Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

Bad Epoll: Linux Kernel eventpoll Use-After-Free Enables Local Root

Date of Publication	Admiralty Code	TA Number
July 6, 2026	A1	TA2026188

Summary

First Seen: May 30, 2026
Affected Products: Linux kernel (Including Linux servers, desktops, and Android)
Impact: Bad Epoll (CVE-2026-46242) is a high-severity (CVSS 7.8) use-after-free race condition in the Linux kernel's eventpoll subsystem that lets an unprivileged local user escalate to root on Linux servers, desktops, and Android devices. A highly reliable public proof-of-concept exists, and the flaw is reachable even from the Chrome renderer and Android app sandboxes, while typically evading KASAN detection after an adjacent fix. It affects mainline kernels from 6.4 onward plus backported 5.15.y and 6.1.y branches, with no configuration workaround, patching is the only remediation. No confirmed in-the-wild exploitation, but the complete public disclosure makes immediate patching urgent.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-46242	Bad Epoll (Linux Kernel Use After Free Vulnerability)	Linux Kernel	✗	✗	✓

Vulnerability Details

#1 The Linux kernel's eventpoll (epoll) subsystem, the core I/O event notification mechanism that servers, network services, and web browsers all rely on to monitor many file descriptors at once, is affected by CVE-2026-46242, a high-severity local privilege escalation vulnerability known as Bad Epoll. Successful exploitation allows an unprivileged process to escalate to root, not only on Linux desktops and servers but also on Android devices, granting full administrative control over the affected host and the data it processes.

#2 The root cause is a use-after-free race condition in the epoll cleanup path. ep_remove() clears file->f_ep under a lock but continues using the file object inside the critical section; a concurrent __fput() in that window frees the watched struct eventpoll, and the subsequent list operation writes into freed memory. Compounding this, struct file is SLAB_TYPESAFE_BY_RCU, so the backing slot can be recycled mid-operation, yielding an attacker-controllable free against the wrong slab cache. The flaw requires only local code execution with no special privileges and no user interaction, though as a race condition it depends on winning a narrow timing window.

#3 That window is unusually exploitable. A public proof-of-concept succeeds around 99% of the time despite a race window only about six instructions wide, and atypically for a kernel bug, it is reachable from inside Chrome's renderer sandbox, which blocks almost every other kernel bug, as well as from Android's app sandbox. After the adjacent CVE-2026-43074 fix, the use-after-free usually no longer trips KASAN, leaving little runtime detection signal, and there is no configuration workaround because epoll cannot be turned off. No wild exploitation has been confirmed and no host or network indicators of compromise have been published as of today, so detection depends on kernel patch-state tracking rather than IOC matching.

#4 The vulnerability affects mainline Linux kernels from version 6.4 onward, with backported exposure on the long-term-support 5.15.y branch (from 5.15.209) and 6.1.y branch (from 6.1.175), and is fixed in builds 6.18.33 and 7.0.10 and the 7.1 branch. As of disclosure, no in-the-wild activity had been confirmed, but a complete public technical writeup and a working exploit make patching urgent, particularly for multi-tenant hosting, CI/CD build environments, and enterprise Android fleets with slow patch cycles.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-46242	Linux kernel: v6.4 up to (excluding) 6.18.33; v6.19 up to (excluding) 7.0.10; stable 5.15.209 up to (excluding) 5.16; stable 6.1.175 up to (excluding) 6.2. Fixed in 6.18.33, 7.0.10, and 7.1.	cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:*	CWE-416

Recommendations

Apply the Upstream Kernel Patch Immediately: Update affected systems to a fixed kernel release (6.18.33, 7.0.10, or 7.1) or to a distribution kernel that has backported the corrective commit a6dc643c6931, then reboot to load the patched kernel. This fix pins the file object via epi_fget() at the entry of ep_remove() and gates the critical section on the pin succeeding, closing the race and both use-after-free conditions. Applying the patch is the only complete remediation.

Prioritize High-Exposure Hosts: Direct patching effort first at multi-user systems, container build hosts, shared-compute and multi-tenant platforms, and any environment where untrusted code can obtain a local shell, because these are the contexts in which local privilege escalation is most readily reachable and most damaging. Inventory Linux hosts and rank them by exposure to untrusted local execution.

Apply Distribution and Android Vendor Updates: Consult each Linux distribution's security advisories for a kernel update that includes the backport, and apply Google Android security updates promptly on devices running kernel v6.6 or later. Devices on v6.1-based kernels that did not receive the vulnerable backport are not affected and do not require this specific fix.

Harden Systems as an Interim Measure Only: Where immediate patching is not possible, raise the exploitation bar by enabling kernel address-space layout randomization (KASLR) and SLUB allocator randomization, since the exploit's race-widening technique depends on predictable heap layout, and use mandatory access controls such as SELinux or AppArmor to constrain what a successfully escalated process can do. These measures do not prevent the escalation itself and epoll cannot be disabled, so treat hardening strictly as a stopgap that buys time until the patch is deployed.

Use Live Patching Where Available: On hosts that cannot tolerate an immediate reboot, apply distribution-provided kernel live patching to close the vulnerable window without downtime, then schedule a full patched-kernel reboot at the earliest maintenance opportunity.

Restrict Untrusted Local Code Execution: Until affected hosts are patched, limit local shell access to trusted users and restrict or suspend untrusted or tenant-supplied workloads on shared kernels to reduce the population of principals able to trigger the race.

Monitor for Exploitation and Corruption Signals: Centralize kernel ring-buffer and crash-dump collection, and alert on kernel oops or panic messages referencing ep_remove, ep_remove_file, hlist_del_rcu, or ep_clear_and_put; KASAN use-after-free reports tied to kmalloc-192 eventpoll allocations; slab-corruption or unexpected kmem_cache_free warnings during heavy epoll workloads; and RCU stalls or general protection faults originating in fs/eventpoll.c. Correlate such telemetry with workloads that intensively close watched epoll descriptors from multiple threads.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Privilege Escalation	T1068: Exploitation for Privilege Escalation	
Resource Development	T1588: Obtain Capabilities	T1588.006: Vulnerabilities

Patch Links

https://git.kernel.org/stable/c/a6dc643c69311677c574a0f17a3f4d66a5f3744b

https://git.kernel.org/stable/c/ced39b6a8062bac5c18a1c3df85634107eb8664a

https://git.kernel.org/stable/c/ef4ca02e95363e78977ca04340d44fe3b4b2b81f

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=9324de74a3a59b9fde9b62ee45ebaa71458ba2e5

References

https://www.sentinelone.com/vulnerability-database/cve-2026-46242/

https://github.com/J-jaeyoung/bad-epoll

What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a free demo of **HivePro.**

REPORT GENERATED ON

July 06, 2026 • 8:30 AM

More at www.hivepro.com