A New Paradigm for Vulnerability Analysis in Human-AI Collaboration: An Analysis of the "Copy Fail" Discovery Process

The original report is in Chinese, and this version is an AI-translated edition.

AI-powered vulnerability discovery and automated attacks are undoubtedly key focuses for 2026. Previously, Anthropic's batch of discoveries of a FreeBSD remote root vulnerability (latent for 17 years), an FFmpeg H.264 heap out-of-bounds write vulnerability (latent for 16 years), and an OpenBSD remote crash vulnerability (latent for 27 years) showcased the capabilities of the Mythos Preview model and Project Glasswing. However, due to the Copy Fail vulnerability being a local privilege escalation (LPE) vulnerability, despite causing a significant amount of overtime work for security operators and analysts worldwide during Labor Day, it received little attention from AI-driven self-media with limited understanding. Clearly, for security researchers, compared to the somewhat veiled approach of Mythos Preview, the exposure process of the Copy Fail vulnerability offers more valuable information, and its combination of "researchers + AI platform" is a more worthy model for study.

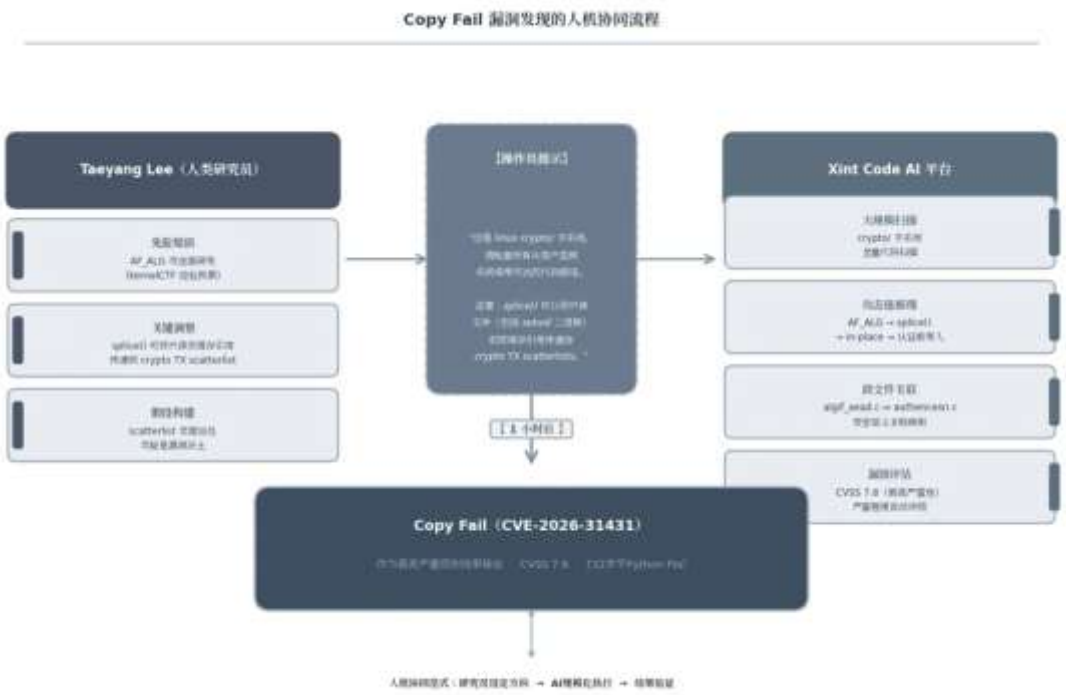
1. Background: The discovery of the ″Copy Fail″ vulnerability represents a new paradigm of human-machine integration.

1.1 The discovery process of the Copy Fail vulnerability

On March 23, 2026, Theori researcher Taeyang Lee submitted a vulnerability report, CVE-2026-31431, codenamed "Copy Fail", to the Linux kernel security team, with a CVSS score of 7.8 (high risk). This vulnerability resides in the authencesn (Authenticated

Encryption with Associated Data Extended Sequence Number) encryption template of the kernel's crypto subsystem. It is a logical flaw vulnerability, not a traditional memory corruption type. Attackers can use this to achieve 100% stable local privilege escalation; the entire exploit requires only a 732-byte Python script.

