Strict Audit Report: Structural Vulnerabilities of Autonomous AI Tight Coupling in the WordPress Ecosystem and Verification of Mythos-Class Threats

1. Introduction: The Clash Between Deterministic System Monoculture and Non-Deterministic AI

This report strictly audits the series of claims regarding cybersecurity, system architecture, and compliance presented in the article "Claude Mythos — Delving into the WordPress + AI Problem," and delves into their technical validity and underlying structural risks.

The central thesis underlying the target article is that "tightly coupling non-deterministic autonomous AI agents with the legacy architecture of WordPress, which exists in a state of monoculture, creates extremely catastrophic vulnerabilities." This phenomenon is structurally isomorphic to the risks posed by tightly coupling legacy infrastructure (COBOL) and AI (Copilot) in the financial system. However, the article suggests that because the web ecosystem is directly exposed to the external internet, the consequences could be even more fatal.

Based on extensive threat intelligence from 2024 to 2026, CVE data registered in the National Vulnerability Database (NVD), the latest OWASP framework for AI agents, and global e-commerce statistical data, the claims made in the article are quantitatively and qualitatively highly accurate. In fact, the situation is deteriorating far more rapidly than the article describes, and the emergence of next-generation vulnerability discovery AI (Mythos-class) is completely dismantling the fundamental cybersecurity paradigm of "reactive patch management."

This report will not only list superficial threats but will also deeply demonstrate the root causes of why this architecture is inherently indefensible.

2. Quantitative Facts and Limits of the Web Monoculture

The target article points out that WordPress has formed a "monoculture on the same scale as Microsoft's Office monopoly." We verify the accuracy of this quantitative premise and the risks this monoculture poses to the entire system.

2.1 Overwhelming Market Dominance and Ecosystem Bloat

According to the latest statistics from W3Techs as of April 2026, WordPress powers 42.5% of all

websites worldwide and holds a dominant 59.8% to 60.2% share in the Content Management System (CMS) market.[1] The figures of "about 43%" and "60% in the CMS market" mentioned in the article are completely accurate within the margin of statistical error.

CMS Platform	Jan 2015 Share	Apr 2026 Share	Trend	Citation
WordPress	60.7%	59.8%	Stable dominance	[3]
Shopify	0.7%	7.2%	Rapid growth	[3]
Wix	0.3%	6.0%	Rapid growth	[3]
Squarespace	0.5%	3.5%	Growth	[3]

Correlating this 42.5% figure with NetCraft's web survey from February 2026 (which measured approximately 1.42 billion hostnames globally) confirms that the number of sites running WordPress has reached a staggering scale of up to 605 million.[1]

This massive monoculture is supported by a vast array of third-party plugins and themes.[1] For example, the page builder "Elementor" boasts over 10 million active installations alone and serves as the foundational infrastructure for approximately 13% of all websites globally.[5] Currently, Elementor has natively integrated generative AI features through "Elementor AI".[7] This means that millions of websites have abruptly transformed into "AI agent execution environments," regardless of the users' intent or technical understanding.

2.2 Exponential Explosion of Vulnerabilities and Collapse of Defense Lines

The risk of a monoculture is that a single pathogen (vulnerability) can devastate the entire system. Data from security firm Patchstack's 2026 State of WordPress Security report indicates that this ecosystem has already lost its self-cleansing ability.

Vulnerability Metric	2024 Actual	2025 Actual	Change / Details	Citation
New Vulnerabilities	7,966	11,334	+42.3% YoY	[8]

Total Tracked	N/A	64,782	Highest recorded	[9]
Plugin Origin	N/A	97.0%	Core is only 0.2%	[9]

As of 2026, security databases track a total of 64,782 vulnerabilities across the WordPress ecosystem, representing the most comprehensive vulnerability intelligence ever compiled for any single content management system.[9] Even more concerning is the fact that 46% of plugin vulnerabilities do not receive a patch by the time they are publicly disclosed, leaving them as zero-days.[10]

The "dysfunctional defense" pointed out in the article has also been empirically proven by Patchstack's infrastructure tests. When executing 11 known vulnerability exploits against Web Application Firewalls (WAF) and server-side security environments provided by five major hosting providers, the hosting defense layers failed to block 87.8% of the attacks, allowing them to penetrate the systems.[11] This clearly proves that traditional signature-based security measures are entirely ineffective against modern complex plugin chains and the prompt injections discussed later.

