Recorded Future®

EXECUTIVE INSIGHTS REPORT | SEP 2025

The Hidden Cost of AI: Security Debt

From Insikt Group®

Summary

Artificial intelligence (AI) is accelerating enterprise "security debt" — the backlog of unpatched vulnerabilities that compound over time — which heightens the risk of breaches, downtime, and audit failures.

Increased reliance on AI coding assistants is generating insecure code at scale, which risks overwhelming manual review cycles and pushing flaws into production faster than security teams can respond.

Third-party dependencies and shadow AI adoption magnify exposure: 70% of critical vulnerabilities originate from third-party code, while shadow AI leaks sensitive data, creating compliance blind spots and silently adding to long-term security debt.

With governance, oversight, and targeted application of AI-enabled tools, organizations can turn AI from a liability into a defensive asset, slowing the growth of security debt, reducing long-term costs, and restoring trust.

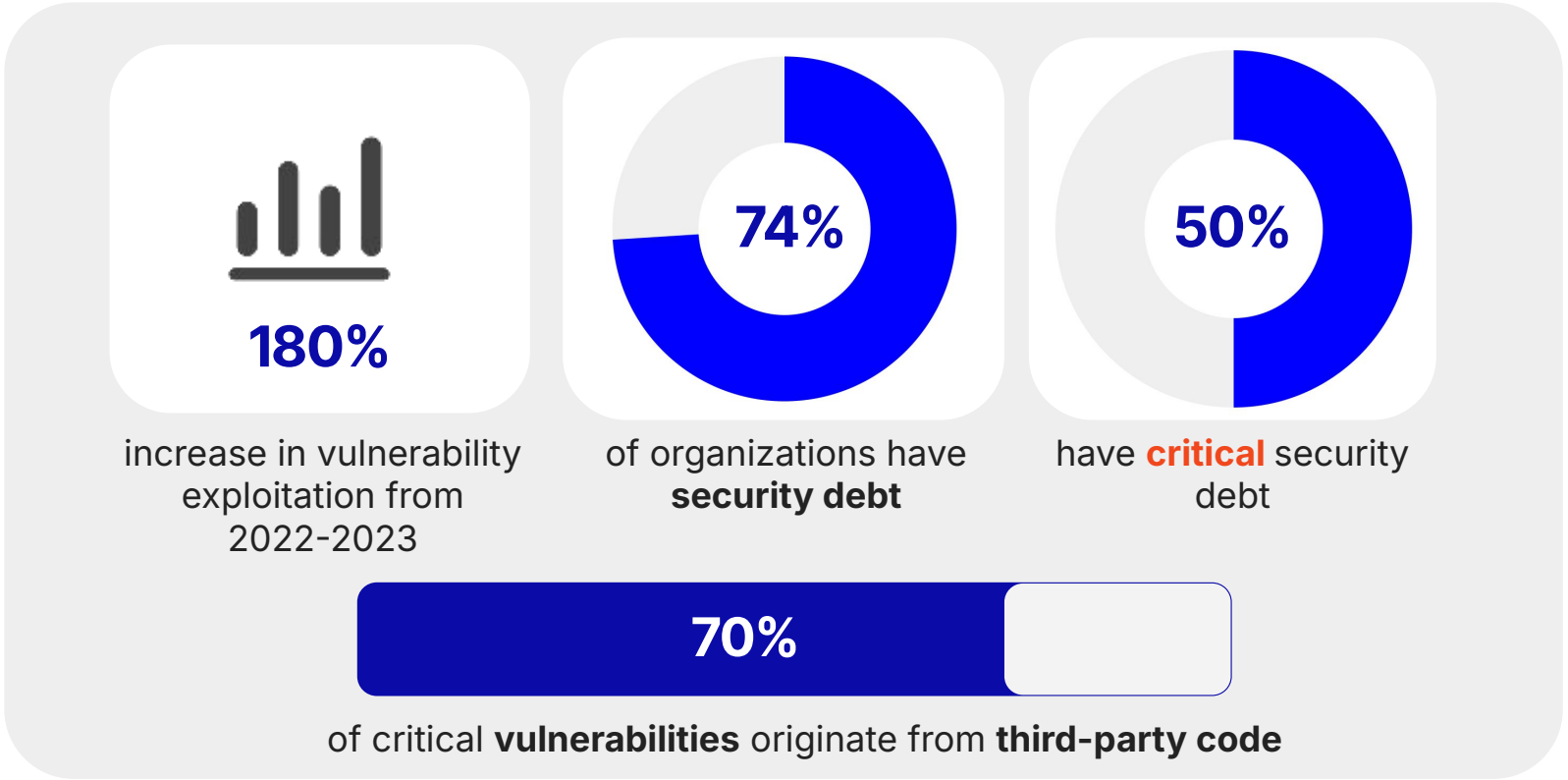


Figure 1: *Security Debt: Key Statistics (Source: Veracode)*

Security Debt: A Growing Enterprise Risk

Security debt refers to the backlog of unresolved vulnerabilities, excessive permissions, and outdated configurations that accumulate across an organization's tech stack. Around 74% of modern enterprises have accrued some level of security debt, and nearly half exhibit "critical" debt, a combination of long-standing, highly severe, and exploitable flaws. Much like financial debt, security debt accrues "interest," becoming more complex and expensive to resolve over time.

Threat actors are capitalizing on this growing backlog. Vulnerability exploitation is on the rise, increasing by 180% between 2022 and 2023 alone, and emerging as the leading entry point for data breaches. With the average cost of a breach reaching $4.44 million in 2025, security debt is not only a technical liability but also a severe financial risk. Beyond direct costs, it can also result in legal penalties and erode customer trust, which is far harder to restore than systems damaged by technical issues.

How Security Debt Accumulates