In terms of the scope of impact, the root cause of the vulnerability can be traced back to commit 72548b093ee3 in January 2017, which introduced in-place decryption optimization of AEAD (Authenticated Encryption with Associated Data) in the algif_aead module. According to this calculation, the vulnerability has existed for more than eight years, covering almost all mainstream distributions since 2017, including Ubuntu 24.04 LTS, RHEL 8/9/10, Amazon Linux 2023, SUSE 16, etc. Antiy CERT has fully explained the basic situation, scope of impact, exploitation conditions, detection and forensics related measures of the vulnerability in the long document "CVE-2026-31431 (Copy Fail) Vulnerability FAQ"[1] released in the past two days. This article hopes to focus on analyzing the new division of labor between analysts and artificial intelligence in the era of highly mature generative large model technology.



The way Copy Fail was discovered is more paradigmatic than the vulnerability itself—it is neither the myth of "AI automatically finding" nor the traditional path of

"human manual auditing", but rather the product of precise collaboration between human researchers and AI platforms.

The process is divided into two closely linked phases. In the first phase, human insight defines the attack surface. Based on his experience with kernelCTF (Kernel Capture The Flag), Taeyang Lee identified the combined attack surface of the AF_ALG (Address Family Algorithm) encrypted socket interface and the splice() zero-copy system call, proposing the core hypothesis: "AF_ALG + splice is equivalent to opening a backdoor directly to the kernel's cryptographic subsystem for unprivileged users". In the second phase, AI enables large-scale scanning. Lee used the Xint Code AI code analysis platform to launch a targeted scan of the crypto subsystem. Xint Code completed a deep cross-file correlation analysis in approximately one hour, traversing all interaction paths between algif_aead and various encrypted templates, accurately locating logical flaws in the authencesn template.

At its core, Copy Fail is the result of the indirect interaction of three independent and seemingly reasonable code changes: the implementation of the Authencesn algorithm in 2011, the introduction of zero-copy support for Splice in 2015, and the in-place performance optimization in 2017. Each change, when examined individually, is flawless, but their combination produces fatal consequences—this "combination explosion" of vulnerability causes is precisely the scenario most difficult for traditional manual auditing to cover.

1.2 Establishment of a new human-machine paradigm

Copy Fail's findings reveal a new paradigm for security work: **without the insights of exceptional analysts, AI platforms will not know where to focus their scanning resources; without Xint Code, Lee could not have completed an exhaustive scan of all interaction paths in complex, large-scale code within a reasonable timeframe.**

This paradigm differs from two misconceptions. It doesn't fulfill the expectation of "AI independently discovering major vulnerabilities"—AI here demonstrates scalable execution

capabilities, not strategic directional judgment. It also indicates that with software complexity far exceeding human design capabilities—including the Linux kernel exceeding 30 million lines of code—relying solely on manual human auditing has reached a fundamental bottleneck. **The closed loop of "direction setting—scalable execution—result verification" formed between human researchers and AI tools represents a more realistic and efficient path forward**.

2. Five reasons for discovering Copy Fail and their implications for human-machine division of labor

2.1 Reason 1: The researcher's precise focus on prior knowledge is transformed into intuitive judgment

2.1.1 Taeyang Lee's systematic construction process of prior knowledge for drawing the AF_ALG attack surface in kernelCTF

The discovery of Copy Fail was not accidental, but **a direct extension of Theori researcher Taeyang Lee's knowledge of the AF_ALG attack surface accumulated in kernelCTF**. KernelCTF requires participants to have a deep understanding of the interaction boundaries of kernel subsystems. In this process, Lee systematically analyzed the call chain, permission model, and data flow path of the AF_ALG (Address Family ALG) interface. The Theori team has consistently won top CTF competitions such as DEF CON since 2013, and this high-intensity competitive training **has cultivated a keen awareness of asymmetric interfaces characterized by "low-privilege entry points and high-privilege kernel operations**".

