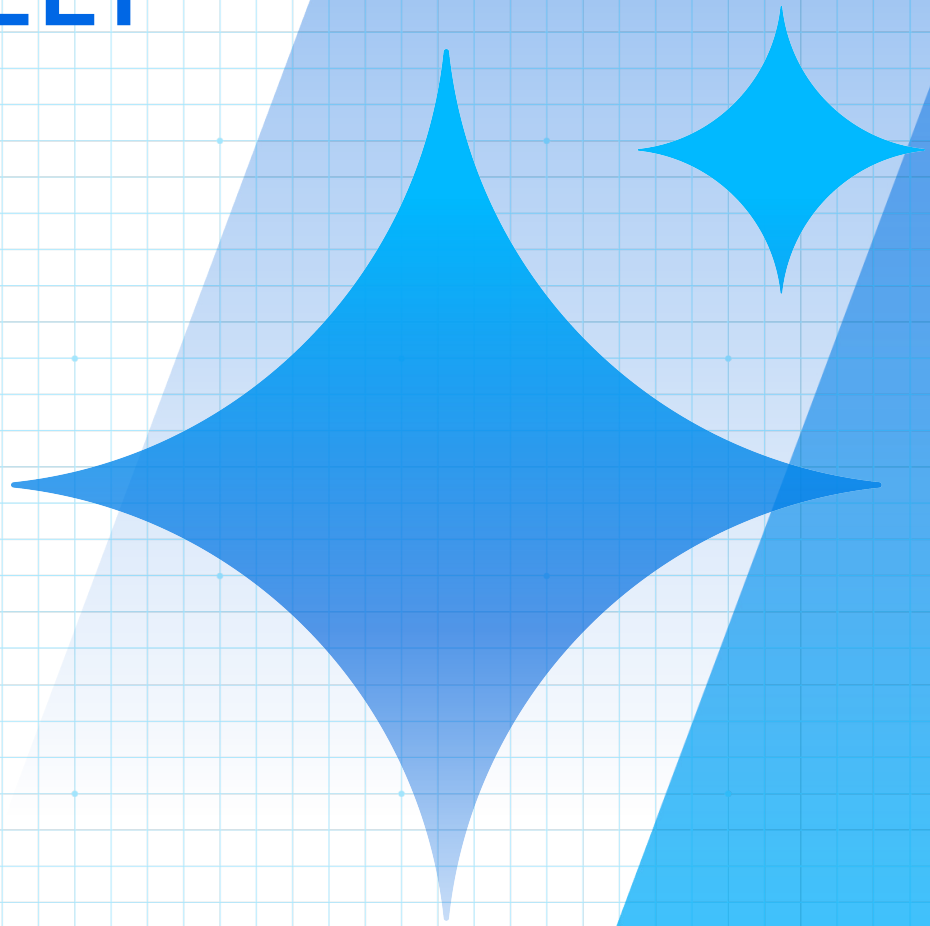
VERACODE

BUILDING SECURELY WITH AI:

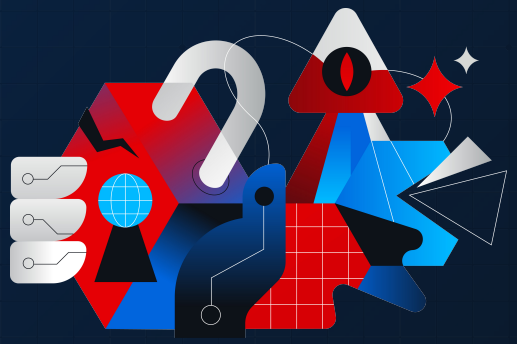
Revolutionizing Software Development in 2025



INTRODUCTION:

The Imperative of Secure AI Integration

Generative AI (GenAI) is fundamentally transforming the landscape of software development.



As organizations rapidly adopt AI to accelerate code generation and drive innovation, the security implications of AI-written code have become paramount. For Chief Information Security Officers (CISOs), security teams, and developers, understanding how to securely leverage GenAI is no longer optional — it is a strategic imperative.

This eBook provides a comprehensive examination of how generative AI is reshaping software development, the security challenges it introduces, and actionable strategies to mitigate associated risks. By integrating Veracode's extensive expertise with current industry trends, we aim to equip CISOs, security teams, developers, and other industry professionals (e.g., DevOps engineers, IT managers, and technology enthusiasts) with the knowledge required to build secure, AI-driven software. Our objective is to establish thought leadership while delivering practical guidance for navigating this evolving technological environment.

