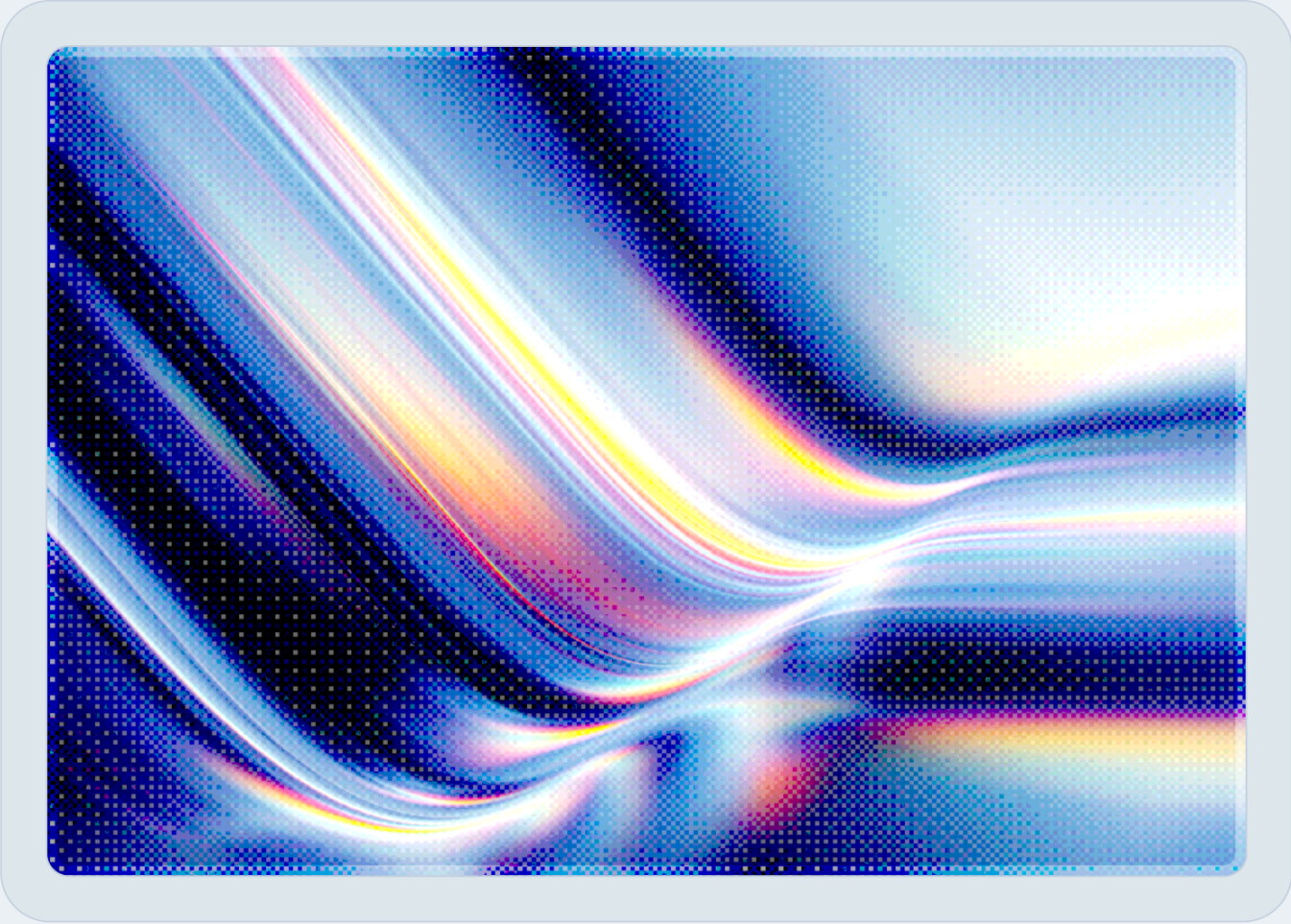
CSAI Foundation | Cloud Security Alliance