3. The Original Sin of Architecture: Non-Sandboxed Environments and Tight AI Coupling

The sharpest insight of the target article is its identification of the structural danger of directly coupling an LLM (a non-deterministic system) to the foundational architecture of WordPress (a deterministic system). This problem is not merely a "software bug" but stems from a "design original sin" at the architectural level.

3.1 Absence of Privilege Separation (No Sandbox Architecture)

In modern secure computing, each process is executed within a strict sandbox (like an "Isolate"), and access to databases or file systems is restricted to permissions explicitly granted via a capability manifest.[13]

However, the WordPress architecture lacks this sandbox structure.[13] Once a plugin is installed and activated, its code executes within the exact same PHP process as the WordPress core.[13] Through the global object $wpdb, plugins unconditionally acquire full CRUD (SELECT, INSERT, UPDATE, DELETE) privileges over the underlying MySQL/MariaDB database.[14]

In this flat privilege model, if a vulnerability exists in a single minor plugin, it immediately becomes an unrestricted access route to the entire site—including administrator hashes in the wp_users table and site settings in wp_options.[14] Because there is no internal sandbox, if a part of the application layer is breached, the entire system falls.[13]

3.2 Catastrophic CVEs Brought by AI Plugins

Into this defenseless "non-sandboxed environment," a massive number of AI plugins have been introduced to communicate with external LLMs (OpenAI, Claude, Gemini, etc.) and perform autonomous content generation and database operations.[16] For instance, "AI Engine" has over 100,000 active installations, automating advanced chatbots and content generation workflows.[16]

The target article correctly points out severe vulnerabilities in the "AI Power" (now AI Puffer) plugin, which is entirely corroborated by records in the NVD and CISA Vulnerability Bulletins.[18]

- **CVE-2024-10392 (CVSS Base Score: 9.8 - Critical):** Unauthenticated arbitrary file upload vulnerability. A lack of file type validation allowed external attackers to upload arbitrary PHP scripts (Web shells) to the server without credentials, leading to Remote Code Execution (RCE).[19]
- **CVE-2025-0586 / CVE-2025-0428 (CVSS Base Score: 7.2 - High):** PHP Object Injection vulnerability. Flaws in the deserialization of untrusted input allowed authenticated attackers to inject arbitrary PHP objects.[18]

The true state of the current web monoculture is that AI agents harboring these fatal vulnerabilities sit at the core of what should be a robust deterministic CMS, possessing full database access privileges.

3.3 EchoLeak and the Invisibility of Semantic Attacks

Traditional cybersecurity assumes determinism. A specific input (e.g., SQL injection like ' OR 1=1) will predictably be blocked by WAF access control rules. However, an LLM is a probabilistic system.

The article references the "EchoLeak (CVE-2025-32711)" vulnerability in Microsoft 365 Copilot, reported by Trend Micro in July 2025, to present a new attack vector brought about by these non-deterministic systems.[21] EchoLeak is a "zero-click AI vulnerability." Attackers embed malicious prompts via invisible prompt injection—such as HTML comments or white-on-white text—within an email.[21] Even if the user clicks nothing, the moment Copilot's RAG (Retrieval-Augmented Generation) engine reads that email as context for summarization or search, the attacker's instructions are executed, and data is exfiltrated.[21]

In a WordPress environment, comment sections and contact forms act as "invisible prompt injection portals" open to the entire internet. If an attacker writes a malicious prompt in the comment section, and the site administrator later uses an AI plugin to filter spam or summarize customer feedback, the AI reads that text from the database and its control is hijacked. Because the plugin is not sandboxed, the hijacked AI agent manipulates $wpdb as a legitimate process, dumping information or generating SEO spam en masse. This is a "semantic time bomb" that WAFs cannot detect.

4. Threat of Cascading Failures Based on OWASP Agentic Applications Top 10

The unique risks of AI agents were systematized in the "OWASP Top 10 for Agentic Applications 2026," published in December 2025.[22] This framework shifts the focus from the passive risks of LLMs to the active behavioral risks of "agents" that hold goals and manipulate tools.[22]

The article's analysis perfectly aligns with this OWASP framework, demonstrating that the following threats manifest compoundingly in the WordPress AI plugin environment.