CHAPTER 1:

Overview of Generative AI in Software Development

Defining Generative AI in the Development Context

Generative AI refers to algorithms capable of creating new content, such as code, text, or images, based on patterns learned from extensive datasets. In software development, tools like GitHub Copilot and OpenAI's Codex are enhancing productivity by generating code snippets, automating repetitive tasks, and suggesting optimizations. A **2025 Gartner study projects that 70% of software development teams will integrate GenAI tools by year-end,** a significant increase from 40% in 2023.

OPPORTUNITIES PRESENTED BY GEN-AI

GenAI offers substantial benefits, including:

- ✔ **Accelerated Development:** AI can reduce coding time by up to 40%, as indicated by a 2024 McKinsey report, significantly improving development cycles.
- ✔ **Enhanced Accessibility:** It enables developers across various skill levels to produce complex code, broadening access to advanced development capabilities.
- ✔ **Fostered Innovation:** AI facilitates rapid prototyping and experimentation, accelerating the development of new solutions.

CHALLENGES INTRODUCED BY GEN-AI

Despite its advantages, GenAI presents notable challenges:

- ✘ **Code Quality Variances:** AI-generated code may contain inefficiencies or adhere to outdated practices.
- ✘ **Significant Security Risks:** A 2024 Veracode study revealed that 45% of AI-generated code contained vulnerabilities such as SQL injection or cross-site scripting (XSS).
- ✘ **Compliance Discrepancies:** AI-generated code may not consistently align with industry security standards, including the OWASP Top 10.
- ✘ **Bias Propagation:** AI models can perpetuate biases from their training data, potentially leading to flawed logic or security vulnerabilities.

KEY CONCEPTS FOR SECURE GEN-AI ADOPTION

Understanding the following concepts is crucial for secure GenAI integration:

- **Prompt Engineering:** The process of crafting precise inputs to guide AI outputs toward desired and secure results.
- **Static Application Security Testing (SAST):** Tools designed to scan and identify vulnerabilities within AI-generated code early in the development lifecycle.
- **Secure Development Lifecycle (SDLC):** The systematic integration of security measures at every stage of software development.

CHAPTER 2:

Security Risks Associated with AI-Generated Code

The Vulnerability Landscape of AI-Generated Code

AI-generated code can introduce subtle and often difficult-to-detect vulnerabilities. Industry analysis indicates that AI tools frequently suggest code with inadequate input validation, leading to common exploits such as XSS or buffer overflows. Unlike human developers, AI lacks contextual awareness, making it susceptible to generating insecure patterns, particularly when trained on outdated or unvetted open-source repositories.

Common Vulnerabilities in AI-Generated Code

Key vulnerabilities frequently observed include:

- **Injection Flaws:** AI may suggest unescaped SQL queries or user inputs, creating opportunities for injection attacks.
- **Authentication Issues:** Weak or absent access controls can be present in generated APIs.
- **Dependency Risks:** AI might recommend libraries with known vulnerabilities (e.g., the Log4j exploits from 2021 remain a pertinent concern).
- **Data Exposure:** Hardcoded secrets or sensitive data may inadvertently appear in generated code outputs.

The Critical Importance of Addressing These Risks

A single vulnerability can result in breaches incurring millions of dollars in costs and severe reputational damage. The 2023 MOVEit breach, caused by an SQL injection flaw, affected over 2,000 organizations globally, underscoring the severe implications. For CISOs, ensuring AI-generated code adheres to established security policies is a paramount priority.

Mitigation Strategies for AI-Generated Code

Effective mitigation strategies include:

Rigorous Code Review

Human oversight is essential for identifying and correcting AI-introduced errors.

Automated Scanning

Utilizing tools like Veracode's SAST to proactively identify vulnerabilities.

Developer Training

Educating developers on secure coding practices specifically tailored for AI tool integration.

CHAPTER 3:

Practical Applications and Solutions for Secure GenAI Integration

Securely Integrating GenAI into the SDLC

To effectively leverage GenAI while maintaining security, organizations must embed security throughout the SDLC. A recommended framework includes:

1. **Define Security Requirements:** Establish clear security standards (e.g., OWASP ASVS) before implementing AI tools.
2. **Utilize Trusted AI Models:** Select tools and models trained on verified, secure datasets.
3. **Implement Continuous Scanning:** Integrate SAST and Dynamic Application Security Testing (DAST) into CI/CD pipelines.
4. **Validate Outputs Systematically:** Manually review AI-generated code for both functional correctness and security flaws.
5. **Monitor Dependencies Continuously:** Employ Software Composition Analysis (SCA) to track and manage third-party libraries.