Knowledge/Experience	Specific content	The path to transforming attack surface intuition	Contribution to the discovery of Copy Fail

kernelCTF Competition Practice	Systematically plot the complete attack surface of the AF_ALG socket, including code paths for socket creation, splice, sendmsg/recvmsg, etc.	Identify AF_ALG as an asymmetric interface with "low-privilege entry point and high-privilege operation" to determine if there are unaudited vulnerabilities in its interaction combination.	The search scope was narrowed from hundreds of subsystems to the crypto/ subsystem.
Deep Understanding of the splice() Zero-Copy Mechanism	splice() passes the page cache pages of a read-only file directly to the kernel subsystem by reference.	The read-only attribute of a page cached page may be reinterpreted within the target subsystem.	The core technical assumptions that constitute operator prompt
Scatterlist memory model knowledge	The kernel cryptosystem uses a scatterlist mechanism to manage non-contiguous physical pages.	The inconsistency between page writability and source attributes in a scatterlist can be exploited.	Instruct AI to focus on the security assumption gap between page origin and destination SGL writability.
CTF Attack Mindset	Systematic thinking on the path from least privilege to greatest impact	AF_ALG + splice was identified as a "backdoor that allows unprivileged users to pass through to the kernel cryptosystem".	Assuming precise targeting, AI can identify the most critical vulnerabilities in one hour of scanning.

The four layers of knowledge overlap and converge into a precise hypothesis: AF_ALG + splice constitute a hidden channel for unprivileged users to inject page cache pages into the kernel cryptosystem, and the attribute misalignment of the scatterlist page source is the most vulnerable link.

2.1.2 Formation and unique value of core assumptions

Lee's core assumption is essentially the identification of a security assumption break in the interaction of the three elements: splice() passes a read-only reference to the cached page, while the 2017 in-place optimization (commit 72548b093ee3) links the page into a writable

destination scatterlist, creating conditions for out-of-bounds writes to the authencesn template.

The remarkable aspect of this hypothesis lies in its precise directionality. It doesn't generalize to "audit the crypto/subsystem", but explicitly states that "splice() can pass a page cache reference of a read-only file to the cryptographic TX scattering list". This directionality is something AI cannot spontaneously generate—even if it grasps the independent semantics of AF_ALG and splice(), it cannot correlate them as a cross-subsystem attack surface. Xint Code's summary is remarkably precise: "A researcher identifies the attack surface, XC analyzes it". The core value of human researchers lies precisely in providing AI with an intuitive focus that cannot be algorithmically derived, directing its computational power towards code areas that truly possess vulnerability potential.

2.2 Reason 2: The scalability and associative capabilities of AI can generalize researchers' intuitive insights

2.2.1 Tasks that the AI platform can accomplish after receiving operator prompts, tasks that are impossible to perform using traditional auditing methods

Taeyang Lee hypothesized, based on kernelCTF experience, that the combination of AF_ALG sockets and splice() zero-copy could open a direct channel to the kernel cryptosystem for unprivileged users. However, translating this intuition into verifiable findings faces a scalability challenge.

The Linux kernel crypto subsystem consists of dozens of source files, multiple algorithm template layers, and asynchronous calls, involving multi-directional interactions between the AF_ALG socket layer, AEAD template layer, scatterlist (SGL) management, and page cache subsystem. Traditional Static Application Security Testing (SAST) relies on

predefined rules and pattern matching, which has two drawbacks: the rule base only covers known vulnerability patterns and cannot identify emergent defects where "multiple independent and reasonable designs combine to produce unreasonable results"; and it lacks cross-file semantic understanding capabilities, making it difficult to trace the complete semantic chain of the same operation, where it is "zero-copy reference" at the splice() layer, "in-place decryption optimization" at the algif_aead layer, and "HMAC verification of the first 4 bytes of the tag write operation" at the authencesn layer.

The Xint Code AI platform completed a full scan of the crypto subsystem in about one hour, breaking through three capability boundaries of traditional auditing tools: cross-file semantic association—establishing a secure semantic mapping between the in-place flag of algif_aead.c, the tag write logic of authencesn.c, and the page cache referenced by splice(); attack path combination reasoning—identifying the complete attack chain of AF_ALG→splice() zero copy→in-place decryption→pre-authentication write operation; and full subsystem coverage—discovering Copy Fail (CVSS 7.8, the highest severity in this scan) and other high-risk vulnerabilities in the coordinated disclosure stage.

2.2.2 The AI platform is capable of establishing secure semantic associations between multiple source files

The discovery of Copy Fail reveals the essential advantage of AI-assisted auditing: Large Language Models (LLMs) possess the ability to understand cross-document context at the semantic level, enabling them to identify "design compositional" flaws that traditional SAST cannot capture. The academic community refers to the space that traditional tools cannot cover as the "semantic gap"—a blind spot between fuzzy testing input coverage and known patterns in rule matching. Copy Fail is located at the core of this gap: the in-place optimization in commit 72548b093ee3 is locally reasonable, splice() zero-copy is standard behavior, and the label processing of authencesn conforms to AEAD specifications; independent review of these three aspects will not trigger SAST warnings.

