Security teams should anticipate additional variants and factor ongoing kernel update discipline into their vulnerability management posture accordingly.

Recommendations

Immediate Actions

Patching both CVEs requires a kernel update followed by a system reboot. There is no runtime configuration change that fully closes the underlying memory corruption primitives for either flaw — the mitigations described below reduce attack surface but do not eliminate the underlying vulnerability [4][5]. The following version thresholds provide distribution-specific guidance:

- **RHEL 8, 9, 10 and AlmaLinux 8**: Apply available Red Hat kernel errata for CVE-2026-46331 and reboot to load the updated kernel.
- **Debian 13 (Trixie)**: Both CVEs are addressed; confirm the running kernel version via `uname -r` against the fixed builds listed above.
- **Ubuntu 24.04 LTS**: CVE-2026-43503 is patched at kernel 6.8.0-124.124; monitor Ubuntu's security tracker for the CVE-2026-46331 errata, which was pending as of June 25.
- **Ubuntu 22.04 LTS**: CVE-2026-43503 is patched at kernel 5.15.0-181.191; same caveat applies for CVE-2026-46331.
- **Debian 11 and 12**: Confirm status against the Debian security tracker for both CVEs; fixed builds are available for CVE-2026-43503 (see builds listed above).

After any kernel update, verify the new kernel is actually running by checking `uname -r`. In cloud VM and managed Kubernetes environments, an explicit reboot step is often required; a successful package installation does not by itself mean the updated kernel is in use.

Short-Term Mitigations

Where immediate kernel patching is not operationally feasible, restricting unprivileged user namespaces eliminates the capability both exploits depend on. On RHEL-family systems, set `user.max_user_namespaces=0` via sysctl; on Debian- and Ubuntu-family systems, set `kernel.unprivileged_userns_clone=0`. This change will break rootless container runtimes such as Podman and rootless Docker, some CI sandbox environments, and sandboxed browser profiles [4][5]. Teams should audit whether any production workloads depend on unprivileged namespaces before applying this control in environments where continuity is critical.

For CVE-2026-46331 specifically, on systems where the `act_pedit` tc action is not in operational use, blocking the kernel module from loading provides targeted reduction without the broad user-namespace trade-off. Adding `install act_pedit /bin/false` to a modprobe configuration file prevents the module from being loaded and removes the exploit's entry point [5]. For containerized workloads that cannot be immediately patched, hardening the container runtime profile — tightening seccomp syscall allowlists, enforcing AppArmor or SELinux policies that deny namespace creation, and disabling the `CAP_NET_ADMIN` capability — provides defense-in-depth, though these controls should be understood as delay measures rather than remediation.

Strategic Considerations

The frequency with which page-cache corruption primitives are emerging from the Linux kernel's zero-copy networking subsystem warrants a review of kernel update cadence at the program level, not merely as a per-incident response. LTS distributions receive backported patches for these vulnerabilities, but the gap between the mainline fix date and the distribution backport availability has consistently spanned several weeks in this cluster of CVEs. Organizations with large Linux fleet infrastructure should evaluate whether their patch deployment SLAs are appropriate for the current vulnerability environment, particularly for workloads that carry elevated internal threat exposure.

Detection strategy also warrants revision in light of DirtyClone's log-free exploitation path [2][11]. Endpoint detection tooling that depends exclusively on kernel audit logs, syslog, or on-disk file-integrity signatures will not produce alerts for an active DirtyClone exploitation. eBPF-based tools that instrument syscall behavior directly — such as Falco, Tetragon, or equivalent — are not subject to this limitation. Behavioral detection capabilities — monitoring for unexpected privilege transitions by non-interactive sessions, anomalous process ancestry on setuid binary execution, sudden appearance of new root-owned processes spawned from unprivileged parents, or unusual network namespace activity — are more likely to surface post-exploitation indicators. Security operations teams should validate their detection coverage includes these behavioral signals before treating the DirtyClone window as closed.

At the architectural level, AI and machine learning workloads running on shared infrastructure deserve specific re-evaluation. Shared inference serving hosts, multi-tenant model training clusters, and ML pipeline orchestration systems that grant multiple users or service accounts shell access on common nodes represent the threat model that these CVEs make exploitable in practice. Workload isolation practices — dedicated node pools segmented by trust level, mandatory access control enforcement at the kernel layer, and ephemeral credentials that limit the value of a root compromise — should be reviewed and validated as part of the response to this cluster of vulnerabilities.