Dirty Frag: Linux Kernel LPE Zero-Day in AI/ML Infrastructure

2026-05-08

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- **Dirty Frag** (CVE-2026-43284, CVE-2026-43500) is an unpatched local privilege escalation (LPE) zero-day affecting all major Linux distributions running kernels released since 2017, disclosed on May 7–8, 2026 after a premature embargo break.
- The vulnerability chains two in-kernel page-cache write flaws — one in the IPsec ESP subsystem (CVE-2026-43284) and one in the RxRPC subsystem (CVE-2026-43500) — to achieve deterministic, near-certain root access without a race condition.
- A public proof-of-concept exploit exists. Upstream patches are partially merged but not yet available in production distributions as of May 8, 2026. Any user with local shell access on an affected system can obtain root today.
- AI/ML infrastructure is acutely exposed: inference servers, Kubernetes worker nodes, and GPU compute clusters almost universally run affected Linux distributions. Containerized workloads are unlikely to provide isolation against this class of host-kernel attack; the predecessor vulnerability, Copy Fail, confirmed container escape through the shared page cache, and the structural conditions for the same outcome are present in Dirty Frag.
- The immediate mitigation — blacklisting the `esp4`, `esp6`, and `rxrpc` kernel modules — eliminates the attack surface at the cost of disabling IPsec tunnels and the RxRPC/kAFS protocol, a trade-off that must be carefully evaluated for each environment.
- Dirty Frag follows Copy Fail (CVE-2026-31431), a closely related LPE disclosed April 29, 2026 that CISA has added to its Known Exploited Vulnerabilities catalog with a May 15, 2026 federal remediation deadline.

Background

The Linux kernel's networking and cryptographic subsystems contain a class of latent vulnerability in which receive-path code performs in-place decryption or modification of memory that is not exclusively owned by the kernel. When an attacker can arrange for a page from the kernel's page cache — the in-memory representation of filesystem data shared across all processes on the system — to be planted into

a socket buffer's fragment slot, the subsequent decryption operation writes attacker-controlled bytes directly into that shared memory. The modification persists in RAM without touching the disk, leaving no forensic trace in file-integrity monitoring or audit logs that examine on-disk state.

This general class of flaw is not new, but it has received renewed attention following a sequence of disclosures in April and May 2026. On April 29, 2026, security researchers at Theori publicly disclosed Copy Fail (CVE-2026-31431), a logic flaw in the `algif_aead` module of the kernel's AF_ALG userspace crypto API that allowed an unprivileged user to write four controlled bytes into the page cache of any readable file [1]. Copy Fail proved exploitable with a 732-byte Python script, affected every major distribution shipping kernels since 2017, and confirmed the viability of container escape via the shared page cache [2][12]. CISA added it to the Known Exploited Vulnerabilities catalog within days, ordering US federal civilian agencies to remediate by May 15, 2026 [3].

Dirty Frag, disclosed by researcher Hyunwoo Kim on May 7–8, 2026, occupies the same threat category but exploits different and previously unreported code paths [4]. The vulnerability was reported to Linux kernel maintainers on April 30, 2026. Its premature public disclosure — caused when an unrelated third party independently published an exploit — led the researcher to release full technical details and proof-of-concept code before distribution vendors had prepared patches, to ensure defenders had accurate information [5]. As a result, organizations faced a working, publicly available exploit with no vendor-supplied fix.

Security Analysis

Technical Mechanism

Dirty Frag chains two independent page-cache write primitives, either of which alone yields root, through different but structurally similar logic errors.

CVE-2026-43284 (ESP variant). The vulnerability resides in the IPsec ESP receive path (`esp_input()`) within the `esp4` and `esp6` kernel modules, present since a kernel commit from January 2017 [5]. When a non-linear socket buffer carrying a splice-pinned page cache reference reaches the ESP decryption path, the code is expected to call `skb_cow_data()` to ensure the kernel privately owns all fragment pages before modifying them. The bug allows this copy-on-write check to be bypassed, after which the ESP AEAD decryption routine performs an in-place four-byte store directly

into the page cache. The attacker controls both the target file offset and the value written. The exploit uses this primitive to overwrite a fragment of `/usr/bin/su` in memory with a small embedded x86-64 ELF root shell, then executes it through the normal setuid mechanism [5].