Human auditors can theoretically uncover cross-component vulnerabilities through deep walkthroughs, but their cognitive capacity has a natural limit when analyzing interaction paths on the scale of crypto subsystems. The value of AI platforms is not to replace human

security intuition—Lee's attack surface assumption remains the logical starting point—but rather to scale and generalize human insights: single-point assumptions are expanded into area scanning, and "this might be problematic" is transformed into traversing a security semantic graph across dozens of source files. SAST cannot perform cross-file semantic reasoning, and human auditors struggle to exhaustively enumerate all subsystem paths; AI platforms precisely fill the structural gap between the two.

2.3 Reason 3: Human-machine collaboration can overcome the blind spots of traditional vulnerability analysis methods

2.3.1 The difficulty in detecting Copy Fail as a cross-subsystem logical defect

The Copy Fail (CVE-2026-31431) is not a traditional memory corruption vulnerability, but rather an emergent vulnerability. It stems from the convergence of three independent and seemingly reasonable kernel design decisions: in 2011, `authencesn` borrowed the target buffer as a temporary draft; in 2015, the AF_ALG interface was opened to non-privileged users; and in 2017, commit 72548b093ee3 introduced in-place optimizations. Each decision is valid individually, but their combination is fatal: when a user passes the page cache to the AF_ALG socket via `splice()`, the page cache is linked into a writable output scatterlist. During `authencesn` decryption, 4 bytes of temporary data are written, tampering with the page cache. The essence of this vulnerability is a mismatch in semantic interactions across subsystems, rendering traditional single-component detection methods almost ineffective.

2.3.2 Blind spot analysis of traditional vulnerability detection methods

Detection methods	Core principles	Unable to find the root cause of Copy Fail
Fuzzing	Random/mutated input triggers crash	It is difficult to construct precise inputs across the three subsystems of crypto, VFS, and splice; no anomalous signals are generated.
Memory security detection	Detecting memory out-of-	It does not involve memory corruption, and the write is semantically "legal"—the scatterlist is

(ASan/KASan)	bounds and UAF	marked as writable.
Static analysis (rule-based)	Predefined rules match dangerous patterns	Unable to identify cross-subsystem implicit association between splice() page cache references and authencesn temporary writes.
Traditional SAST tools	AST traversal and taint analysis	The granularity is limited to a single function or file, making it difficult to track data flows across splice.c, algif_aead.c, and authencesn.c.
Manual code review	Relying on the reviewer's experience	The crypto subsystem has hundreds of thousands of lines, making it difficult to cover all cross-subsystem paths.

Traditional detection methods all target local defect identification, while Copy Fail's exploitability is distributed across the interaction boundaries of the three subsystems. Fuzzing is expected to trigger a crash, but this vulnerability is written into the underlying layer and is completely "legitimate"; static analysis is expected to match dangerous patterns, but this combination is not in any known pattern library. This is the root cause of the vulnerability's lurking for nearly 9 years.

2.3.3 The essential differences in AI vulnerability analysis

Human-machine collaboration enabled a fundamental shift in the analysis paradigm. Taeyang Lee, based on his understanding of the kernelCTF attack surface, proposed the hypothesis that "the combination of AF_ALG and splice() opens a direct channel to the kernel cryptosystem for unprivileged users", a logical conclusion drawn from the intuitive understanding of cross-subsystem semantic associations. Subsequently, the Xint Code platform used LLM to scan the crypto subsystem, completing cross-file semantic association analysis in approximately one hour.

LLM's advantage lies in its semantic-level contextual understanding, rather than syntax tree traversal. LLM can simultaneously "understand" the zero-copy semantics of `splice()`, the interface contract of `AF_ALG`, and the buffer convention of `authencesn`, recognizing that "multiple independent and reasonable design combinations produce unreasonable results"—when `splice()` passes the page cache to the encrypted path, in-place optimization changes the page cache from "read-only input" to "writable output", and the

temporary write of `authencesn` is transformed into "unauthorized modification of the page cache". This recognition of the semantic tension between design intent and implementation is unattainable by traditional methods .^[14] It is this human-machine collaboration that brings emergent logical flaws like Copy Fail to the surface.

