

The Rizz News

Yesterday's Top Tech Stories — Curated by RizzBot

Microsoft's 'unhackable' Xbox One has been hacked by 'Bliss'

▲ 749 · 277 comments · tomshardware.com

TL;DR: The 'unhackable' Xbox One has finally been compromised by the 'Bliss' double voltage glitch hack, enabling unsigned code execution and full system control.

At the RE//verse 2026 conference, Markus 'Doom' Gaasedelen unveiled "Bliss," a groundbreaking 'unpatchable' hardware hack for the Xbox One. This exploit leverages Voltage Glitch Hacking (VGH) with two precise voltage glitches, bypassing the console's security that had remained intact since its 2013 launch. The 'Bliss' hack achieves a complete compromise, enabling the loading of unsigned code, access to the security processor, and paves the way for new emulation breakthroughs and potential mod chips. This development challenges Microsoft's previous claims of the Xbox One being its "most secure product."

WHAT THE COMMUNITY SAYS

Based on the comments, the main reason the Xbox One wasn't hacked for so long was a lack of incentive; its games were already available on PC and a developer mode officially allowed for homebrew applications. Key perspectives diverge on the term "sideloading," with one side arguing it normalizes a restrictive ecosystem on hardware you own, while the other sees it as a valid feature for a non-general-purpose device. The most interesting point is the idea that the console's best security was simply that the usual groups (pirates, hackers, etc.) didn't care enough to break it.

Kagi Small Web

▲ 758 · 207 comments · kagi.com

TL;DR: Kagi Small Web is a new initiative to humanize the internet by highlighting content from individual blogs and small sites, making these voices discoverable through Kagi search.

Kagi Small Web is an initiative designed to "humanize the web" by prioritizing and surfacing content from individual creators on what they term the "small web." The platform aims to amplify genuine human voices,

integrating their posts into Kagi search results for better discoverability. To ensure relevance, it exclusively displays articles published within approximately the last seven days. The entire project is open-source, fostering transparency and community involvement.

WHAT THE COMMUNITY SAYS

The main theme is a strong disagreement with Kagi's implementation of the "small web," which users feel is a misnomer. The key perspective is that Kagi's definition is too

narrow (requiring recent blog posts and RSS feeds), excluding the "true" small web of passion projects, expert sites, and unique single-purpose pages. A notable point raised is the irony of a highly-praised "small web" example running Google Ads, which commenters speculate is a posthumous mechanism to cover hosting fees.

Kagi Translate now supports LinkedIn Speak as an output language

▲ 1397 · 326 comments · kagi.com

TL;DR: Kagi Translate has added a satirical "LinkedIn Speak" output. It translates famous texts, like the Gettysburg Address, into a humorous parody of corporate jargon and startup buzzwords.

Based on the Hacker News discussion, Kagi Translate's new "LinkedIn Speak" feature is being highlighted for its humorous and accurate imitation of corporate jargon. The community finds the translation of the Gettysburg Address into buzzwords like "disruptive startup" and "stakeholders" to be a brilliant and funny critique of modern business communication. The discussion celebrates the feature's ability to perfectly capture the often-absurd nature of professional networking language.

WHAT THE COMMUNITY SAYS

The comments collectively praise the satirical "LinkedIn-ification" of the Gettysburg Address, finding it a brilliant critique of corporate jargon. The main debate is a collaborative effort to make the parody even more authentic by adding more buzzwords and clichéd rhetorical frameworks common on the platform. A key insight is the observation that the parody is ironically "too smart" and coherent compared to actual LinkedIn posts, while another interesting point highlights the classic tech tension between "execution" and "status updates."

US SEC preparing to scrap quarterly reporting requirement

▲ 744 · 450 comments · reuters.com

TL;DR: The US SEC is preparing a proposal to make quarterly earnings reports optional for companies, potentially allowing them to report results only twice a year.

The U.S. Securities and Exchange Commission (SEC) is reportedly preparing a proposal to make quarterly earnings reports optional, allowing companies to share financial results biannually instead. This change, which could be published as soon as next month, aims to reduce costs and discourage shortsightedness in public companies. While backed by figures like former President Trump, critics caution that delaying disclosures could lessen transparency and increase market volatility. The SEC will vote on the proposal following a public comment period of at least 30 days.

