WHITEPAPER

A CISO'S GUIDE TO SAFELY UNLEASHING THE POWER OF GENAI

CUSTOM CONTENT FOR SNYK BY STUDIOID

Developers have their eye on generative AI. As developers harness the power of AI-generated code, this reliance poses a critical challenge — security tools must evolve at a parallel pace.

GenAI can create more code faster, which boosts developers' workflows and productivity, but unfortunately, security awareness and practice surrounding genAI haven't mirrored the rapidly advancing technology. **Gartner predicts that 80% of code will have used genAI APIs or deployed genAI-enabled apps by 2026**, so companies must address the growing and dynamic relationship between genAI-driven code and the imperative for security tools to adapt.

Despite the numerous advantages of genAI development assistant tools, real challenges abound. GenAI, especially large language models (LLMs), requires massive amounts of training data to learn and generate responses, so extensive and diverse datasets become essential, and much of this data comes from notoriously risky open-source repositories. Making matters worse, many developers overly trust AI-generated code — blindly believing in the inherent security of AI-generated content.

Also, many modern genAI security tools and practices aren't effective in ensuring the safety of genAI-created content. Security checks, if done, are often conducted by the same software that wrote the code — clearly a conflict of interest. Today's developers need the **fastest and most accurate** scanning tools to handle an increased influx of code, and sadly, many current SAST tools can't keep pace.

This playbook will explore the pros and cons of genAI, make recommendations for building, scaling and governing a secure genAI program, and explain what to look for in a genAI security tool to help your team drive innovation and adoption of AI through securing AI-generated code so that teams can use AI with peace of mind.

A CISO's Perspective on GenAI: The Good Behind the Optimism

"Whenever we talk to customers, we tell them every time you talk about code generation or AI tools, you need to treat them like another developer on your team," said Edgar Kussberg, staff product manager at Snyk.

And not just any developer — but a super developer capable of a pace and production volume impossible for humans to match. Today's savvy coding professionals are aware of the advantages of the genAI assistant tool — 92% of U.S.-based developers already use AI coding assistants in and out of work, according to a GitHub survey.

However, despite the immense potential, keeping a close eye on genAI assistants, like any inexperienced developer lacking field time and an understanding of context, is critical. Be excited about the many advantages your team's new developer brings — but be prepared to diligently monitor genAI output while running rigorous security checks on all genAI-generated code.

> "Whenever we talk to customers, we tell them every time you talk about code generation or AI tools, you need to treat them like another developer on your team."
>
> Edgar Kussberg,
> Staff Product Manager at
> Snyk

What exactly is the technology behind this latest development trend?

GenAI coding assistant tools such as GitHub Copilot, Code Whisperer and IBM WatsonX Code Assistant use natural language processing and machine learning algorithms to break down and understand code construction, then instantaneously offer code-writing suggestions based on provided analysis and goals. GenAI's ability to create code rests on extensive training with volumes of data fed to AI models while closely following developer guidelines for desired outcomes.

GenAI coding assistants undeniably have the power to revolutionize the development industry by generating more code faster while helping developers expedite code testing and debugging. "**Generative AI has introduced unprecedented acceleration to the software development process, and tasks that used to take weeks or months now only take a fraction of the time. This speed enhances productivity and enables faster adaptation to changing market demands and technological trends. Consequently, GenAI holds immense potential to expedite innovation and bring products to market swiftly,**" explained Tom Terronez, CEO at Medix Dental, who has a significant background in data security and software development.

Just how much of a difference can genAI assistant tools make? Developers who used Copilot completed a task **55% faster than those who didn't, and developers generally feel the technology makes their job easier, with 70% saying that AI coding tools will offer them an advantage at work** and cite better code quality, completion time and resolving incidents as some of the top anticipated benefits.

GenAI's potential excites many because it can take over mundane manual components, freeing humans to work in more complex, value-added, creative-development roles. GenAI assistant tools can also democratize development, opening the industry to those with minimal development experience, allowing them to use genAI assistant's coding suggestions to do more with code than they otherwise would have been able to. However, democratization of development can be a double edged sword.