CSA Resource Alignment

Several CSA frameworks and guidance areas apply directly to the risks described in this note. The Cloud Controls Matrix (CCM) Vulnerability and Patch Management domain requires that organizations maintain a capability to assess vulnerability severity and remediate within defined timeframes calibrated to that severity. Both CVE-2026-46331 and CVE-2026-43503 present working public exploits, which typically triggers the highest-priority remediation tier in any CCM-aligned vulnerability management program. Organizations that have implemented CCM controls TVM-01 through TVM-09 should verify that their

patch SLAs and escalation processes apply to kernel-level vulnerabilities in the same way they apply to application-layer flaws, as kernel CVEs are sometimes under-tracked in application-centric vulnerability programs.

CSA's Zero Trust guidance is directly relevant to the multi-tenant exposure scenario these vulnerabilities create. A Zero Trust architecture that enforces per-workload micro-segmentation and least-privilege access means that a root compromise of a single host node does not automatically translate into lateral movement or credential access across the broader environment. The Zero Trust principle of continuous verification at the workload boundary — rather than trusting identity based on network position — provides meaningful containment even when kernel-level exploitation occurs. Organizations should evaluate whether their current micro-segmentation and credential issuance architecture would limit the blast radius of a successful privilege escalation on a shared node.

The STAR (Security Trust Assurance and Risk) program provides a structured mechanism for cloud service providers and managed infrastructure operators to disclose their patching posture on vulnerabilities such as these. Organizations procuring cloud infrastructure or managed Kubernetes platforms should request evidence through STAR attestations or equivalent supplier security documentation that kernel updates addressing CVE-2026-46331 and CVE-2026-43503 have been applied and that hosts have been rebooted to activate the updated kernels. Absence of this evidence should be treated as an open risk item in supplier assurance reviews.

CSA's AI Safety Initiative work on secure AI infrastructure connects directly to the elevated risk profile of shared AI compute environments described in this note. The same unprivileged-user-to-root attack chain that threatens general enterprise Linux hosts also threatens the worker nodes and inference servers that run AI workloads. Infrastructure teams building or operating AI systems should apply the same rigor to underlying kernel security as they apply to AI-layer controls, recognizing that a kernel-level compromise on a shared GPU node may expose not only system credentials but also in-memory model weights, training data cached in host memory, and API keys used by inference services running on that host.

References

[1] CyberSecurityNews. "New Linux pedit COW Exploit Allows Attackers to Gain System Root Access." CyberSecurityNews, June 2026.

[2] JFrog Security Research. "Dissecting and Exploiting Linux LPE Variant: DirtyClone (CVE-2026-43503)." JFrog Security Research Blog, June 25, 2026.

[3] Latest Hacking News. "DirtyClone Is the Fourth 'Dirty' Linux Kernel Exploit in Six Weeks." Latest Hacking News, June 28, 2026.

[4] TuxCare. "pedit-cow (CVE-2026-46331): Linux tc Flaw Grants Root." TuxCare Blog, June 2026.

[5] CloudLinux. "pedit COW (CVE-2026-46331): Mitigation and Kernel Update on CloudLinux." CloudLinux Blog, June 2026.

[6] The CyberSec Guru. "Two New Linux LPEs Hit Page Cache from Opposite Ends of the Kernel." The CyberSec Guru, June 2026.

[7] Ubuntu Security. "CVE-2026-46331." Ubuntu Security Tracker, June 2026.

[8] The Hacker News. "New DirtyClone Linux Kernel Flaw Lets Local Users Gain Root via Cloned Packets." The Hacker News, June 2026.

[9] CyberSecurityNews. "New DirtyClone Linux Vulnerability Allows Attackers to Gain Root Access Via Cloned Packets." CyberSecurityNews, June 2026.

[10] Threat-Modeling.com. "CVE-2026-43503: 'DirtyClone' Linux Kernel Local Privilege Escalation to Root via Cloned Network Packets." Threat-Modeling.com, June 2026.

[11] TechTimes. "Linux Kernel Root Exploit Published: DirtyClone Attack Leaves No Trace." TechTimes, June 27, 2026.

[12] SC Media. "2 Linux Kernel Flaw PoCs Published, Enabling Local Privilege Escalation." SC Media, June 2026.

[13] HowToUseLinux. "DirtyClone (CVE-2026-43503): What It Is and How to Patch It." HowToUseLinux, June 2026.

[14] NIST National Vulnerability Database. "CVE-2026-46331 Detail." NVD, June 2026.

[15] CyberPress. "New Pedit COW Linux Kernel Flaw Lets Local Users Gain Root Access." CyberPress, June 2026.

[16] Linux Kernel Git Repository. "commit 48f6a5356a33 – net: fix SKBFL_SHARED_FRAG propagation in __pskb_copy_fclone." kernel.org, May 21, 2026.

[17] Debian Security Team. "CVE-2026-43503 – Debian Security Tracker." Debian Security Tracker, June 2026.