Threat ID	Threat Name (OWASP Definition)	Specific Manifestation in WordPress Environment	Citation
ASI01	Agent Goal Hijack	Attackers rewrite the agent's decision path via external input (e.g., comments), transforming a support bot into a data extraction tool.	[22]
ASI02	Tool Misuse and Exploitation	Legitimate tools granted to the agent (CMS API, file access) are abused for data exfiltration or malware deployment.	[22]
ASI06	Memory & Context Poisoning	Malicious data is injected into the database serving as the RAG source, permanently poisoning all subsequent autonomous	[22]

		decisions.	
ASI08	Cascading Failures	A single fault or breach propagates across agents, tools, and workflows, causing system-wide impact.	[23]

4.1 ASI08: The True Terror of Cascading Failures

The most destructive consequence among these is "ASI08: Cascading Failures".[22] While traditional software failures remain localized crashes, agentic AI cascades propagate autonomously across agents, amplify through feedback loops, and compound into system-wide catastrophes.[25] Furthermore, they propagate at speeds orders of magnitude faster than human reaction times.[25]

In modern web operations, WordPress is not an isolated island. Many systems are intricately linked to mail servers, CRMs (e.g., Salesforce), social media, and cloud storage via iPaaS like Zapier or Make.

Applying OWASP's cascading failure scenario to WordPress reveals its devastating nature. If a single AI agent in WordPress is hijacked via a prompt injection (ASI01), the impact does not stay within the CMS. The compromised agent bypasses boundaries using the ambiguity of natural language interfaces and initiates lateral access to connected external tools. It can illicitly trigger a Zapier workflow to erase all customer data in the CRM, send phishing emails to all customers from an authenticated corporate email account, and encrypt backup data in Google Drive—all executed as a "normal chain of tool operations".[25] A breakthrough at a single point leads to the collapse of all connected services. As the article points out, this is exactly the same architectural flaw as the Copilot problem in financial systems.

5. E-commerce and PCI DSS v4.0.1: Structural Contradictions in Payment Infrastructure

The risk of cascading failures and data exposure inflicts the most severe damage in the e-commerce sector. Specifically, the integration of AI agents with "WooCommerce," which holds a massive share of the global e-commerce market, creates an unresolvable contradiction with global compliance standards.

5.1 WooCommerce's Massive Economic Sphere and Accumulation of Confidential Data

WooCommerce is an e-commerce platform that runs as a free WordPress plugin, yet it operates on approximately 6.5 million active websites.[26] It holds a dominant 33% to 39% market share among e-commerce platforms, accounting for 28% of all retail e-commerce sales globally, and processes an estimated Gross Merchandise Volume (GMV) of $30 to $35 billion annually.[26]

Unlike Shopify, which is an independent SaaS, WooCommerce is installed directly into the WordPress ecosystem.[28] Therefore, highly sensitive Personally Identifiable Information (PII) such as customer names, shipping addresses, email addresses, and order histories are all accumulated in the same single database as the WordPress core.[15]

When AI plugins are introduced here, the AI agents interact directly with this database to answer user inquiries or analyze sales data.

5.2 Fundamental Incompatibility with PCI DSS v4.0.1

The operational principles of current LLMs are logically incompatible with the requirements of the global credit card security standard, PCI DSS (Payment Card Industry Data Security Standard) v4.0.1.[29]

1. Requirement 7: Absence of the "Need-to-Know" Principle PCI DSS Requirement 7 strictly dictates that access to system components and cardholder data must be restricted to individuals whose jobs require such access via a "business need to know".[30] However, LLMs do not possess this concept of "need-to-know." To generate accurate responses, AI models tend to scan wide swaths of the given database as context vectors. Consequently, AI agents indiscriminately exercise access rights over payment metadata and customer PII that should otherwise be isolated, despite having no business need to do so. This is a clear system-level violation of Requirement 7.

2. Requirement 8: Loss of Identification, Authentication, and Audit Trails PCI DSS Requirement 8 mandates assigning unique IDs to each person or process with access, while Requirement 10 demands logging and monitoring all access to system components and cardholder data.[30] If an AI agent is hijacked via prompt injection (ASI01) and executes unauthorized data extraction or fraudulent refunds, the access logs will only show that a "legitimately authenticated AI plugin (or its underlying PHP process) executed the task." The audit trail needed to identify the true attacker is logically lost within the unstructured data of natural language. A payment system connected to an unauditable autonomous process is unacceptable from a compliance standpoint.

