

TEMPESTOS

Secure by Design

Detect-and-patch assumes defenders find flaws first. AI-assisted research has ended that assumption. TempestOS removes the attack surface before the next flaw is named.

In a single month this spring, two unrelated Linux kernel flaws, each dormant in shipping code for years, were found, weaponized, and published before any patch existed. This was not bad luck. AI-assisted analysis can now surface vulnerabilities faster than the people who maintain the code can fix them. For platforms whose entire purpose is to be trustworthy under hostile conditions, **that changes what good security has to look like.**

■ THE PATTERN

Two kernel flaws in a month, surfaced by AI

Copy Fail

CVE-2026-31431 · CVSS 7.8 · late April 2026

A flaw in the kernel userspace crypto interface, dormant since 2017.

Any local user gains root in seconds, on every major Linux distribution.

Dirty Frag

CVE-2026-43284 (8.8) + CVE-2026-43500 (7.8) · May 7, 2026

A separate pair of flaws in the IPsec and RxRPC subsystems, unrelated to Copy Fail and dormant since 2023.

Any local user gains root, with a working exploit public before any patch shipped.

Both flaws sat in shipping kernels for years, one since 2017 and the other since 2023, until they surfaced within weeks of each other. Code does not become vulnerable on a schedule. That sudden cluster, after years of dormancy, is the signature of a new method: AI-assisted analysis now reads the kernel at a scale no human review ever could. Reported AI-related security incidents rose more than 56 percent in a single year per Stanford's AI Index, and the window between disclosure and exploitation is collapsing toward zero.

■ THE PROBLEM

Detect-and-patch is reactive by definition

The standard response to a vulnerability has always been the same: find the bug, write the fix, and wait for the patch to reach every fielded system. That model assumes defenders find flaws before adversaries and that patches outrun exploits. AI-assisted research inverts both. Discovery is faster, exploit code is often public on day one, and embargoes break: Dirty Frag's working exploit went public before any distribution shipped a fix. An operating system designed around detect-and-patch is a step behind by construction, and for systems built to be trustworthy under contested or disconnected conditions, that is not acceptable.

The deeper problem is the operating system itself. Windows, Red Hat Enterprise Linux, and every mainstream distribution are built for maximum compatibility, so they ship thousands of drivers, modules, and services enabled by default, most of which a given platform will never use. Neither Copy Fail nor Dirty Frag lived in code an autonomous platform has any reason to run: one sat in a userspace crypto interface, the others in IPsec and a distributed-filesystem protocol, yet every stock distribution shipped them switched on. The tell is in the fixes: the official mitigation for both was simply to disable the affected modules, which means **the industry's emergency response was to delete code that should never have been running.** TempestOS deletes these and thousands more at build, before the CVE is ever known.

■ THE PRINCIPLE

Security by design, not perpetual patching

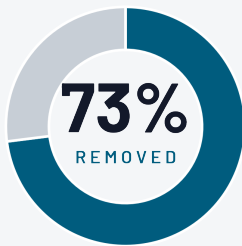
The durable answer is to build systems secure from the ground up, with the architecture hardened by design and the attack surface reduced before the next vulnerability is named. This is not a vendor theory. It is the conclusion the upstream kernel maintainers reached: their lasting fix for the Copy Fail class was not a patch alone but the removal of the rarely-used interface that hosted the flaw, so bugs in that class cannot recur.

A feature that is not compiled into the binary cannot be exploited, cannot be misconfigured, and needs no emergency patch when the next CVE lands on it. Removal is the resilient move.

■ THE PROOF

How TempestOS applies this

TempestOS is a purpose-built, immutable operating system for the tactical edge. It is built down to the mission: a workload trace identifies exactly which kernel features a platform uses, and everything else is compiled out.



■ Removed at build ■ Retained, 1,282 options

Stock Linux ships 4,774 enabled kernel options. TempestOS ships 1,282, each one traced to evidence.

TempestOS, hardened by design

- **Immutable, signed image.** No configuration drift in the field.
- **Built down to the mission.** Roughly 95 percent STIG compliant at build.
- **The emerging platform of trust** for defense primes and new innovators.

Every removed feature is code that cannot be attacked, because it is not in the binary at all.

CASE IN POINT • COPY FAIL

When Copy Fail was disclosed, the vulnerable subsystem was not present in TempestOS. No emergency patch, no module blocklist, no waiting on backports. **The nodes were never exposed.**

■ WHY IT MATTERS

Why this matters now

The global shift toward autonomous force design demands a complete re-evaluation of the trust and security layer on which software-defined national security will be built. Autonomy multiplies the number of systems that must be trusted and pushes them into contested, disconnected environments where a patch may never arrive in time. The operating system is that trust layer. Building it secure by design, with the attack surface removed before the threat is named, is the foundation the rest of the force depends on.

Orchestrate the Storm

Caliburn Technologies
sales@caliburn.us • caliburn.us

Sources: distribution and vendor advisories for CVE-2026-31431 (Copy Fail) and CVE-2026-43284 / CVE-2026-43500 (Dirty Frag), including Tenable, Sysdig, Wiz, AlmaLinux, Ubuntu, and Red Hat; The Register and the Stanford HAI AI Index on AI-assisted vulnerability discovery. TempestOS kernel surface figures from internal build report, build 20260603. Business Proprietary.