GenAI's Immense Potential Doesn't Come Without Challenges

GenAI assistant tools do have their challenges — as algorithms are only as strong as their data. Today, many genAI models train on code from open-source repositories, which is notoriously risky and can present significant security issues for developers. As well, the data can be biased and deliver biased results. "Open-source code can be full of vulnerabilities, and if you train your AI models on vulnerable code, they will spit out vulnerable code," Kussberg said.

Unfortunately, vulnerability-laden open-source code is disturbingly common. In various diverse programming scenarios, GPT-3.5-generated C programs manifested vulnerabilities in 51.2% of cases. In October 2023, researchers at Cornell University found that **35.8% of Copilot-generated code snippets contained instances of common weaknesses (CWEs)** across multiple languages, and, more worryingly, about 26% of the CWEs identified were among 2022's top 25 CWEs. Possibly even more concerning — **84% of codebases contained at least one known open-source vulnerability, and 48% contained high risk open-source vulnerabilities,** according to the Open Source Security and Risk Analysis Report.

Code vulnerabilities open the door for various threat actors and cyberthreats, many of which can inflict serious financial or reputational harm to your organization. Cybercriminals can use vulnerabilities to inject harmful code into AI-training data, then manipulate and trick the AI tools to generate compromised code for stealing data or further infiltrating corporate networks. Poorly optimized AI models can also tax computational systems, opening them to exhaustion/overload Denial of Service attacks, which can devastate any business.

> "Open-source code can be full of vulnerabilities, and if you train your AI models on vulnerable code, they will spit out vulnerable code."
>
> - Edgar Kussberg, Staff Product Manager at Snyk

The creative power of genAI does democratize development — but unfortunately, not just for the good actors. Here's a chilling thought for cybersecurity pros today — AI code generation is "useful for those criminal actors with little to no knowledge of coding and development," according to the European Union Agency for Law Enforcement Cooperation (EUROPOL) report ChatGPT: The impact of Large Language Models on Law Enforcement. Many worry that the creative power of genAI in the wrong hands could increase the frequency and sophistication of cybersecurity attacks moving forward.

Worse yet, security surrounding genAI assistant tools today isn't close to what it needs to be. Lagging security starts with many developers maintaining an unfounded and unrealistic trust in genAI. Many genAI assistants look polished and arrive as easy to-deploy solutions — **plus everyone uses them — so they must be trustworthy, right?**

A vast majority agree, based on a recent Snyk survey in which 75% of respondents said they felt AI code was more secure than human code. Many developers put too much faith into AI tools, feeling they are nearly infallible in providing secure code or when used for security practices. Some feel this unwarranted trust in AI may boil down to software development's culture. Developers have a million things to do with pressing production deadlines, so the high value they place on the speed and efficiency of genAI tools can lead to an inflated belief and unjustified trust in AI generated output.

75%

of respondents said they **felt AI code was more secure than** human code

Developer trust can lead to security issues in light of recent statistics indicating that genAI code is often less secure than code written by human developers. **Yes, you read that correctly — the machines, with all their speed, occasionally deliver less secure code than slower humans.** A study by Cornell University researchers in December 2022 found that participants with access to an AI assistant "wrote significantly less secure code" than those without access, but ironically, they were more likely to believe they wrote secure code. Also, 91.6% of respondents in Snyk's report said that AI coding tools generated insecure code at least some of the time, while 56.4% said they sometimes or frequently encountered security issues in AI code suggestions.

87%

of respondents said they were concerned about the security implications of using AI coding assistant tools

Here's an ironic and troubling trend — **general concern for genAI security remains high, yet a lack of consistent, disciplined security practices exists.** Eighty-seven percent of respondents said they were concerned about the security implications of using AI coding assistant tools, and 34% of organizations used AI app security tools to mitigate the risks of genAI. However, few genuinely prioritized security, as 80% of developers admitted to bypassing security policies in their organization, and less than 10% of respondents said they used automation in most security scanning checks.

