Chapter 7
Secure Code Generation and Software Engineering With Generative AI

Hanh Le
http://orcid.org/0009-0007-8172-0809
University of the Cumberlands, USA

ABSTRACT

The integration of generative AI into software engineering is transforming how developers write and secure code. Tools like GitHub Copilot and ChatGPT enhance productivity but also introduce risks to code quality, security, and accountability. This study explores how generative AI intersects with secure software engineering by examining its role within the Secure Software Development Life Cycle (SSDLC). It investigates how AI-generated code can propagate vulnerabilities and how prompt engineering may mitigate these risks. Emphasis is placed on the importance of explainability, code auditing, and human oversight in AI-augmented development environments. Through case studies in DevSecOps (Prates & Pereira, 2025), the paper illustrates both the promise and the pitfalls of AI-assisted coding. Findings underscore the need to evolve traditional practices to ensure safe, transparent, and ethical AI integration, offering actionable strategies for aligning intelligent automation with secure development standards.

INTRODUCTION

Generative AI has precipitated a fundamental transformation in the field of software engineering. Large Language Models (LLMs), such as OpenAI's ChatGPT and GitHub Copilot, have become essential tools throughout the software development

DOI: 10.4018/979-8-3693-8397-1.ch007

process. These models possess the capability to generate functional, context-aware source code, while also aiding developers in bug detection, offering code refactoring suggestions, and promoting adherence to security protocols. Consequently, tasks that were once the exclusive domain of seasoned software engineers, including the development of optimized algorithms, the identification of vulnerabilities, and the enforcement of secure coding practices, are now susceptible to partial or complete automation through the application of generative AI. As a result, this change is redefining the traditional boundaries of software engineering expertise, establishing new models for secure, efficient, and AI-enhanced development processes. As organizations strive to incorporate these tools into their development processes to boost productivity, cost-effectiveness, and innovation, concerns regarding software security and AI accountability have become increasingly pressing.

Generative AI offers both advantages and disadvantages for software security. It could simplify secure coding and help developers find security flaws more quickly. Conversely, the possibility of inadvertently producing insecure code is present, potentially stemming from the integration of, or amplification of, vulnerabilities inherent in the training data. Given that AI models often utilize extensive public code repositories, which frequently exhibit insecure coding practices, a legitimate risk emerges that these models could unintentionally disseminate such harmful methodologies or even introduce vulnerabilities into operational systems.

This study investigates a crucial issue at the intersection of artificial intelligence and cybersecurity: how software development teams can use generative AI while maintaining the integrity, security, and ethical standards that are essential to the software engineering process. To this end, we analyze the integration of artificial intelligence within the Secure Software Development Life Cycle (SSDLC). Moreover, this study investigates the dissemination of vulnerabilities within code generated by artificial intelligence. Ultimately, we suggest methodologies, including prompt engineering and AI explainability, to aid developers in managing the security of AI-generated outputs. Consequently, this research advocates for a collaborative approach, merging human expertise with AI augmentation, to facilitate the creation of secure, dependable, and resilient software systems.

This research presents several significant advancements that extend beyond the scope of prior investigations. Firstly, it provides empirically validated qualitative insights into developer interactions with generative AI tools within genuine, secure coding contexts, thus revealing patterns of trust, cognitive reliance, and risk evaluation. Moreover, it conducts a systematic analysis of AI-generated security vulnerabilities, linking these findings to established security frameworks like OWASP (Open Web Application Security Project) (OWASP, 2025), consequently bridging the gap between AI-enhanced development and traditional secure coding practices. Finally, this study explicitly incorporates three complementary theoretical frameworks—the

A Fuzzy-Based Approach to Support Decision Making in Complex Military Environments

Timothy P. Hanratty, E. Allison Newcomb, Robert J. Hammell II, John T. Richardsonand Mark R. Mittrick (2016). *International Journal of Intelligent Information Technologies (pp. 1-30).*

www.irma-international.org/article/a-fuzzy-based-approach-to-support-decision-making-in-complex-military-environments/145775