2.4 Reason 4: The maturation of multiple combination conditions

The discovery of Copy Fail is the result of the convergence of multiple factors in 2026, including computing infrastructure, the tipping point of AI capabilities, the accumulation of researchers' experience, and changes in the economics of attackers.

2.4.1 The development of computing power provides a foundation for AI-assisted vulnerability discovery

Large-scale language model (LLM)-assisted code analysis demands far more computing resources than traditional static analysis. The Linux kernel encryption subsystem comprises approximately 68,000 lines of C code, involving dozens of AEAD (Authenticated Encryption with Associated Data) templates and complex cross-file calls. The widespread adoption of GPU clusters and cloud computing makes it possible to complete a deep semantic scan of the entire subsystem within one hour.

2.4.2 The critical point for LLM understanding semantic complexity across subsystems by 2026

2026 marks a critical inflection point for LLM capabilities. The root cause of Copy Fail involves the interaction of three subsystems: the AF_ALG (Address Family Algorithm) socket interface, the splice() zero-copy mechanism, and the authencesn template. After receiving key observations from researchers, Xint Code completed a cross-file correlation analysis within approximately one hour, identifying the fatal interaction between the 2017 in-place optimization (commit 72548b093ee3) and out-of-bounds authencesn writes. Brumley points out that this breakthrough in cross-subsystem semantic reasoning

capabilities indicates that "the cost of discovering deep logical flaws may have decreased by about an order of magnitude".

2.4.3 Continuous accumulation of experience by specialized researchers and mature understanding of page cache vulnerability patterns

The domain expertise of human researchers forms the directional anchor for AI analysis. Theori researcher Taeyang Lee outlined the AF_ALG attack surface map in kernelCTF (Kernel Capture The Flag), based on which he formed the key hypothesis: the combination of AF_ALG and splice() creates a direct path to the kernel cryptosystem for unprivileged users.

The security community's understanding of page cache vulnerability patterns has matured over the past decade. In 2016, Dirty Cow (CVE-2016-5195) revealed the possibility of tampering with the page cache through a race condition copy-on-write mechanism; in 2022, Dirty Pipe (CVE-2022-0847) demonstrated a path to inject data through a pipe buffer. These two vulnerabilities accumulated experience in page cache attacks, enabling researchers to identify anomalies in page cache references within cryptographic subsystems.

2.4.4 Active exposure of side effects and changes in attacker economics

The three components of Copy Fail—the introduction of AF_ALG sockets in 2011, the in-place optimization of algif_aead in 2017, and the splice() zero-copy mechanism—were only linked as an attack chain after nearly a decade. The widespread adoption of containerization and cloud-native architectures has increased the frequency of these interface calls, naturally expanding the attack surface.

A deeper driving force stems from a structural shift in attacker economics. In cloud environments, the value of Local Privilege Escalation (LPE) vulnerabilities has

dramatically increased—page caches are shared between the host machine and containers, meaning a single modification can affect the entire node. Brumley described this as "worth a house". Meanwhile, HackerOne's Internet Bug Bounty project suspended operations in March 2026 due to a surge in reports generated by AI tools, reflecting an imbalance between discovery and remediation. This shift in incentive structure has driven more resources toward uncovering kernel-level logic flaws, leading to the discovery of Copy Fail vulnerabilities.

2.5 Reason 5: The "transparency" of disclosure methodss

2.5.1 Theories/Xint Code's exceptionally transparent disclosure strategy and its profound significance for the security community

In the discovery of the Copy Fail, the disclosure method was an independent determining variable. Theori/Xint Code's public disclosure on April 29, 2026, far exceeded industry norms: the official blog fully disclosed the root causes, including authentication scratch write and writable reference chains in the page cache, and provided a 732-byte standard library Python PoC. The disclosure pace was equally professional—reported on March 23, the fix patch (commit a664bf3d603d) merged into the main branch on April 1, and publicly released on April 29, with a 37-day coordinated disclosure cycle providing ample integration window for the release.