CVE-2026-43500 (RxRPC variant). This variant resides in the `rxkad_verify_packet_1()` function within the `rxrpc` kernel module, which implements the rxkad authentication layer for the AFS/RxRPC network protocol, introduced in June 2023 [5]. The logic error is structurally similar: when an externally-backed splice page reaches the rxkad decryption path, the kernel performs an in-place eight-byte `pcbc(fcrypt)` decryption operation over attacker-influenced fragments. The attacker can precompute a suitable decryption key entirely in user space, then trigger the kernel write to modify the `/etc/passwd` root entry's password field to an empty string. Standard PAM authentication with the `nullok` option then allows password-free root login [5].

Both variants share important characteristics that elevate their exploitability beyond typical kernel bugs. Neither relies on a timing window or race condition — both are deterministic logic errors. The kernel does not panic when an exploit attempt fails, making repeated attempts safe. The success rate is described by the researcher as very high, and the exploit leaves no trace on disk because it operates entirely in the page cache, not on backing storage [5].

Affected Scope

Dirty Frag affects all Linux kernels from approximately January 2017 (when the ESP variant was introduced) through at least kernel 7.0.x. Confirmed vulnerable distributions include Ubuntu 24.04.4 (kernel 6.17.0-23-generic), Red Hat Enterprise Linux 10.1, CentOS Stream 10, AlmaLinux 8, 9, and 10, openSUSE Tumbleweed (kernel 7.0.2-1-default), and Fedora 44 [5][6][7][11]. Amazon Web Services issued a security bulletin on May 8, 2026 confirming that Amazon Linux is affected and that patches are pending [8]. The affected distributions represent the majority of enterprise and cloud Linux deployments; organizations running any major distribution on kernels from January 2017 onward should treat themselves as affected until a vendor advisory explicitly states otherwise.

AlmaLinux 8 is affected only by CVE-2026-43284, as its kernel configuration does not include the `rxrpc` module by default. AlmaLinux 9 and 10 are affected by both CVEs, though CVE-2026-43500 requires the optional `kernel-modules-partner` package to be installed [7]. This nuance is significant for organizations auditing exposure: the absence of rxrpc-based exposure does not eliminate risk from the ESP variant, which is present in all configurations that load `esp4` or `esp6`.

AI/ML Infrastructure Exposure

The AI/ML infrastructure stack is built almost entirely on Linux. Training clusters, inference servers, GPU compute nodes, and the Kubernetes worker nodes that schedule AI workloads all run distributions confirmed vulnerable to Dirty Frag. Because the vulnerability requires only local shell access — not network reachability, not a specific service, and not elevated starting privileges — any user with an SSH session or interactive terminal on an affected system can reliably escalate to root.

This exposure is particularly acute in several common AI/ML deployment patterns. Multi-tenant GPU clusters, where multiple teams or customers share physical GPU hardware through Kubernetes and container abstractions, represent a high-risk configuration: a single compromised or malicious tenant could use Dirty Frag to escalate from container-level access to root on the host node, potentially accessing models, training data, and API credentials belonging to all other tenants on that node. While the researcher has not explicitly confirmed container escape as a tested primitive, the page cache is shared between the host kernel and all containers running on it — and the predecessor vulnerability, Copy Fail, was confirmed to enable container escape through this mechanism [2][9].

AI inference pipelines that expose shell access to external-facing model serving processes, agentic frameworks that spawn subprocesses or code interpreters, and CI/CD runners that execute untrusted training code are all environments where the "local" precondition for exploitation can be satisfied by attackers who do not have direct system access.

IPsec VPN tunnels, which are commonly used to secure traffic between AI compute nodes and data stores or between cloud regions, rely on the `esp4` and `esp6` kernel modules. The recommended module-blacklisting mitigation for Dirty Frag disables these tunnels as a side effect, a trade-off that infrastructure teams must evaluate carefully [7][8].

Patch and Disclosure Status

The upstream patch for CVE-2026-43284 was merged into the Linux kernel's netdev development tree on May 7, 2026 [5]. The upstream patch for CVE-2026-43500 (RxRPC variant) remained unmerged as of the same date. AlmaLinux released patched kernels to its testing repository on May 7–8, 2026, noting that community verification is required before the packages are promoted to production [7]. CloudLinux released kernel updates and announced that KernelCare livepatches were in preparation [10]. Most major distributions, including Ubuntu, RHEL, Amazon Linux, and openSUSE, had not released production kernel updates as of May 8, 2026.

