Lecture VI — Working Safely with AI Code

AI-Assisted Programming for PhD Researchers

Dr. Tobias Vlćek
Helmut Schmidt University — September 2026

The Numbers

How Bad Is It?

Independent audits keep landing in the same place.

. . .

- About **45%** of AI-generated samples carry OWASP-class vulnerabilities — 80 tasks across 100+ models [1]
- AI-authored pull requests: **~1.7× more issues** overall, up to **2.74×** for XSS specifically — correlational, n=470 PRs [2]

And It Replicates

- This is not a new-model glitch: **~40%** of Copilot programs were already vulnerable in the 2022 academic baseline [3]
- Three years, different tools — the rate barely moves

Why?

- Trained on public code — including the insecure half
- Optimized for plausible, not for safe
- No adversarial mindset — it does not ask “who attacks this?”
- Your context rarely says “this will face the internet”

And You’ll Feel Safer Anyway

- Stanford ran a randomized trial with and without an AI assistant

. . .

- Participants with the assistant wrote **less secure** code, and rated it **more secure** than the control group did [4]
- Overconfidence is the failure mode; the review habits below are the counter

Security Failure Modes

Injection

Plausible, passes tests, ships a hole. Spot the problem:

```
def get_measurements(conn, station):
    query = f"""
        SELECT * FROM readings
        WHERE station = '{station}'
    """
    return conn.execute(query).fetchall()
```

. . .

The `station` value is pasted straight into SQL. Pass it as a parameter (`?`), never an f-string.

Secrets in Code

- Hardcoded API keys, tokens, and passwords in tracked files
- Agents echo the patterns they saw in training — including bad ones
- AI-assisted commits leak secrets at **3.2%** vs a **1.5%** baseline [5]

The Secrets Rule

> ⚠ Warning
>
> Secrets live in environment variables or your keychain — never in files the agent writes, and never pasted into a prompt.

Hallucinated Dependencies

- The agent imports a package that does not exist
- Squatters register those invented names — "slopsquatting"
- Check every new dependency it adds: real? maintained? license?

The Quiet Ones

- Path traversal: unsanitized filenames escaping a directory
- Unsafe `eval` or `pickle` on data you did not create
- Missing input validation on the boring, trusting path

You do not memorize these — you build review habits and checks.

Prompt Injection

- Untrusted content the agent reads (a webpage, a tool result, a file) can carry instructions it then obeys
- The term was coined in 2022 [6]
- The rule: never combine the lethal trifecta — private data + untrusted content + external communication [7]

Hallucination and Overreliance

Confidently Wrong

- Invented APIs, wrong parameter names, outdated idioms

- Delivered in the same confident tone as correct code
- Training cutoffs mean the model's world is months old

Countermeasures

- Verify against real docs — MCP docs servers, this afternoon
- Run the code; a claim is not a result
- Prefer boring, well-documented libraries over the clever new one

Licensing and IP

Who Owns AI Output?

- Legally unsettled, and it depends on your jurisdiction
- Courts and institutions are still deciding
- Practical stance: treat it as your code and your responsibility

Three Practical Rules

- Check the licenses of any dependencies it pulls in
- Do not paste licensed code in as context and ask for a "rewrite"
- Know your institution's policy — it may already bind you

Reviewing AI Code

Read Every Diff

- Non-negotiable: every line, every time
- `git add -p` turns review into a ritual, hunk by hunk
- If the diff is too big to review, the step was too big — redo it smaller

Writer and Reviewer

- A fresh agent session reviewing the diff has no attachment to the code
- Review against the spec, not your taste — you arbitrate
- A checklist you are handed does not transfer; it has to be practiced [8]
- That practice is exactly what Lab 4 is

What Review Catches vs What Tests Catch

Tests catch	Review catches
Behavior regressions	Design smells
Broken edge cases	Security holes
Wrong outputs	Silent data handling

You need both — neither substitutes for the other.

Your Data

What Leaves Your Machine

- Every prompt you type

- Every file the agent reads
- Every command output it sees

All of it goes to the model provider's API. Plain fact, not a scare.

Where It Goes Matters

- Jurisdiction, retention, and whether it trains future models
- Our stack: Mistral, EU-hosted, training opt-out. You set that in setup, check it now
- Zed-hosted models run in the US — know which you are calling

Unpublished Research Data

Before you send it, ask:

- Personal data under GDPR?
- Under embargo or an NDA?
- Owned by a third party?

Any "yes" → use a synthetic sample or a local model. Your data-management plan may already answer this.

Rules of Thumb

> ⚠ Warning
>
> Lab data is synthetic, so send it freely. Your own data: check first, sample small, anonymize. Secrets and credentials: never.