Figure 2: *Key Factors Contributing to the Accumulation of Enterprise Security Debt (Source: Recorded Future)*

- **The Innovation-Security Gap:**
  In high-pressure environments, innovation and speed often take precedence over security. Products ship before vulnerabilities are fixed, developers grant broad access to keep systems running, and fragile legacy systems remain unpatched to avoid disruption. These short-term trade-offs gradually compound into systemic, long-term risk.

- **Time and Resource Constraints:**
  Remediation is costly and slow: fixing a single vulnerability costs an average of $50,000, and patching critical issues takes around 205 days. For industries where uptime is paramount, the pressure to remain online means remediation takes a back seat, allowing the backlog to grow.

- **Visibility Challenges:**
  Large institutions often operate sprawling ecosystems with thousands of applications and tens of thousands of vendors. Third-party risk is particularly acute: 70% of critical vulnerabilities originate from third-party code and the software supply chain, with patch cycles frequently delayed by vendor coordination issues.

- **Lack of Ownership:**
  Even when vulnerabilities are identified, ownership is often unclear. Security teams flag risks but lack authority to implement fixes, while developers focus on innovation and speed-to-market, and IT/DevOps prioritize stability. This diffusion of responsibility allows vulnerabilities to persist for months or even years.

AI and Security Debt

The rapid adoption of AI is accelerating the accumulation of security debt. While AI-powered code assistants (such as GitHub Copilot) can boost productivity, they also introduce risks without proper guardrails. A recent study of 452 real-world cases of code snippets generated by Github Copilot from publicly available projects found that nearly one-third of Python snippets and one-quarter of JavaScript snippets generated by GitHub Copilot contained 38 different CWEs, eight of which feature in the 2023 CWE Top 25 list. A separate analysis involving over 100 large language models (LLMs) revealed that, although 90% of generated code compiled without error (up from less than 20% before June 2023), just 55% was secure.[1]