Table 2-1 Analysis of Transparency Strategy

Disclosure dimensions	Specific practices	Significance for safe communities	Comparison with traditional disclosure
Technical details disclosed	Full root cause revealed: scratch write, page cache writable reference chain	Defenders can understand the mechanism without reverse engineering, which guides the design of detection rules.	Traditional disclosures typically only provide CVSS scores and a version list.
PoC code	A 732-byte Python scr	The security team c	Most high-risk vu

is publicly available.	ipt has been successfully verified on Ubuntu, RHEL, SUSE, and other systems.	an directly reproduce and verify the effectiveness of the patch.	lnerabilities do not have publicly disclosed Proof-of-Concept (PoC) statements.
Impact verification public	The disclosure is divided into two parts: Part 1 Local Privilege Escalation, Part 2 Container Escape	Cloud-native teams can directly assess container cluster risks.	Traditional methods often lack version-specific verification.
Repair plan released	Temporary mitigation is provided concurrently: disabling algif_aead and limiting AF_ALG using seccomp.	The operations team can deploy mitigation before the patch.	Traditional disclosures often lack prior mitigation guidance.

The core characteristic of the table above is information symmetry—PoC grants both attackers and defenders equal exploitation capabilities, but the defender simultaneously gains complete understanding of the underlying mechanisms and temporary mitigation measures. Under traditional asymmetric disclosure, attackers often gain a "window of opportunity" lasting several weeks through reverse engineering patches, while the transparent strategy integrates the patch four weeks in advance, compressing the window to near zero.

2.5.2 The fact that it was exposed despite not being included in the "no-but-us" mechanism prompted an industry-wide defense effort

The disclosure of Copy Fail vulnerabilities has pioneered a new paradigm in cybersecurity. Its core value lies in breaking through the closed nature of traditional vulnerability distribution, particularly challenging the "no-but-us" vulnerability monopoly mechanism led by the United States. This mechanism is essentially a vulnerability hoarding system built by Western countries, especially the U.S. National Security Agency (NSA), through the Vulnerability Equivalence Procedure (VEP), aiming to control undisclosed vulnerabilities as strategic resources within intelligence agencies. Historical lessons show that this mechanism is extremely harmful: the Equation Group toolkit leaked by Shadow

Brokers in 2017 contained multiple Linux kernel zero-day vulnerabilities that the NSA had secretly held for over three years, directly leading to a global cybersecurity crisis. Even more alarming is that the U.S. National Security Memorandum No.28 of 2025 further lists "logic vulnerabilities discovered by AI" as "critical intelligence gathering capabilities", allowing them to be retained for up to 36 months without disclosure. This policy direction runs counter to global cybersecurity interests.

Theori/Xint Code's transparent disclosure strategy stands in stark contrast to the US approach, primarily in three key dimensions: In terms of time efficiency, it took only 37 days from vulnerability discovery to public disclosure, far shorter than the NSA's usual 540-day window; in terms of technical transparency, it not only fully disclosed the root cause of the vulnerability but also provided a verifiable 732-byte Python Proof-of-Concept (PoC), while the NSA's reports to the Linux Foundation often only contain vague descriptions; and in terms of international collaboration, it promoted simultaneous remediation by multiple countries, rather than limiting vulnerability information to the "Five Eyes" alliance as the US does. This difference is not only technical but also reflects fundamentally different cybersecurity governance philosophies.

From an industry perspective, this transparent disclosure model has multiple positive implications: First, it enabled the timely patching of a vulnerability affecting Linux Kernel 4.14+ that had been dormant for nearly nine years, with major global distributions completing security updates within days. Second, by disclosing technical details and Proof-of-Concept (PoC), it allows defenders to quickly develop detection rules and mitigation measures. Most importantly, it breaks the deadlock of information asymmetry regarding vulnerabilities, preventing them from becoming "digital weapons" for a few countries. The successful implementation of this model proves that even for high-risk vulnerabilities, transparent disclosure can balance security and timeliness.

3. Summary and reflections

3.1 The discovery of Copy Fail is the intersection of three elements

The fundamental reason why Copy Fail was discovered is the historic convergence of three evolutionary elements in 2026.

The first element is the continuous accumulation of human risk insight. Taeyang Lee, based on kernelCTF experience, formulated a precise hypothesis—"The combination of AF_ALG and splice opens a backdoor to the kernel encryption subsystem for unprivileged users"—pointing to a misalignment of the source attributes on the scatterlist page. Such directional hypotheses stem from practical attack surface intuition, something AI cannot spontaneously generate: LLMs lack a "feel" for exploit chains and cannot extract patterns from adversarial practice to transfer hypotheses. **Human prior knowledge plays the role of "direction setter**".

