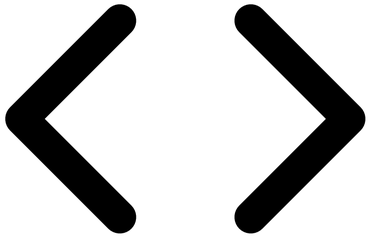
AI Coding & Cybersecurity

Why AI coding tools aren’t ready for production yet

Michele Magri

The AI Revolution

Scale and Speed of adoption



62% of developers use some form of AI assistance (as of 2024)

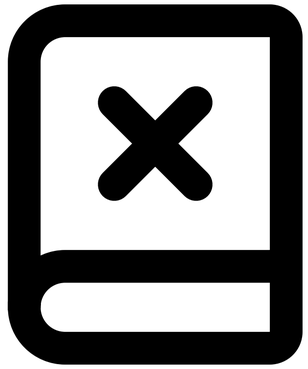
Most of them trust the LLM output and don’t double check the code

Source: 2024 Stack Overflow Developer Survey

The vulnerability landscape

What are we really writing in our code?

The AI generated code shouldn’t be any less secure than human-written code, right?

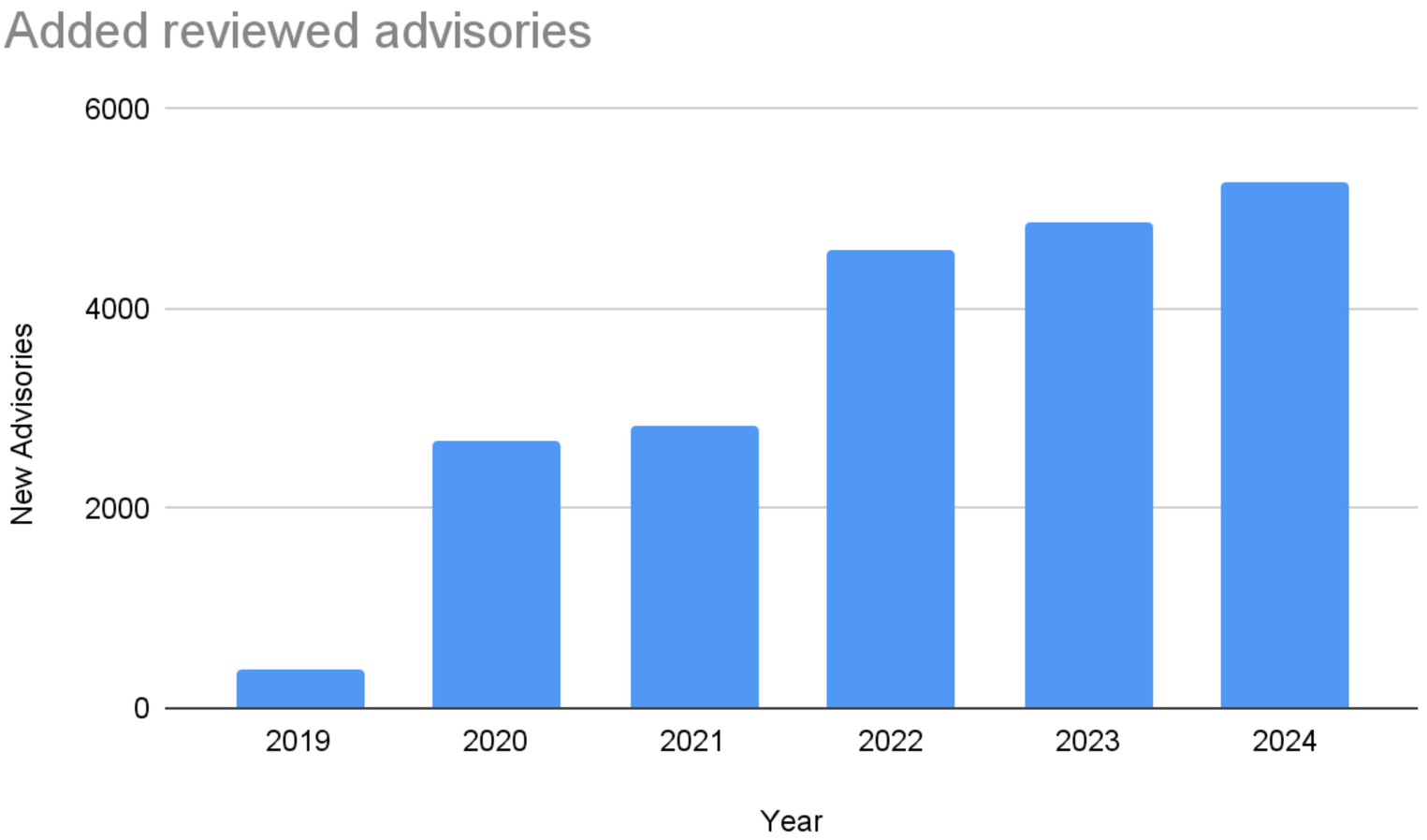


AI agents aren’t able to recognize common vulnerabilities

Source: Security Weaknesses of Copilot-Generated Code in GitHub Projects: An Empirical Study

The vulnerability landscape

What are we really writing in our code?



Source: GitHub Advisory Database by the numbers: Known security vulnerabilities and what you can do about them

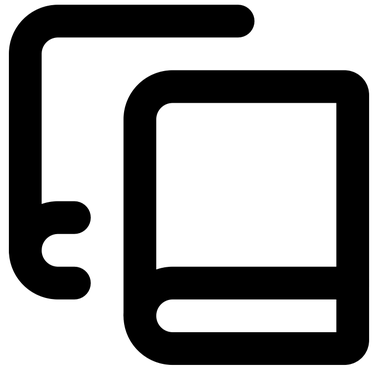
Supply chain in AI generated code

What dependencies should the AI use?

- AI systems can’t reason the choice behind the usage of libraries yet
- No CVE checks or security assessments
- Can’t recognize malicious code in libraries by themselves
- Developers should always evaluate the AI choices

The training data problem

Vulnerabilities in —> Vulnerabilities out



LLMs learn from existing
content and repositories

Source: Github Copilot's FAQ

Privacy and Data protection

AI agents are exposed to important data!

- AI systems/services are often exposed to company code, technologies and sensitive database data
- We can’t know how the service handles and if it stores this data
- We should treat all the data exposed to such services as public
- Organizations should evaluate if sharing data is acceptable or not

The mitigation

Transparency is the key!

- Many AI tools operate as black-boxes
- Impossible to asses the reliability
- We need clear policies and laws for AI development and usage
- Policies should promote the transparency of the technology
- Organizations must be punished heavily for infringing such policies

But what can we do now?

Security-first AI prompting

You have this powerful weapon, use it!

- Use security-conscious prompts
- These prompts can drastically reduce vulnerabilities

Source: Security Weaknesses of Copilot-Generated Code in GitHub Projects: An Empirical Study

Review imperative

Human review of AI generated code

- Reviewing AI generated code is not just a technological need
- Developers have a professional responsibility behind what they deploy!

Automated Security Integration

Tools are your friend!

- There are a lot of tools that do automatic security assessment of your code
- Chef-Inspec, Snyk…
- MCP Servers
- Human review is still required!

Education and Skill-Development

AI assisted coding is a learnable skill

- AI system integration in development workflows require specific education
- Developers need to understand AI limits and learn code reviewing techniques
- Schools and Universities could assist in this.

Are we going to use AI responsibly to build faster and develop more secure software, or will we create new vulnerabilities that threaten our digital future?