It's clear security trends must change (and fast!) as AI coding tools are already a significant part of a majority's regular software development, but few users demonstrate acceptable levels of safety. It's time for organizations to prioritize proper security practices and use effective tools to counter the risks of genAI in development workflows.

How To Build, Scale and Govern Your Secure GenAI Program

Building, scaling and governing a secure genAI program sounds like quite the undertaking — especially considering genAI is a new technology with few established guidelines. **So where can your team begin?** Here are our recommended steps for your organization to build, scale and govern a secure genAI program:

→ Establish leadership

Your organization's first step toward a secure genAI program is establishing a base of committed security leadership. Your team should begin by creating an internal task force, ideally with a voice from all departments, dedicated to developing an organized, strategic set of guidelines addressing all aspects of security surrounding genAI tools.

→ Develop a framework of policies and procedures

Effective genAI governance requires policies and procedures to guide the safe use of genAI coding tools. Your genAI task force must shoulder the responsibility of developing this framework and emphasizing the critical importance of these genAI guidelines to every member of your organization.

Practical procedures should include compiling a comprehensive overview of where genAI tools exist in your team's development workflow. This step is critical because AI code can easily mix with human code, and it becomes hard to monitor with the necessary higher-level, stricter checks if your team isn't aware of the former's location.

Next, map out approved genAI tools and how your team will use them, specifying which team member is responsible for which tool with their contact details. Finally, establish specific safeguarding policies and practices surrounding all genAI tools. A good example would be requiring layers of checks, such as having security guardrails and human validation of AI-generated code by developers using genAI assistant tools.

→ Automate, automate, automate

Prepare to automate all genAI security processes wherever and whenever possible. Automation brings critical real-time speed to testing, validating and verifying AI code, helping developers maximize production at scale.

→ Select the best AI-empowered security tools

Choose your security tools wisely, emphasizing:

Speed. Today's security solutions must deliver real-time actionable results for fast threat response and mitigation without slowing or disrupting the development workflow. Snyk Security tools should review code directly in the IDE with AI-empowered speed (Snyk is 2.4x faster on average than other solutions) and be capable of running automated real-time security audits and performing one-click instant vulnerability remediation. Speedier and earlier discovery and remediation means less cost and time wasted, increasing the flow of code to production.

Design by security experts. It matters whom you trust to design your AI security tools, so be sure to choose tools designed by focused security experts over industry generalists. Trustworthy experts use highly specific data and code pulled only from proven, quality, open source repositories for AI training. Expert-crafted tools also hold higher code standards rather than the standards of the mean across the board, like most AI coding and security providers. A little background here — because of the breadth and volume of data needed to train general-purpose LLMs, there can be a wide variance in the level of code quality regarding security, which necessitates a lower mean standard compared with an LLM set up and trained with security in mind.

Tools designed by security experts incorporate extensive human-based input to curate training data, and use AI to help refine the resulting rules, among other things. These expert practices translate to higher accuracy rates and more relevant results while cutting out the noise of countless false positives and negatives. For an example of this success, look at Snyk's automated software vulnerability detection tool — **Snyk's SAST solution received a 72% accuracy score** on the OWASP benchmark.

Specialized methodologies. The most effective AI security tools available today employ the latest methodologies to provide every security advantage for users. Examples include Snyk's use of hybrid AI, which combines the rule-based classical AI with the statistical learning methods of modern AI to garner the best features of each, with a human-in-the-loop process.

Human-in-the-loop involves human experts in creating training datasets, rule generation, and code review and fixing. Human experts oversee AI, constantly checking AI inputs, tweaking the AI's frameworks and validating AI outputs. This powerful combination of human thought, expertise and discretion with AI power helps deliver higher accuracy rates and significantly fewer false positives and negatives.

Leading security tools also help users work with a complete understanding of app context with each scan, finding and fixing issues as developers work, but more important, also automatically rescanning an entire app with a potential vulnerability fix in place and ensuring that this does not create new security issues, before proposing suggested fixes to users. Fully understanding app context with each scan is critical because every introduction of new code changes the risks within the codebase, and understanding context helps to catch evolving risks and different vulnerabilities with every new introduction of code.