The second key element is AI's scalability. Xint Code completed a full cross-file correlation analysis of the crypto subsystem in about one hour, while manual auditing would take several weeks. LLM breaks through the "semantic gap" of traditional SAST, achieving cross-module semantic correlation. **The core value of AI is to exponentially generalize human insights in the code space—transforming "direction" into "result" on a large scale**.

The third element is the technological maturity tipping point. By 2026, LLM capabilities will have crossed the threshold of understanding the semantic complexity of the kernel across subsystems. Page cache vulnerabilities, after a decade of accumulation from Dirty Cow → Dirty Pipe → Copy Fail, have formed high-order patterns that AI can recognize. **The maturity of computing power and changes in attacker economics further reduce the cost of discovering deep logical flaws.**

Kernel vulnerability research is rapidly transitioning from an "artisan era" to a "human-machine collaborative era". Previously, discovering such cross-subsystem mismatch vulnerabilities required months of auditing by top experts; now, experienced engineers, with the help of AI, can complete equivalent analysis during a coffee break. This is not a rejection of human experts, but rather a restructuring of the work paradigm: **humans focus on "asking the right questions", while AI is responsible for "asking all relevant questions".**

3.2 Two different evolutionary paths of human-machine integration

Undoubtedly, both Mythos Preview (Anthropic) and Xint Code (Theori) are objects of attention and research for Chinese cybersecurity professionals. However, the significant difference in computing power between the two is crucial: while Xint Code has built its own 1000-calorie cluster, Anthropic possesses 100 to 200 times its peak computing power. Given this substantial computing power gap, and lacking the code productivity driven by Anthropic Opus, Xint Code relies more on engineers' accumulated vulnerability insights and deep knowledge of the system kernel, which to some extent offsets its shortcomings in computing power and resources. For Chinese cybersecurity researchers, whose resources are relatively scarce, Xint Code's approach is more instructive.

3.3 Shouts in battle

This shift presents challenges to the domestic security industry that go far beyond a simple question of whether to embrace AI. **Every technological revolution is both a catalyst for development and an amplifier of problems**. If we examine this from two perspectives—researchers (teams) and AI resources—we see that: the value of fundamental work such as low-level analysis, deep source tracing, and attribution has been neglected for a long time; the research community for low-level security analysis capabilities, kernel-driven analysis capabilities, and the integration of algorithms and system security is shrinking; and more and more security products are downplaying kernel-based defense, relying solely on ring 3 hooks to support perception and data collection—forming a trend of "weak kernelization" from human to machine. Of course, the vulnerability discovery capabilities, threat countermeasures capabilities, and effective product protection capabilities of the domestic security industry are not stagnant, but compared to the first-mover advantage and accelerated pace of strategic competitors, it is in a risky state of "not advancing is retreating, and slow progress is also retreating".

True human-machine collaboration in security analysis and research requires human stakeholders to aggregate experience, ask precise questions, guide generalization, and output insights. Insightful researchers and combat-ready teams are key elements of

capability and competition—those who can extract adversarial intuition from historical work and the experience of others, propose hypotheses like "AF_ALG + splice" at the technical level, and develop rigorous methodological frameworks at the tactical level. Researchers and teams that cannot ask the right questions, even with the most advanced AI platforms, will only waste computing power efficiently in a flawed space.

From the perspective of AI resources, even Xint Code's thousand-card cluster is still an unimaginable resource for domestic security teams. Although we have always insisted on "using engineers' insights and experience to offset the lack of computing power", experience and initiative cannot be separated from the most basic material foundation—a skilled cook cannot cook without rice.

China's cybersecurity industry cannot be caught up in the unrealistic fantasy that "artificial intelligence is a magic bullet" while simultaneously failing to secure resource allocations comparable to those of its strategic rivals. Nor can it fantasize that the arrival of AI will naturally offset the continued decline in underlying security analysis, kernel attack and defense, and system-level understanding, or compensate for the dwindling original achievements in in-depth analysis and source tracing, as well as security insights.

Writing the above at the end of a technical analysis is not out of pessimism or despair.

because:

Warriors always erupt from silence, and battles are always accompanied by shouts!