Academic Integrity

Disclosure Is the Norm Now

- The rules converged over 2023–2024 and they agree
- Nature and Science: AI cannot be an author, and its use must be disclosed [9], [10]

The Rules Agree Everywhere

- COPE and ICMJE extend that to thousands of journals [11], [12]
- DFG, ALLEA, and the EU living guidelines say the same for funders and integrity codes [13], [14], [15]
- A settled expectation, not a frontier

How to Disclose

- In methods or acknowledgments: which tools, for what, and that a human verified the output
- Keep your prompts and session logs — they are your lab notebook

In This Course

- Disclosure is part of your final presentation — the process-reflection section
- It is assessed content, not a confession

Lab 4

Your Mission

In Lab 4 the agent works fast and unsupervised on a branch:

- It builds three features without asking you questions
- You find what is wrong with them before they reach main
- Then you earn the merge through review, not vibes

Lab 4 — Review and Explain

Continue Your Journey

Next Up

- You can now spot the failure modes and review AI code deliberately
- Lab 4 — Review and Explain starts now
- Lecture VII — Extending the Agent — 13:15
- Course literature and references

Bibliography

[1] Veracode, "2025 GenAI Code Security Report." Accessed: July 09, 2026. [Online]. Available: https://www.veracode.com/wp-content/uploads/2025_GenAI_Code_Security_Report_Final.pdf

[2] "AI vs human code gen report: AI code creates 1.7x more issues." Accessed: July 09, 2026. [Online]. Available: https://coderabbit.ai/blog/state-of-ai-vs-human-code-generation-report

[3] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions," in *2022 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA: IEEE, May 2022, pp. 754–768. doi: 10.1109/SP46214.2022.9833571.

[4] N. Perry, M. Srivastava, D. Kumar, and D. Boneh, "Do Users Write More Insecure Code with AI Assistants?," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, Copenhagen Denmark: ACM, Nov. 2023, pp. 2785–2799. doi: 10.1145/3576915.3623157.

[5] A. Nabiullina and C. Winqwist, "The State of Secrets Sprawl 2026: AI-Service Leaks Surge 81% and 29M Secrets Hit Public GitHub." Accessed: July 09, 2026. [Online]. Available: https://blog.gitguardian.com/the-state-of-secrets-sprawl-2026/

[6] S. Willison, "Prompt injection attacks against GPT-3." Accessed: July 09, 2026. [Online]. Available: https://simonwillison.net/2022/Sep/12/prompt-injection/

[7] S. Willison, "The lethal trifecta for AI agents: private data, untrusted content, and external communication." Accessed: July 09, 2026. [Online]. Available: https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/

[8] D. J. Bouvier *et al.*, "The Rest of the Robots: Generative AI in Post-introductory Computing Education," in *Proceedings of the 2025 Working Group Reports on Innovation and Technology in Computer Science Education*, Nijmegen Netherlands: ACM, June 2025, pp. 61–107. doi: 10.1145/3760545.3783970.

[9] "Tools such as ChatGPT threaten transparent science; here are our ground rules for their use," *Nature*, vol. 613, no. 7945, p. 612, Jan. 2023, doi: 10.1038/d41586-023-00191-1.

[10] H. H. Thorp, "ChatGPT is fun, but not an author," *Science*, vol. 379, no. 6630, p. 313, Jan. 2023, doi: 10.1126/science.adg7879.

[11] Committee on Publication Ethics, "Authorship and AI tools." Accessed: July 09, 2026. [Online]. Available: https://publicationethics.org/guidance/cope-position/authorship-and-ai-tools

[12] International Committee of Medical Journal Editors, "Recommendations for the Conduct, Reporting, Editing, and Publication of Scholarly Work in Medical Journals: Defining the Role of Authors and Contributors." Accessed: July 09, 2026. [Online]. Available: https://www.icmje.org/recommendations/browse/roles-and-responsibilities/defining-the-role-of-authors-and-contributors.html

[13] Deutsche Forschungsgemeinschaft, "Statement of the DFG Executive Committee on the Influence of Generative Models for Text and Image Creation on Science and the Humanities and on the DFG's Funding Activity." Accessed: July 09, 2026. [Online]. Available: https://www.dfg.de/download/pdf/dfg_im_profil/geschaeftsstelle/publikationen/stellungnahmen_papiere/2023/230921_statement_executive_committee_ki_ai.pdf

[14] ALLEA - All European Academies, *The European Code of Conduct for Research Integrity*. DE: ALLEA - All European Academies, 2023. doi: 10.26356/ECoC.

[15] European Commission, "Living guidelines on the responsible use of generative AI in research." Accessed: July 09, 2026. [Online]. Available: https://research-and-innovation.ec.europa.eu/document/download/2b6cf7e5-36ac-41cb-aab5-0d32050143dc_en?filename=ec_rtd_ai-guidelines.pdf