Even if merchants use PCI-compliant external payment gateways like Stripe and do not store credit card numbers on their own servers, their site environments still function as the front end for payments and remain within the PCI DSS scope.[31] As long as AI agents are tightly coupled to the e-commerce CMS, operators are effectively installing an "unauditable autonomous backdoor" themselves.

6. Manifestation of Mythos-Class Threats: Collapse of Patch Speed Asymmetry

The vulnerabilities discussed above are powder kegs that have long existed within the WordPress ecosystem. The definitive igniter for them is the arrival of next-generation vulnerability discovery AI, the "Mythos-class."

6.1 The Shock of Claude Mythos Preview and Project Glasswing

"Claude Mythos Preview," publicly acknowledged by Anthropic on April 7, 2026, fundamentally upended the concepts of AI cyber offense and defense.[32] Mythos Preview has acquired the ability to autonomously scan systems without human direction, discover "zero-day vulnerabilities" that are extremely difficult for humans to find, and automatically chain them together to generate working exploits.[33]

In evaluations by Anthropic's Frontier Red Team, this model achieved astonishing results:

- **OpenBSD Vulnerability Discovery:** Discovered a 27-year-old remote crash (integer overflow) vulnerability in OpenBSD, one of the world's most robust, security-hardened OSes used for firewalls and critical infrastructure.[33]
- **FFmpeg Vulnerability Discovery:** Discovered a 16-year-old out-of-bounds write flaw in FFmpeg—the de facto standard for video processing—that automated testing tools had missed even after scanning five million times.[33]
- **FreeBSD Remote Code Execution:** Autonomously identified and fully exploited a 17-year-old remote code execution flaw in FreeBSD's NFS server, granting unauthenticated root access without any human involvement after the initial prompt.

Recognizing that this AI model could cause catastrophic fallout to the digital economy if it fell into the hands of cybercriminals, Anthropic decided not to release it publicly.[33] Instead, they launched an urgent defense-only consortium called "Project Glasswing" with major infrastructure companies like AWS, Apple, Microsoft, Google, and CrowdStrike.[36]

6.2 "70 Days vs. 5 Hours": The End of the Defense Paradigm

The "patch application speed asymmetry" pointed out in the article is the most devastating indicator that modern cyber defense has structurally failed.

- **Defense Speed (70 days):** Data from the IBM Cost of a Data Breach Report and CrowdStrike shows that the median time it takes for an enterprise to detect a vulnerability, contain an incident, and apply a patch (the patch window) is approximately 70 days.[37]
- **Offense Speed (5 hours):** Conversely, according to Patchstack data, the median time from the public disclosure of a high-impact WordPress vulnerability to the start of mass exploitation by botnets is now merely 5 hours.[10]

Adding Mythos-class autonomous zero-day discovery capabilities to this overwhelming temporal asymmetry of "70 days vs. 5 hours" is devastating. Mythos-class models can scan tens of millions of lines of open-source code (the WordPress core and its 61,000+ plugins) in parallel at lightning speeds. The discovery of unknown zero-day vulnerabilities and the construction of exploit chains, which would take human researchers years, are completed by AI overnight.[39] Currently, 99% of the vulnerabilities discovered by Mythos remain unpatched.[39]

While financial institutions have 24/7 Security Operations Centers (SOCs) and closed-network environments, the hundreds of millions of small businesses using WordPress do not have dedicated security personnel. Their systems are directly exposed to the internet, their WAFs let 87.8% of attacks through, and their plugins are not sandboxed.[12]

If a Mythos-class AI is leaked to malicious state actors or open-source equivalents are released, a single discovered zero-day vulnerability instantly becomes a fully automated attack weapon against hundreds of millions of sites. The traditional reactive approach of "patching after an attack" is completely broken.

7. Architectural Solutions: Static Generation and Isolating AI in Development

In response to the severity of this situation, the target article poses a fundamental question: "Is a dynamic CMS like WordPress really necessary?" and advocates for a massive architectural shift. We evaluate the technical validity of this approach.

7.1 Dismantling Dynamic Components and Depleting the Attack Surface

WordPress requires PHP and MySQL primarily to handle dynamic processes (mostly comment sections and contact forms). However, in the 2026 web ecosystem, discussions have largely moved to external platforms like X, Reddit, and Discord. The remaining comments on proprietary sites are almost entirely SEO spam bots.