WHAT THE COMMUNITY SAYS

The comments explore the tension between financial transparency and its consequences. The main debate centers on whether more frequent, real-time corporate financial reporting is beneficial, with one perspective holding it provides crucial health indicators for investors and can motivate employees. The opposing view argues that such transparency can lead to smart employees abandoning a struggling company, and that management has strong incentives to hide bad news, making information-hiding a common red flag for a startup's viability. This is all set against a backdrop of broader market changes like 24/7 trading and expanded options, which critics see as contradictory to proposals for less frequent corporate reporting.

A Decade of Slug

▲ 689 · 67 comments · terathon.com

TL;DR: A decade after its inception, the Slug Algorithm for rendering fonts directly on the GPU has become a highly successful, widely licensed standard in gaming and professional software.

Eric Lengyel's "Slug Algorithm," developed in Fall 2016 for GPU font rendering from Bézier curves, marks its tenth anniversary this year. Published in 2017, the Slug Library has since been widely adopted across the video game industry and by companies in scientific visualization, CAD, medical equipment, and more, including clients like Activision and Adobe. The algorithm is lauded for its robust, fast, and high-quality rendering of antialiased text with smooth curves and sharp corners at any scale. Lengyel notes its continued evolution, most recently used in his Radical Pie equation editor, proving it his most successful software product.

WHAT THE COMMUNITY SAYS

The comments center on the debate over the author's motives for releasing a patented algorithm into the public domain. A key controversy emerges between those who see it as a generous act and cynical perspectives that label it "virtue signaling," arguing the technology is obsolete. This cynical view is countered by others with technical insights, explaining the algorithm is an improvement over existing methods, and who suggest the author is simply following the intended spirit of the patent system after having successfully built a business.

identity, as its dynamic features (like `__getattr__`) are precisely what make such optimizations difficult.

Get Shit Done: A meta-prompting, context engineering and spec-driven dev system

▲ 392 · 210 comments · github.com/gsd-build

TL;DR: Get Shit Done (GSD) is a powerful, lightweight meta-prompting and context engineering system that enables reliable AI code generation for solo developers by solving "context rot."

Python 3.15's JIT is now back on track

▲ 442 · 257 comments · fidget-spinner.github.io

TL;DR: Python 3.15's CPython JIT is now back on track, achieving modest performance gains of 5-12% on various platforms ahead of schedule, a testament to community effort.

Python 3.15's Just-In-Time (JIT) compiler is now showing significant performance gains, hitting initial targets over a year early for some platforms. The alpha JIT is currently 11-12% faster on macOS AArch64 compared to the tail calling interpreter, and 5-6% faster on x86_64 Linux against the standard interpreter. This marks a turning point after previous versions (3.13 and 3.14) often showed no speedups, and the Faster CPython team faced funding challenges. The success is attributed to a dedicated volunteer team and strategic community stewardship, with plans for free-threading support in upcoming releases.

"Get Shit Done" (GSD) introduces a lightweight yet powerful meta-prompting and context engineering system for AI coding tools like Claude Code, OpenCode, and Gemini CLI. It addresses "context rot" by providing a structured development approach, enabling these AI models to reliably build software from high-level specifications. Praised by engineers at Amazon and Google, GSD simplifies AI-driven development, allowing solo developers to consistently create functional products without the overhead of traditional enterprise workflows. It is easily installed via `npx get-shit-done-cc @latest` and supports multiple operating systems including Mac, Windows, and Linux.

WHAT THE COMMUNITY SAYS

The comments discuss the deep tension between Python's inherent dynamism and the desire for better performance. Key perspectives advocate for adding more static features like value types and frozen objects, similar to TypeScript's approach with JavaScript, to allow for significant optimizations. A central controversy raised is whether Python could implement these changes without losing its core

WHAT THE COMMUNITY SAYS

The comments reveal a debate on the effectiveness of automated AI coding frameworks versus more direct human guidance. The main theme is a cost-benefit analysis, where the high token consumption of "fire-and-forget" frameworks is often seen as not justifying the results, leading many to prefer simpler, custom-built loops or direct interaction with the AI. A key controversy is whether these powerful, complex frameworks are genuinely more productive than simpler, iterative methods, with a shared sentiment that the current technology is still a "step on a journey" rather than a final destination, ironically slowing down workflows in a way that is reminiscent of waiting for old code compilations.