TOP 5 Vibe Coding SECURITY RISKS *You Can't Ignore*

Vibe coding refers to the increasingly common practice of leveraging AI-powered tools and generative models to rapidly produce code snippets, entire functions, or even complete applications. It's about getting the 'vibe' of the code quickly, accelerating development cycles, and boosting productivity.

However, this speed comes with a significant caveat: the code generated by AI is not inherently secure. As we embrace this new era of automated code creation, understanding the potential pitfalls is paramount.

1 *Insecure Code Generation*

AI models can produce code snippets that contain common and critical vulnerabilities such as SQL injection, cross-site scripting (XSS), insecure direct object references, hard-coded credentials, and authentication/ authorization flaws. These are often replicated from insecure patterns found in their training data.

2 *Lack of Contextual Security*

AI tools often generate code without a deep understanding of the specific application's security context, unique business logic, or the broader system architecture. This can lead to code that, while syntactically correct, misses crucial security configurations, access controls, or appropriate input validation for the specific environment it's intended for.

3 *Over-reliance and Blind Trust* *(Skill Erosion)*

Developers, especially those under pressure or with less security expertise, may blindly trust and integrate AI-generated code without thorough review or understanding of its implications. This "comprehension gap" means vulnerabilities can easily go unnoticed, and developers' own security critical thinking and manual code review skills may erode over time.

4 *Insecure Dependencies and Supply Chain Risks*

AI can introduce vulnerable or outdated third-party libraries and dependencies. In some cases, AI tools might even "hallucinate" non-existent package names, which attackers can then register (typosquatting) to inject malicious code into projects. This expands the attack surface significantly through the software supply chain.

5 *Sensitive Data Exposure*

If AI models are given access to internal or sensitive data (e.g., API keys, internal documents, proprietary algorithms) through prompts or training, there's a risk of this sensitive information being inadvertently exposed. The AI might regurgitate parts of its training data or insecurely handle credentials within generated code.

VERACODE

Contact us today to learn how to *vibe* securely with Veracode.

Contact our team at **www.veracode.com** to schedule a demo or request additional information.