72%

Accurancy Score recieved on the OWASP Benchmark for Snyk's SAST Solution

Other specialized methodologies include issue triaging, which uses vulnerability risk-scoring to help developer teams prioritize the most impactful fixes, and native reporting function, which gives unfettered, smooth access to users' vast code security data, employing extensive filtering options for deep insight into security data and empowering security leads and CISOs to create more impactful security tactics and strategy.

Easy integration and mitigation. The best security tools work where you need them, alerting users to danger but also providing them with convenient solutions. Help your team by choosing an AI security tool that runs in dimensional IDEs, is fueled by automation, tests code as soon as it's generated, alerts users on vulnerabilities in real time, and readily offers and initiates mitigation pathways to fix issues that arise.

Maximum control coverage. Remember to select an AI security tool capable of overseeing and monitoring security posture across your entire network, pinpointing area owners for accountability, and sharing this oversight with you through a useful interface. Leading AI security solutions arrive with the ability to scan your whole estate and confidently determine scanned code and unscanned code while prioritizing all risks, vulnerabilities and mitigation efforts.

Independent analysis. Unfortunately, today, many teams trust the same tools that write code to perform the security checks on that same code. And where they have separate code security tools, many of these security tools belong to the same company that produced the AI coding tools these teams use. Ensure your team chooses a fully independent security tool to check all genAI-created code for a separate, unbiased view of your security posture. This independence is essential for governance, especially for companies in highly regulated finance, banking or health care industries.

→ Conduct regular training

Your leadership task force must prepare, practice and maintain regular training for all development team personnel using genAI coding assistant technology and genAI security tools. For example, all developers should understand the importance of human validation of genAI code, why it is crucial and how they would need to validate it. Leadership should also be responsible for staying in contact with vendors to remain current on all genAI tool updates and security patches.

→ Have an emergency plan

The adage that failing to plan is essentially planning to fail couldn't be more accurate than in the emerging landscape of genAI. Set aside ample time and resources for your team to prepare a thorough response plan for any possible security incidents related to genAI tools, and regularly practice organization wide incident-response drills.

In Conclusion

GenAI is poised to significantly affect the future of business across industries, with software development benefiting from speed and productivity, as AI assistants help developers to write code. GenAI is still very new and high in the hype cycle, but it is poised to change every organization's approach to software development, with adoption becoming mandatory to remain competitive in any digital market.

For all the advantages that AI coding assistants can undoubtedly provide, **we must remember that they are tools meant to augment, not replace, developers.** GenAI assistant tools aren't infallible, and developers must be careful not to overly trust their coding suggestions. GenAI can create functional but not necessarily secure code, and these tools come with many potential security challenges that users cannot overlook or minimize.

"GenAI, like any powerful technology, possesses a dual nature. On one hand, it offers immense potential and benefits, revolutionizing the software development process with efficiency and innovation. However, this very power can be harnessed for malicious purposes if not adequately secured," Terronez said.

Organizations must prioritize security across development workflows for maximum business benefit while minimizing the potential risks associated with genAI assistant tools. Terronez explained, "Today, organizations must prioritize securing their genAI-powered systems. Comprehensive security measures are paramount to harnessing genAI's upside while mitigating the risks, and a combination of AI-driven automated tools teamed with human expertise can create a formidable defense against security risks in genAI-driven software development."

Snyk Code for SAST provides an automated, assistant fleet that doesn't disrupt your team's development AI-fueled, real-time security intelligence workflow.

For more information or to get started with Snyk for free, visit https://snyk.io.

snyk

studio / ID

BY INDUSTRY DIVE

studioID is Industry Dive’s global content studio offering brands an ROI rich tool kit: Deep industry expertise, first-party audience insights, an editorial approach to brand storytelling, and targeted distribution capabilities. Our trusted in-house content marketers help brands power insights-fueled content programs that nurture prospects and customers from discovery through to purchase, connecting brand to demand.

Learn more