Essential Tools and Frameworks

Key tools and frameworks for secure AI development include:

Static Application Security Testing (SAST): Key to identifying vulnerabilities in AI-generated code.

Standard Security Risk Checklist: A foundational guide for mitigating common security risks in AI outputs.

CI/CD Automation Tools: Facilitates the automation of security scans within continuous integration and deployment pipelines.

Dependency Analysis Tool: An open-source solution for identifying vulnerable software libraries.

CHAPTER 4:

Veracode's Role in Securing AI-Written Code

Veracode's Application Risk Management platform is specifically designed to address the complex security challenges posed by AI-generated code. Our platform offers comprehensive capabilities:

- **Static Analysis (SAST):** Scans code for vulnerabilities such as XSS, SQL injection, and insecure dependencies.
- **Dynamic Analysis (DAST):** Tests running applications to identify runtime issues and configuration flaws.
- **Software Composition Analysis (SCA):** Identifies and manages risks within third-party libraries and open-source components
- **Package Firewall:** Monitors and governs for proactive defense against malicious pacages.
- **Fix:** Accelerates secure coding with AI-driven remediation.
- **Policy Enforcement:** Ensures AI-generated code adheres to critical compliance standards (e.g., PCI DSS, HIPAA).

The Veracode Advantage

Choosing Veracode offers distinct advantages:

Proven Accuracy: A 98% true-positive rate in vulnerability detection (2024 Veracode metrics) minimizes false positives and maximizes efficiency.

Seamless Integration: Integrates effortlessly into existing CI/CD pipelines with tools like Jenkins and GitHub.

Expert Training: Provides developer-focused courses on secure AI coding practices, enhancing internal capabilities.

Veracode's Proprietary AI

Veracode's AI is responsible by design, meaning it's purpose-built to address security and privacy concerns from the ground up, rather than having them added as an afterthought. This approach focuses on several key areas.

Data and Training

Veracode's AI, particularly in its code remediation tool Veracode Fix, is not trained on open-source code or customer data. This is a critical distinction from many other general-purpose generative AI tools. Instead, Veracode uses its own proprietary dataset, which includes millions of historical scan results and expert-curated patches. This ensures that the AI's suggestions are based on best practices and a validated understanding of security vulnerabilities.

Human Oversight and Transparency

The system is designed with a "developer-in-the-loop" model. The AI provides code fix suggestions, but a human developer must review and choose to implement them. The suggestions are clear and tied to the specific vulnerabilities they address, promoting transparency and trust. This also helps mitigate any potential for "hallucinations" or unreliable outputs that are common in less constrained AI models. Veracode's expert analysts also review the patches, which helps ensure accuracy and further refines the model over time.

Security and Privacy

Veracode's responsible AI design prioritizes the security of the customer's data. All customer code is encrypted both in transit and at rest, and is not used to train the AI model. This architecture insulates customers from risks related to intellectual property, copyright, and licensing. The system is built to provide consistent and reliable solutions without retaining customer code, ensuring privacy and preventing the introduction of insecure or malicious code.

CHAPTER 5:

Future Trends and Predictions in AI Security

Emerging Trends to Monitor

The future of AI security will be shaped by several key trends:

1. **Increased AI Model Auditing:** By 2027, **80% of organizations are projected to audit AI models for bias and security,** according to a 2025 Forrester report.
2. **Zero Trust Integration:** GenAI will increasingly align with zero trust architectures, necessitating rigorous verification of all code outputs.
3. **Evolving Regulatory Landscape:** New regulations, such as the EU AI Act, will mandate comprehensive security checks for AI-generated code by **2026.**

Forward-Looking Predictions

Key predictions for the future include:

Automated Remediation

AI tools are expected to evolve to suggest and even implement fixes for their own identified vulnerabilities.

Mandatory Developer Upskilling

By **2028, 90% of developers will require specialized AI-specific security training.**

Persistent Open-Source Risks

Vulnerable open-source libraries will continue to be a significant threat, requiring ongoing vigilance.

CONCLUSION:

Securing the Future of Software Development with AI

Generative AI is undeniably revolutionizing software development, offering unprecedented speed and innovation. However, this transformation introduces novel and critical security risks. By proactively understanding these challenges and implementing robust solutions like **Veracode's Application Risk Management Platform**, organizations can develop **secure, compliant, and highly efficient applications**. For CISOs, security teams, and developers, prioritizing security is not merely a defensive measure; it is the **fundamental enabler for realizing AI's full potential**.

Are you prepared to secure your AI-driven development initiatives?

Visit veracode.com for access to valuable resources, product demonstrations, and expert consultations.

VERACODE