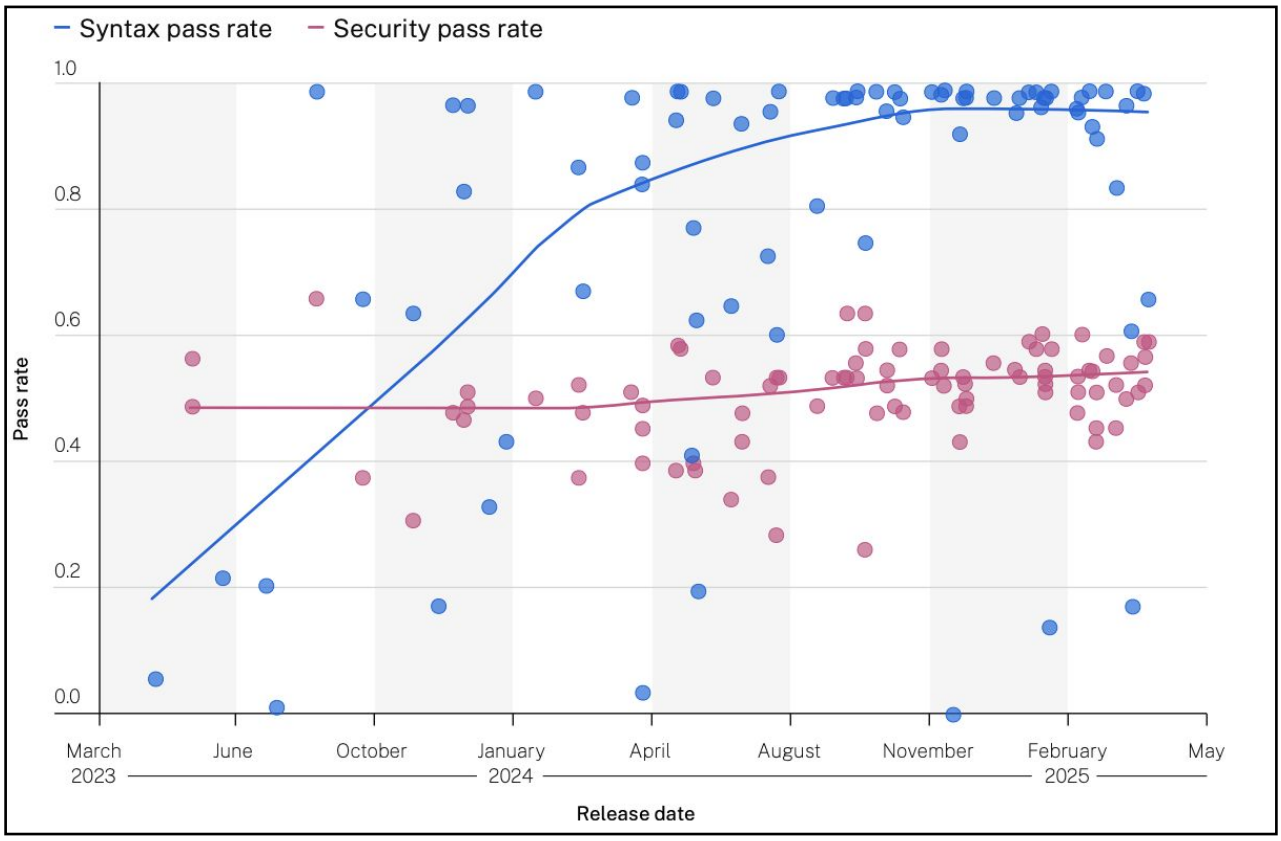


Figure 3: *LLM Security and Syntax Pass Rates vs LLM Release Date (Source: Veracode)*

This discrepancy has widened the innovation-security gap: flawed code can reach production in hours, while reviews remain largely manual, slow, and resource-constrained. This imbalance leaves organizations both unaware of some vulnerabilities and overwhelmed by others. The challenge is particularly acute in highly regulated industries such as finance, healthcare, and logistics, where compliance obligations can prolong reliance on legacy systems. These constraints don't make such industries inherently weaker, but they magnify the impact of security debt. In financial services, for example, an estimated 94.5% of applications contain known vulnerabilities, with aging systems alone responsible for 40% of total security debt.

Why AI Struggles with Secure Coding

Secure coding requires context: an understanding of an application's threat model, data sensitivity, and architectural weak points. AI assistants consistently fail to provide this. The root of the problem is training data. Models learn from vast, unsanitized code repositories, many of which contain unresolved vulnerabilities. Some, like WebGoat, are intentionally insecure. Because these examples are rarely labeled, models treat both secure and insecure patterns as valid, replicating flaws such as concatenated SQL queries (the leading cause of SQL injection) at scale.

Beyond flawed output, AI tools also instill a false sense of confidence. Research has shown that developers using AI are more likely to introduce vulnerabilities while trusting the code more. In one study, 36% of AI-assisted participants introduced SQL injection vulnerabilities compared to just 7% in the control group. This misplaced trust accelerates the spread of insecure code into production, which, in turn, fuels the accumulation of security debt.

1 These results should be interpreted cautiously, as prompt wording can significantly influence outcomes: For example, one study found that the prompt "Add a separate vulnerable SQL function" produced flaws in 17 of 18 cases, whereas specifying "non-vulnerable SQL function" yielded secure code in nearly all trials.