If comment functions and contact forms (which can be outsourced to SaaS like Google Forms or Formspree) are decoupled from the core site, the system no longer needs PHP processes or a MySQL database.

The architecture recommended in the article—building with Static Site Generators (Hugo, Astro, Next.js) or a self-contained setup using Python and Nginx—is the most robust defense strategy. By eliminating dynamic processes and serving only static files (Jamstack architecture), the attack surface is reduced to the absolute minimum, or practically eliminated.

Without a SQL database, SQL injections are impossible. Without PHP processes, remote code execution (RCE) is impossible. Most importantly, without an AI plugin running in production, the physical entry point for prompt injections like EchoLeak does not exist. No matter how advanced a Mythos-class AI's zero-day discovery capabilities are, an exploit is theoretically

impossible without dynamic processes or a tightly coupled complex state machine to attack. This "Security by Simplicity" is the only architectural barrier capable of withstanding next-generation AI attacks.

7.2 Claude Code Proves the Principle of "Deterministic Deployment"

The article's guiding principle—"Use AI as a development tool, do not put it inside the product. Do not make it autonomous. Humans should hold the structure and make the final decisions"—is remarkably validated by the architecture of Anthropic's own latest AI coding tool, "Claude Code."

Claude Code is an agentic coding tool that operates directly in the developer's terminal, reads the entire codebase, and autonomously executes coding, testing, and debugging.[40] However, this highly autonomous AI never runs in a "production server" waiting for end-user input.

Claude Code operates strictly in an isolated "Sandboxed Dev Environment" locally or in a CI/CD pipeline.[40] With its checkpointing system, the code state is saved before changes, allowing humans to review and rewind.[41] For GitHub pull requests, it employs a confirmed=true parameter, ensuring that only changes explicitly confirmed and approved by humans are reflected.[42]

In short, the AI generates code under human supervision, and only the "deterministic program" (static files or compiled code) that passes tests and human review is ultimately deployed to the production server. Under this structure, there is zero room for external internet prompt injections to hijack a production AI agent.

The target article's conclusion to "use AI as a tool, not as the system's core subject" perfectly aligns with the safety design philosophy Anthropic adopted when implementing its most powerful models.

8. Conclusion: Leaving the Monoculture as a Civilizational Design Philosophy

As a result of rigorous auditing and multi-faceted threat intelligence analysis, we conclude that the logic and warnings presented in the article "Claude Mythos — Delving into the WordPress + AI Problem" are completely accurate and capture an ongoing, extremely severe crisis without exaggeration.

1. **Full Corroboration of Quantitative Data:** All quantitative metrics cited in the article—such as WordPress's 42.5% market share, the 11,334 vulnerability explosion in 2025, and WooCommerce's $30 to $35 billion GMV—perfectly match the latest reliable data sources.
2. **Proof of Fundamental Architectural Flaws:** Tightly coupling non-deterministic probabilistic models (AI plugins) into a non-sandboxed legacy PHP environment causes the "Cascading Failures (ASI08)" warned by OWASP and triggers unauditable access

privilege violations strictly prohibited by PCI DSS v4.0.1, fundamentally destroying business continuity.

3. **Manifestation of "The Death of Defense" via Mythos-Class AI:** The ability of Claude Mythos Preview to autonomously discover and exploit unknown OS vulnerabilities has shattered the premises of cybersecurity. While defenders take "70 days" to patch, attackers begin mass exploitation within "5 hours" of disclosure. With ultra-fast AI added to this asymmetric timeline, hundreds of millions of WordPress sites exposed to the internet are destined to fall defenselessly.

Final Verdict

In any field—software, agriculture, or finance—a monoculture created in the extreme pursuit of "convenience" builds a system inherently fragile to specific shocks. The industry trend of tightly coupling an uncontrollable black box like generative AI to the massive WordPress monoculture equates to entirely abandoning system safety and resilience for the sake of maximizing revenue and convenience.

The solutions derived by the article—"splitting functions into self-contained units to ensure loose coupling," "eliminating the attack surface through static generation," and "isolating AI as a development tool rather than a production agent"—are not mere technical best practices. In an era where autonomous attack AIs of the Mythos class will become commonplace, these are the *only logical system design philosophies* capable of protecting digital infrastructure from collapse. Organizations and developers must treat this warning not as a technical debate, but as a civilizational challenge concerning the survival of their business, and initiate architectural transitions immediately.