No CVSS scores had been formally assigned to either CVE as of the time of writing, because CVE assignment followed rather than preceded the embargo break. In the authors' assessment, the impact profile — full root compromise, deterministic exploitability, and broad affected scope — is consistent with a CVSS Critical rating, though formal scores had not been assigned as of the time of writing.

Recommendations

Immediate Actions

The priority response for AI/ML infrastructure operators is to assess exposure and apply module-level mitigations before distribution patches are available. Operators should first inventory which systems have `esp4`, `esp6`, and `rxrpc` modules loaded, using `lsmod | grep -E 'esp4|esp6|rxrpc'`, then assess whether IPsec tunnels or RxRPC/kAFS are in active use before applying the blacklist.

For systems where the dependency inventory confirms that IPsec is not in active use, the following configuration eliminates the attack surface immediately without disrupting workloads:

```
sudo sh -c "printf 'install esp4 /bin/false\ninstall esp6
/bin/false\ninstall rxrpc /bin/false\n' >
/etc/modprobe.d/dirtyfrag.conf"
sudo modprobe -r esp4 esp6 rxrpc 2>/dev/null || true
sudo sh -c 'echo 3 > /proc/sys/vm/drop_caches'
```

The cache flush after module removal is recommended by AlmaLinux to clear any pages that may have been modified by exploit activity [7]. The `modprobe.d` configuration file ensures that modules do not reload on the next kernel restart.

For systems where disabling IPsec would disrupt legitimate connectivity, AWS additionally recommends disabling kernel module autoloading entirely with `sysctl -w kernel.modules_disabled=1` after completing normal boot, and reviewing whether user namespace creation can be restricted with `sysctl -w user.max_user_namespaces=0` to reduce the attack surface for this and related privilege escalation classes [8].

Short-Term Mitigations

Organizations should prioritize production kernel updates as soon as their distribution releases patched packages. Patch availability should be tracked through official distribution security channels: Ubuntu Security Notices, Red Hat Customer Portal, Amazon Linux Security Center, and SUSE Security Advisories. AlmaLinux users can accelerate testing of the available patched kernel in non-production environments to validate stability before production rollout.

Multi-tenant environments — particularly GPU clusters and shared Kubernetes nodes — should be treated as highest priority for patching, given that exploitation by one tenant can compromise the entire node's workload isolation. Organizations operating such environments should consider whether additional host-level isolation, such as hypervisor-based separation between GPU tenants, is warranted given the current state of container-level isolation against host-kernel LPE attacks.

Detection of exploitation attempts should focus on anomalous setuid binary execution from unexpected user contexts, unexpected modifications to `/etc/passwd` group or authentication behavior, and unusual `splice()` or `sendmsg()` system call patterns in kernel audit logs. Standard file-integrity monitoring tools will not detect page-cache-only modifications because those modifications do not touch on-disk state.

Security teams should also evaluate their exposure to Copy Fail (CVE-2026-31431) concurrently. That vulnerability is already under active exploitation, has a CISA KEV remediation deadline of May 15, 2026 for federal agencies, and targets an adjacent code path [3]. Organizations patching for Dirty Frag should verify that Copy Fail remediation is also complete.

Strategic Considerations

The rapid succession of Dirty Frag and Copy Fail — two structurally related kernel vulnerabilities disclosed within ten days, both affecting all major distributions, both exploitable with minimal prerequisites — suggests that security researchers are systematically examining the Linux kernel's in-place cryptographic operation code paths, and organizations should assume that threat actors are monitoring these disclosures closely. Organizations should assume that additional variants in this class may be discovered.

More broadly, Dirty Frag illustrates a systemic risk in AI/ML infrastructure: the assumption that container or namespace boundaries provide meaningful isolation against a privileged local attacker. When the underlying host kernel contains a deterministic LPE requiring only local shell access, container-level security controls are insufficient. This is consistent with findings from Copy Fail and from the November 2025 runC vulnerability cluster, which demonstrated container escape across Docker, containerd, and CRI-O [13]. Organizations operating AI/ML workloads in shared-kernel environments should review

whether workloads requiring strong isolation, such as models processing confidential prompts or training on sensitive datasets, should be migrated to hypervisor-isolated environments such as dedicated VMs or MicroVM runtimes [9].