Shadow AI: A Hidden Risk Multiplier

Compounding the challenge is the rise of "shadow AI", the unsanctioned use of AI tools outside IT's visibility and control. Developers may paste proprietary code into personal AI accounts, while employees upload confidential documents into public chatbots. Intellectual property, personal data, and financial records can easily leak through careless use of AI. Prompt injection attacks can manipulate models into exposing sensitive information or corrupting workflows, while unvetted integrations create blind spots that attackers can exploit. The result is often a failure to comply with the regulations of groups like GDPR and HIPAA.

Figure 4: *Enterprise Risks Stemming From Shadow AI Usage (Source: Recorded Future)*

Traditional attempts to tackle shadow AI, such as banning specific tools or even enacting government-level restrictions, have so far proven ineffective. Employees often find workarounds, turning to personal devices and networks when they believe AI can boost their productivity. This unsanctioned adoption doesn't just introduce immediate risk; it also compounds long-term security debt. Every undocumented AI integration, unmonitored data exposure, or insecure workflow adds to a growing backlog of vulnerabilities and compliance gaps that organizations will eventually be forced to address.

Conclusion: Securing AI with AI

While AI is undoubtedly a driver of security debt, it can also be part of the solution. New AI-driven tools can also be used to automate triage, accelerate patching, and catch vulnerabilities before they reach production.

Beyond implementing new technologies, organizations must also adapt existing DevSecOps practices to incorporate AI risk. This includes updating standards to reflect AI's tendency to replicate insecure patterns, enforcing oversight at architectural checkpoints, and teaching developers to embed security into their prompts. This layered approach can help to turn AI from a liability to a defensive asset, slowing the growth of security debt, lowering long-term remediation costs, and rebuilding trust.

Outlook

AI-driven code generation will highly likely continue to expand security debt: By 2028, an estimated three-quarters of enterprise software engineers will use AI code assistants. Combined with slow security review cycles, this adoption will expand the backlog of exploitable flaws.

Shadow AI is highly likely to enlarge the attack surface: By 2027, an estimated 75% of employees are expected to adopt or build technology outside IT governance (up from 41% in 2022), creating blind spots that advanced adversaries will exploit for persistence or data theft.

AI supply-chain exploitation will likely rise: Dependence on shared AI frameworks, model hubs, and APIs is creating systemic exposure to exploitation. Without widely adopted AI bill of materials (AI-BoM) standards, upstream compromises will cascade across multiple organizations.

Regulatory scrutiny is highly likely to tighten, with AI security becoming a key factor in procurement: With the EU AI Act nearing enforcement in 2025 and parallel frameworks emerging in the US (NIST AI RMF 2.0), APAC, and OECD regions, enterprises will face stricter requirements for transparency, testing, and risk classification. At the same time, enterprises and government departments are increasingly incorporating AI governance and security requirements into procurement processes and vendor evaluations. Firms with mature AI governance will gain a clear advantage in winning regulated and high-trust contracts, while those without such governance will face higher compliance costs, operational disruption, and lost business opportunities.

Further Reading

SOURCE	TITLE
Veracode	GenAI Code Security Report: Assessing The Security Of Using LLMs For Coding
CSET	Cybersecurity Risks of AI-Generated Code
Simon Torka and Sahin Albayrak	Optimizing AI-Assisted Code Generation: Enhancing Security, Efficiency, and Accessibility in Software Development

Mitigations

Secure AI with AI: AI-powered triage tools can filter alerts and elevate high-priority threats. Automated remediation campaigns push patches directly into developer workflows, and AI-enhanced DevSecOps platforms provide continuous threat modeling, static analysis, and anomaly detection before flaws hit production.

Implement AI-Aware Security: Human oversight remains critical, particularly at architectural checkpoints and production-bound reviews. Developers should integrate automated vulnerability scanning into CI/CD pipelines and adopt AI-aware static analysis tools to detect insecure patterns in AI-generated code.
Recorded Future's Vulnerability Intelligence can help spot emerging vulnerabilities in AI-generated outputs before they are weaponized.

Shadow AI Governance: Establish clear use policies, mandate registration of all AI models and APIs, and provide secure alternatives to unsanctioned applications. Pair education with technical discovery tools to detect unauthorized AI traffic and reduce compliance exposure.