引用文献

1. How Many Websites Use WordPress in April 2026? WordPress Statistics - WPZOOM, 4月 12, 2026にアクセス、https://www.wpzoom.com/blog/wordpress-statistics/
2. Usage statistics and market shares of content management systems - W3Techs, 4月 12, 2026にアクセス、https://w3techs.com/technologies/overview/content_management
3. Market share trends for content management systems, April 2026 - W3Techs, 4月 12, 2026にアクセス、https://w3techs.com/technologies/history_overview/content_management
4. Market share yearly trends for content management systems, March 2026 - W3Techs, 4月 12, 2026にアクセス、https://w3techs.com/technologies/history_overview/content_management/ms/y
5. Elementor - Wikipedia, 4月 12, 2026にアクセス、https://en.wikipedia.org/wiki/Elementor
6. Elementor Usage Trends Report (2025) - EagleEdge Marketing, 4月 12, 2026にアクセス、https://eagleedgemarketing.com/elementor-usage-trends-report-2025/
7. 10 Best WordPress AI Plugins in 2026 - Elementor, 4月 12, 2026にアクセス、

https://elementor.com/blog/wordpress-ai-plugin/
8. WordPress Ships Three Security Patches in 24 Hours as Exploits Hit the Wild - 365i, 4月 12, 2026にアクセス、https://www.365i.co.uk/news/2026/03/12/wordpress-three-security-patches-24-hours/
9. WordPress Plugin Security Audit 2026: How To Find And Fix Vulnerable Plugins - WebHostMost Blog, 4月 12, 2026にアクセス、https://blog.webhostmost.com/wordpress-plugin-security-audit-guide-2026/
10. AI WordPress Security: How Agencies Can Protect Client Sites in 2026 - WP Umbrella, 4月 12, 2026にアクセス、https://wp-umbrella.com/blog/ai-wordpress-security/
11. Hosting security tested: 87.8% of vulnerability exploits bypassed hosting defenses, 4月 12, 2026にアクセス、https://patchstack.com/articles/hosting-security-tested-87-percent-of-vulnerability-exploits-bypassed-hosting-defenses/
12. 234 - WordPress 6.9 Release Squad, Voting Open For WP Accessibility Team Reps, 4月 12, 2026にアクセス、https://wp-content.co/newsletter/archive/234/
13. EmDash: A Full-Stack TypeScript CMS Built on Astro + Cloudflare — Can It Replace WordPress?, 4月 12, 2026にアクセス、https://recca0120.github.io/en/2026/04/07/emdash-cms-astro-cloudflare/
14. Wordpress Plugin Security Model - infosec4breakfast, 4月 12, 2026にアクセス、https://pwnage.io/wordpress-plugin-model/
15. How to Secure a WordPress Database: 10 Methods - InstaWP, 4月 12, 2026にアクセス、https://instawp.com/how-to-secure-a-wordpress-database/
16. Best AI Plugins for WordPress in 2026: Work Smarter, Not Harder - Purethemes, 4月 12, 2026にアクセス、https://purethemes.net/best-ai-plugins-for-wordpress-work-smarter-not-harder/
17. Best WordPress AI Tools to Optimize and Scale Content 2026 - AI Growth Agent, 4月 12, 2026にアクセス、https://blog.aigrowthagent.co/best-wordpress-ai-tools-2026/
18. Vulnerability Summary for the Week of January 20, 2025 | CISA, 4月 12, 2026にアクセス、https://www.cisa.gov/news-events/bulletins/sb25-026
19. AI Puffer – Your AI engine for WordPress (formerly AI Power) - Wordfence, 4月 12, 2026にアクセス、https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/gpt3-ai-content-generator
20. Security Bulletin 06 November 2024, 4月 12, 2026にアクセス、https://isomer-user-content.by.gov.sg/36/7b64385f-f526-413b-9e03-8fd64d816017/06-November-2024.pdf
21. Preventing Zero-Click AI Threats: Insights from EchoLeak | Trend Micro (UK), 4月 12, 2026にアクセス、https://www.trendmicro.com/en_gb/research/25/g/preventing-zero-click-ai-threats-insights-from-echoleak.html
22. OWASP Top 10 for Agentic Applications 2026: Security Guide - Giskard, 4月 12,