References

[1] CVE-2026-31431 (Copy Fail) Vulnerability FAQ (Part 1) - Basic Information, Scope of Impact and Handling of the Vulnerability [EB/OL]. (2026-04-30). https://mp.weixin.qq.com/s/MMeMCSPJ_IO4peFb94fXbw

[2] CVE-2026-31431 (Copy Fail) Vulnerability FAQ (Part 2) - Vulnerability Technical Mechanism, Historical Coordinates and Strategic Implications [EB/OL]. (2026-05-01). https://mp.weixin.qq.com/s/PX6KauylsXTihk3M_wCLgg

[3] Secure.com. Linux Kernel "Copy Fail" Zero-Day (CVE-2026-31431)[EB/OL]. (2026-04-29). https://www.secure.com/news/linux-kernel-copy-fail-zero-day-cve-2026-31431.

[4] XINT CODE. Copy Fail: A Critical Linux Kernel Vulnerability in authencesn[EB/OL]. (2026-04-29). https://xint.io/blog/copy-fail-linux-distributions.

[5] Kaspersky SecureList. Copy Fail: Linux Kernel Root Privilege Escalation[EB/OL]. (2026-04-29). https://securelist.com/tr/copyfail-root-linux/119634/

[6] CN-SEC Chinese website. In-Depth Analysis of the Copy Fail Vulnerability: The "Combination Explosion" Logic Defect in the Linux Kernel [EB/OL]. (2026-04-30). https://cn-sec.com/archives/5209927.html

[7] gitcode.csdn.net. Copy Fail Vulnerability Source Code Analysis [EB/OL]. (2026-04-30). https://gitcode.csdn.net/69f3f418825d1d76e8cf502e.html.

[8] Theori. Theori CTF Awards and Achievements[EB/OL]. (2026). https://theori.io/award.

[9] innovatopia.jp. Linux Kernel Copy Fail Vulnerability Technology Interpretation [EB/OL]. (2026-04-29). https://innovatopia.jp/cyber-security/cyber-security-news/99668/.

[10] GM7. Analysis of DeepSeek Copy Fail Vulnerability Reproduction [EB/OL]. (2026-04-30). https://www.gm7.org/archives/95671.

[11] ArXiv.org. Cross-Stage Semantic Vulnerabilities in Large Systems[EB/OL]. (2025-05-22). https://www.arxiv.org/pdf/2505.22052.

[12] ReversingLabs. Copy Fail: 5 YARA Rules for Detection[EB/OL]. (2026-04-30). https://www.reversinglabs.com/blog/copy-fail-5-yara-rules.

[13] GM7. Copy Fail Cross-Subsystem Semantic Vulnerability Technical Analysis [EB/OL]. (2026-04-30). https://www.gm7.org/archives/95674.

[14] CERT-EU. Security Advisory 2026-005: Copy Fail Vulnerability[EB/OL]. (2026-04-29). https://cert.europa.eu/publications/security-advisories/2026-005/.

[15] mediaTUM. Static Analysis Limitations for Cross-System Vulnerabilities[D/OL]. (2026). https://mediatum.ub.tum.de/doc/1774977/8xj90jmhe759q26867lisbqwn.thesis.pdf.

[16] Penligent. Claude Code Security Breaks the Old Model of Code Scanning[EB/OL].

(2026-04-29).

https://www.penligent.ai/hackinglabs/de/claude-code-security-breaks-the-old-mode

[17] SecurityWeek. Copy Fail Logic Flaw in Linux Kernel Enables System Takeover[EB/OL]. (2026-04-29).

https://www.securityweek.com/copy-fail-logic-flaw-in-linux-kernel-enables-system.

[18] Sysdig. CVE-2026-31431 Copy Fail: Linux Kernel Flaw Lets Local Users Escalate Privileges[EB/OL]. (2026-04-29).

https://www.sysdig.com/blog/cve-2026-31431-copy-fail-linux-kernel-flaw-lets-loca.

[19] BankInfoSecurity Asia. Linux "Copy Fail" Flaw Delivers Root-Level Access to Distros[EB/OL]. (2026-04-29).

https://www.bankinfosecurity.asia/linux-copy-fail-flaw-delivers-root-level-acces.

[20] Rare Earth Mining. In-Depth Analysis of Linux Kernel Copy Fail Vulnerability [EB/OL]. (2026-04-30).

https://juejin.cn/post/7634085400538316815.

[21] Cyber Technology Insights. HackerOne Stops Bug Bounty Program Over AI-Generated Report Flood[EB/OL]. (2026-03-18).

https://cybertechnologyinsights.com/cybertech-news/hackerone-stops-bug-bounty-pr.