The premature embargo break that preceded Dirty Frag's disclosure also highlights a coordination gap in the Linux vulnerability disclosure ecosystem. When an unrelated party independently discovers and publishes an exploit before the coordinated disclosure window closes, distribution vendors are left without patches at the moment of greatest exposure. AI/ML infrastructure teams should treat coordinated disclosure announcements on the `oss-security` mailing list and `linux-distros` list as real-time threat signals, not post-incident notifications, and should have runbooks for applying module-blacklist mitigations within hours of a high-severity kernel LPE disclosure.

CSA Resource Alignment

Dirty Frag maps directly to control domains and risk models defined across several CSA frameworks.

The CSA **MAESTRO** framework (Multi-layer AI Threat and Responsibility Operations) identifies host operating system privilege escalation as a Tier 1 infrastructure threat that can undermine all higher-level AI safety controls. When an attacker gains root on an AI inference host, they can modify model weights in memory, intercept inference requests and responses, exfiltrate training data and API credentials, and disable or tamper with runtime monitoring agents. MAESTRO's infrastructure hardening requirements — network segmentation, least-privilege host access, and kernel hardening baselines — are directly applicable to the mitigations described in this note.

The **AI Controls Matrix (AICM)**, CSA's AI security control framework and superset of the Cloud Controls Matrix, addresses the intersection of virtualization security, host hardening, and multi-tenant isolation under its infrastructure security control families. The Dirty Frag scenario — where shared-kernel container isolation fails against a host-kernel LPE — illustrates why AICM's isolation controls extend to the hypervisor layer for workloads handling sensitive data, not merely to the container runtime. Organizations implementing AICM should review whether their AI workload isolation controls account for host-kernel LPE as a realistic threat scenario given the current threat landscape.

CSA's **Zero Trust Guidance** for cloud and hybrid environments emphasizes that trust should not be implicitly granted based on network position or container boundary. Dirty Frag concretely demonstrates why: a user with legitimate restricted access to a Linux shell — a position that would normally be

considered low-privilege and constrained — can trivially elevate to root. Zero trust implementations should treat the local host as a potential adversarial environment and minimize the secrets, credentials, and model assets available to any local process.

The **AI Organizational Responsibilities** guidance addresses the governance dimension of rapid vulnerability response. The Dirty Frag disclosure scenario — a working exploit published before patches are available, affecting every major AI infrastructure distribution simultaneously — represents exactly the type of event for which organizations need pre-defined response procedures, pre-authorized emergency change management pathways for kernel updates, and clear ownership of AI infrastructure security responsibilities at the team level.

References

[1] Theori. "Copy Fail: 732 Bytes to Root on Every Major Linux Distribution." copy.fail, April 29, 2026.

[2] University of Toronto Information Security. '"Copy Fail" Linux kernel LPE and container escape.' security.utoronto.ca, April–May 2026.

[3] CISA. "Known Exploited Vulnerabilities Catalog." cisa.gov, May 2026.

[4] Hyunwoo Kim (V4bel). "dirtyfrag — Universal Linux LPE." GitHub, May 7–8, 2026.

[5] Hyunwoo Kim. "Dirty Frag: Universal Linux LPE." oss-security mailing list (Openwall), May 7, 2026.

[6] BleepingComputer. "New Linux 'Dirty Frag' zero-day gives root on all major distros." BleepingComputer, May 8, 2026.

[7] AlmaLinux. "Dirty Frag (CVE-2026-43284, CVE-2026-43500) vulnerability fix is ready for testing." AlmaLinux Blog, May 7, 2026.

[8] Amazon Web Services. "Dirty Frag and other issues in Amazon Linux kernels." AWS Security Bulletins, May 8, 2026.

[9] LWN.net. "Dirty Frag: a zero-day universal Linux LPE." LWN.net, 2026.

[10] CloudLinux. "Dirty Frag [CVE-2026-43284]: Mitigation and Kernel Update on CloudLinux." CloudLinux Blog, May 2026.

[11] The Hacker News. "Linux Kernel Dirty Frag LPE Exploit Enables Root Access Across Major Distributions." The Hacker News, May 8, 2026.

[12] Help Net Security. "Nine-year-old Linux kernel flaw enables reliable local privilege escalation (CVE-2026-31431)." Help Net Security, April 30, 2026.

[13] Cloud Native Computing Foundation. "runc container breakout vulnerabilities: A technical overview." CNCF Blog, November 28, 2025.