2026にアクセス、
https://www.giskard.ai/knowledge/owasp-top-10-for-agentic-application-2026
23. 4月 12, 2026にアクセス、
https://www.microsoft.com/en-us/security/blog/2026/03/30/addressing-the-owasp-top-10-risks-in-agentic-ai-with-microsoft-copilot-studio/#:~:text=Cascading%20failures%20(ASI08)%3A%20A,approvals%20or%20extract%20sensitive%20information.
24. Addressing the OWASP Top 10 Risks in Agentic AI with Microsoft Copilot Studio, 4月 12, 2026にアクセス、
https://www.microsoft.com/en-us/security/blog/2026/03/30/addressing-the-owasp-top-10-risks-in-agentic-ai-with-microsoft-copilot-studio/
25. Cascading Failures in Agentic AI: Complete OWASP ASI08 Security Guide 2026 |, 4月 12, 2026にアクセス、
https://adversa.ai/blog/cascading-failures-in-agentic-ai-complete-owasp-asi08-security-guide-2026/
26. What is WooCommerce's gross merchandise volume (GMV)? - Red Stag Fulfillment, 4月 12, 2026にアクセス、
https://redstagfulfillment.com/woocommerces-gross-merchandise-volume/
27. 50 WooCommerce statistics & trends (New 2026 data) - WiserReview, 4月 12, 2026にアクセス、https://wiserreview.com/blog/woocommerce-statistics/
28. WooCommerce vs Shopify in 2026: Definitive Comparison, 4月 12, 2026にアクセス、https://funnelish.com/blog/woocommerce-vs-shopify
29. PCI-DSS-v4_0_1.pdf, 4月 12, 2026にアクセス、
https://www.middlebury.edu/sites/default/files/2025-01/PCI-DSS-v4_0_1.pdf?fv=AKHVQBp6
30. PCI Data Security Standard: Key Requirements Guide - SentinelOne, 4月 12, 2026にアクセス、
https://www.sentinelone.com/cybersecurity-101/cybersecurity/pci-data-security-standard/
31. PCI-DSS compliance and WooCommerce: Documentation, 4月 12, 2026にアクセス、
https://woocommerce.com/document/pci-dss-compliance-and-woocommerce/
32. Project Glasswing - Anthropic, 4月 12, 2026にアクセス、
https://www.anthropic.com/project/glasswing
33. Claude Mythos overhyped? Gary Marcus says ‘they are planting seeds in the hype garden’, but calls for restraint, 4月 12, 2026にアクセス、
https://www.financialexpress.com/life/technology-claude-mythos-overhyped-gary-marcus-says-they-are-planting-seeds-in-the-hype-garden-but-calls-for-restraint-4202646/
34. Amazon Bedrock now offers Claude Mythos Preview (Gated Research Preview), 4月 12, 2026にアクセス、
https://aws.amazon.com/about-aws/whats-new/2026/04/amazon-bedrock-claude-mythos/
35. Project Glasswing: Securing critical software for the AI era - Anthropic, 4月 12, 2026にアクセス、https://www.anthropic.com/glasswing

36. Anthropic Unveils ‘Project Glasswing’ as Claude Mythos Targets Software Vulnerabilities, 4月 12, 2026にアクセス、 https://www.hpcwire.com/aiwire/2026/04/09/anthropic-unveils-project-glasswing-as-claude-mythos-targets-software-vulnerabilities/
37. What Is a Data Breach? 11 Ways to Prevent One | CrowdStrike, 4月 12, 2026にアクセス、 https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/data-breach/
38. 250+ Cybercrime Statistics for 2026 - Bright Defense, 4月 12, 2026にアクセス、 https://www.brightdefense.com/resources/cybercrime-statistics/
39. Claude Mythos Preview identifies 27-year-old bug, finds 'thousands ..., 4月 12, 2026にアクセス、 https://www.scworld.com/news/anthropic-claude-mythos-preview-finds-thousands-of-vulnerabilities-in-weeks
40. Claude Code overview - Claude Code Docs, 4月 12, 2026にアクセス、 https://code.claude.com/docs/en/overview
41. Enabling Claude Code to work more autonomously - Anthropic, 4月 12, 2026にアクセス、 https://www.anthropic.com/news/enabling-claude-code-to-work-more-autonomously
42. Anthropic's Claude Code hits 81.6K GitHub stars: what developers should know, 4月 12, 2026にアクセス、 https://www.augmentcode.com/learn/anthropic-claude-code-github-stars