[Securely] Diving in with Vibe Coding

Prepare for the next wave of [secure] development!

/zoebraiterman

@zbraiterman

@zbraiterman

Summary

- How vibe coding can increase productivity when coding
- Ways you can build security into these workflows
- Resources that may help along the way

$ whoami

Zoe Braiterman

- IT / Security consultant
- Web3 executive - Currently bridging technical and knowledge gaps between the worlds of Web2 and Web3.
- Hackathon enthusiast
- OWASP roles: Threat Modeling Manifesto Co-Author, Blockchain AppSec Standard Co-Leader, NYC Chapter Leader

Inspiration for This Talk

- Getting something live on a blockchain network during a bootcamp/hackathon
- Moving fast and breaking things with my dev

What is vibe coding?

- “Vibe Coding” = building software with awareness of the **human experience**: clarity, safety, and maintainability as shared values.
- Developer mood, design ethics, and security posture are connected.
- Account for context, while iterating

Vibe Coding and Security

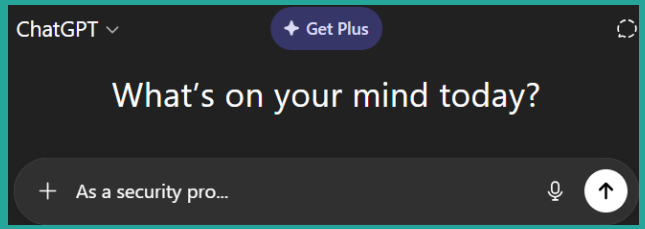
- AI accelerates output but can also accelerate vulnerabilities
- Common AI-assisted defects:
  - Injection bugs
  - Broken auth
  - Weak crypto
  - Insecure defaults
- Secure “vibe” = AI coding tools + guardrails (security policies, AI training, security verification)

Build Security into Your Vibe Coding Workflow

The Flow of Security Energy

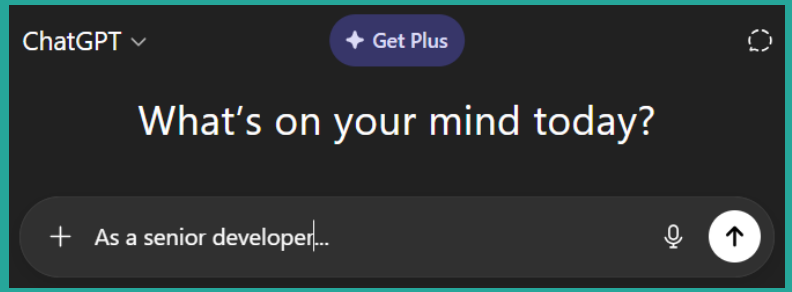
- Flow state drives productivity — but unguarded flow breeds risk.
- Integrate guardrails early: linting, secure IDE hints, AI-assisted prompt packs.
- Replace friction with awareness and intent: input validation, auth boundaries, context-aware trust.

The Human Layer of Security



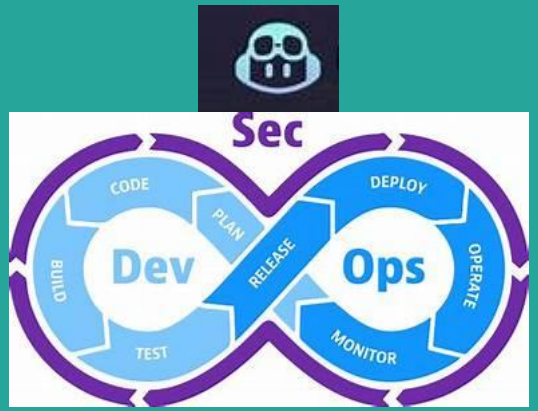
- Most breaches trace to human shortcuts under pressure.
- Vibe-aligned security culture = psychological safety + technical rigor.
- Encourage “pause moments”: code review, threat modeling, self-check.
- Developers protect users best when they’re not afraid to slow down.

Vibe Security Principles



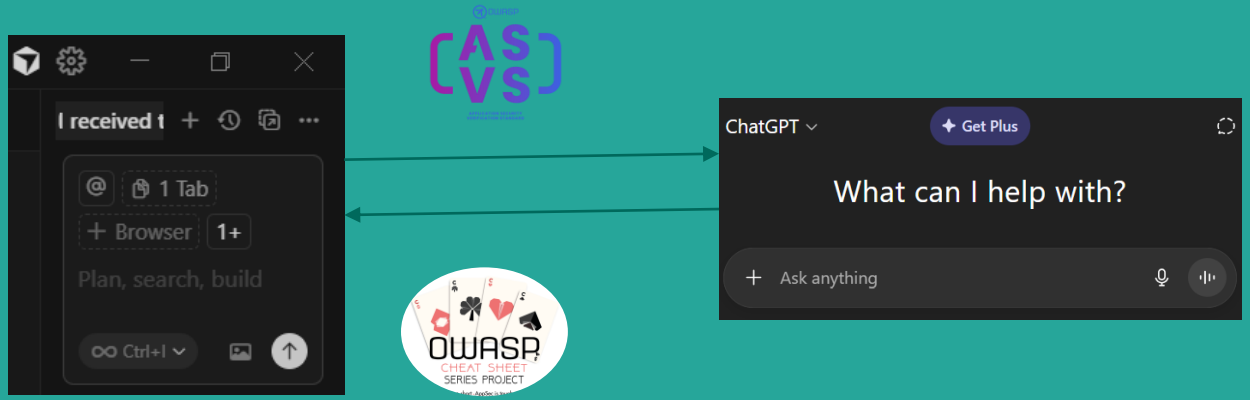
- Authenticity → write code that represents intent clearly.
- Harmony → minimize entropy, reduce attack surface.
- Resonance → align secure design with user trust.
- Silence → less code = fewer bugs = smaller exploit window.

The Secure Vibe Culture



- Security is emotional architecture: the trust people feel when systems behave.
- In teams: feedback loops > checklists.
- In AI systems: explainability > opacity.
- In organizations: *security by vibe* = consistent rhythm of care, clarity, and courage.

A Possible Vibe Coding Workflow



The developer uses Cursor’s AI-powered editor to generate and refactor code securely in real time, applying best practices directly as they code.

The developer consults ChatGPT for deeper explanations, secure coding guidance, and quick checks against OWASP standards/guides while vibe coding.

Resources

Get community guidance throughout your journey…

Teaching Security Through Vibe

Adam Shostack’s Four Question Framework for Threat Modeling

- What are we working on?
- What can go wrong?
- What are we going to do about it?
- Did we do a good job?

OWASP Resources

- **Cheat Sheet Series:** Practical guidance on securely implementing specific coding tasks during development
- **ASVS:** Define and measure the required level of security assurance for the application
- **Proactive Controls:** Identify and prioritize key security activities to integrate into the software development lifecycle
- … Engage with the community!

Thank you

Keep in touch!