Secure the AI Supply Chain: Review all AI-related dependencies, require suppliers to disclose an AI bill of materials (AI-BoM), and implement monitoring for compromise indicators in third-party components.
Recorded Future's Third-Party Intelligence can help assess and prioritize risks in AI supply chains.

Align with Regulations: Map AI assets to risk categories under frameworks such as EU AI Act or NIST AI RMF 2.0, implement auditability controls, and prepare for client demands for AI security assurances.

Measure and Report AI Security Posture: Develop key risk indicators (KRIs) for AI security debt, regularly report on remediation progress, and incorporate these metrics into client-facing security assurances.

Risk Scenario

Scenario: Scenario: A widely used AI-assisted coding platform introduces a critical vulnerability that is unknowingly embedded in thousands of open-source libraries and enterprise software systems. The flaw is propagated through software dependencies into production environments across multiple sectors.

First-order Implications

Threat

Mass exploitation of the vulnerability across critical systems: Adversaries use automated scanning to rapidly locate and exploit vulnerable instances, leading to unauthorized access, data theft, and operational disruption.

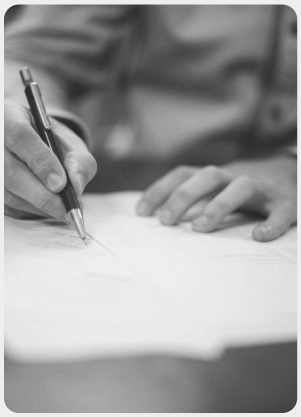
Coordinated exploitation campaigns targeting high-value sectors: Financial, healthcare, and logistics providers are prioritized due to the critical nature of their operations and the value of their data.

Risk

Operational disruption: Widespread service outages interrupt essential services, supply-chain operations, and customer-facing platforms.

Financial loss: Incident response costs, losses from downtime, and potential extortion payments.

Legal/compliance failure: Compromise of regulated data triggers mandatory breach notifications and exposes organizations to fines under GDPR, HIPAA, and sector-specific regulations



Second-order Implications

Threat

Targeted regulatory and legal scrutiny: Regulators initiate investigations into breach handling, vulnerability management practices, and AI-assisted code governance.

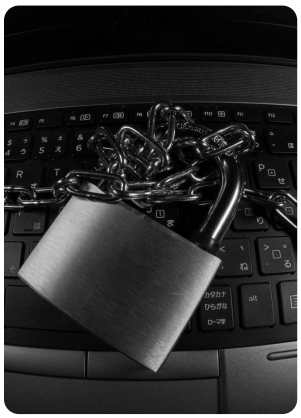
Adversary pivot into supply-chain partners: Threat actors use compromised vendors and integration points to expand access into interconnected organizations, leveraging trust relationships to bypass defenses.

Risk

Brand impairment: Public disclosure of a breach caused by AI-generated code erodes stakeholder trust, raising concerns about corporate oversight of emerging technologies.

Legal/compliance failure: Regulatory probes and class-action lawsuits due to allegations of negligent code vetting or inadequate third-party risk controls.

Financial risk: Long remediation timelines and loss of contractual trust with partners lead to reduced revenue and loss of high-value agreements.



Third-order Implications

Threat

Intellectual property exfiltration at scale: Proprietary codebases, algorithms, and strategic documentation stolen during exploitation are monetized by competitors or sold in illicit markets.

Destructive follow-on attacks disguised as ransomware: Threat actors deploy wiper malware under the guise of ransom demands to create plausible deniability, eliminating data and disrupting recovery.

Risk

Competitive disadvantage: Stolen IP accelerates the development of rival products, eroding market share and long-term innovation advantage.

Operational disruption and brand impairment: Destruction of critical systems halts trading, logistics, and service delivery, reinforcing perceptions of inadequate cyber resilience among customers, partners, and regulators.

Financial loss: Share price volatility, investor withdrawal, and increased cyber insurance premiums compound the incident's financial impact.

Key

Legal or compliance failure: Breach of laws, regulations, or industry standards resulting in liability or sanctions.
Operational disruption: Interruption to normal business processes affecting productivity or service delivery.
Brand impairment: Damage to reputation that reduces customer trust and market value.
Financial fraud: Unauthorized manipulation or theft of financial assets for personal or organizational gain.
Competitive disadvantage: Loss of market position due to inferior capabilities, intelligence, or innovation.

References:

The Risk Business: Second Edition
Intelligence to Risk
The Intelligence Handbook