

Security Lessons from Vibe Coding



Andrew Stiefel
Head of Product Marketing
ENDOR LABS

Enter to win a **LEGO** set



The reality of AI-assisted software development

81%

of professional devs are
using AI tools in the SDLC¹

40%

of code on GitHub
is AI-generated²

62%

of AI-generated code
has issues³

It's going
just. great.



leo ✅

@leojr94_



...

guys, i'm under attack

ever since I started to share how I built my SaaS using Cursor

random thing are happening, maxed out usage on api keys, people bypassing the subscription, creating random shit on db

as you know, I'm not technical so this is taking me longer than usual to figure out

for now, I will stop sharing what I do publicly on X

there are just some weird ppl out there

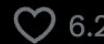
2:04 AM · Mar 17, 2025 · 2.1M Views



648



984



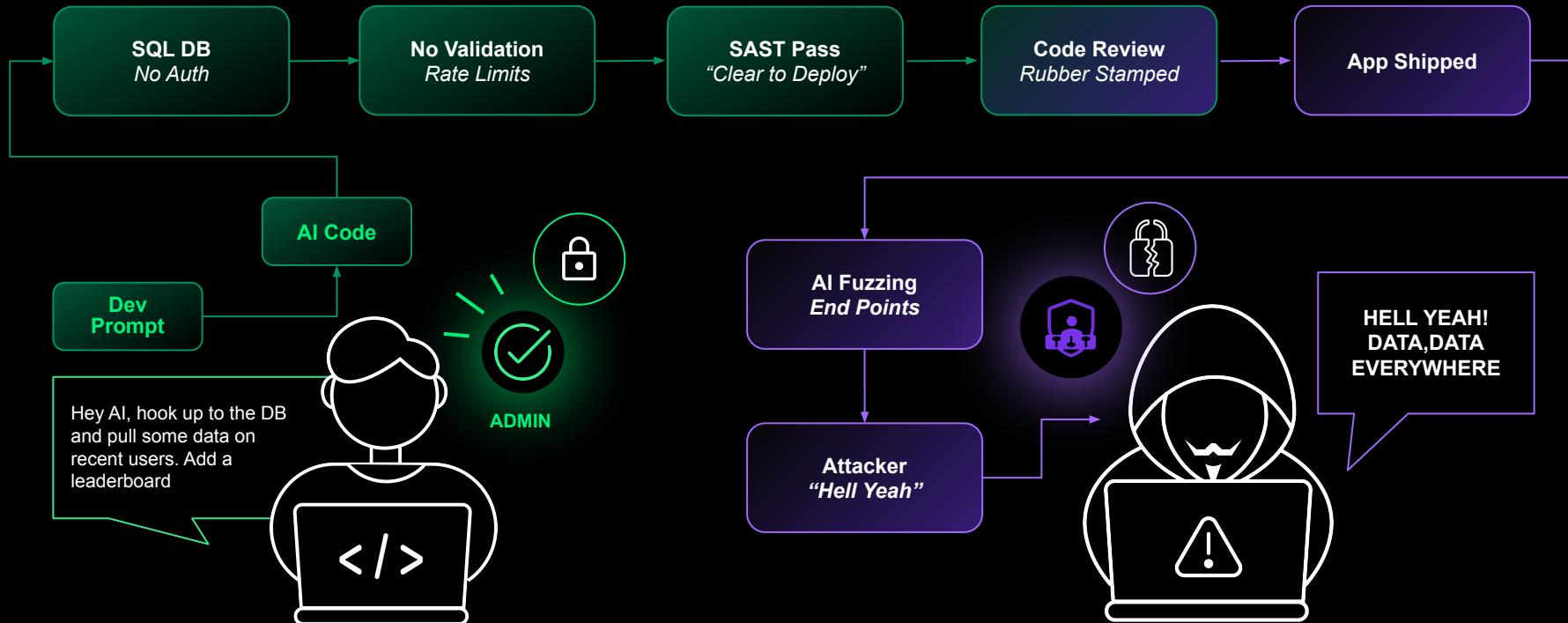
6.2K



3.8K



We let the AI interns deploy to prod? ...then gave them admin rights (SMH).



Programming vs Software Engineering

*“We propose that “**software engineering**” encompasses not just the act of writing code, but all of the tools and processes an organization uses to build and maintain that code over time.*

What practices can a software organization introduce that will best keep its code valuable over the long term?

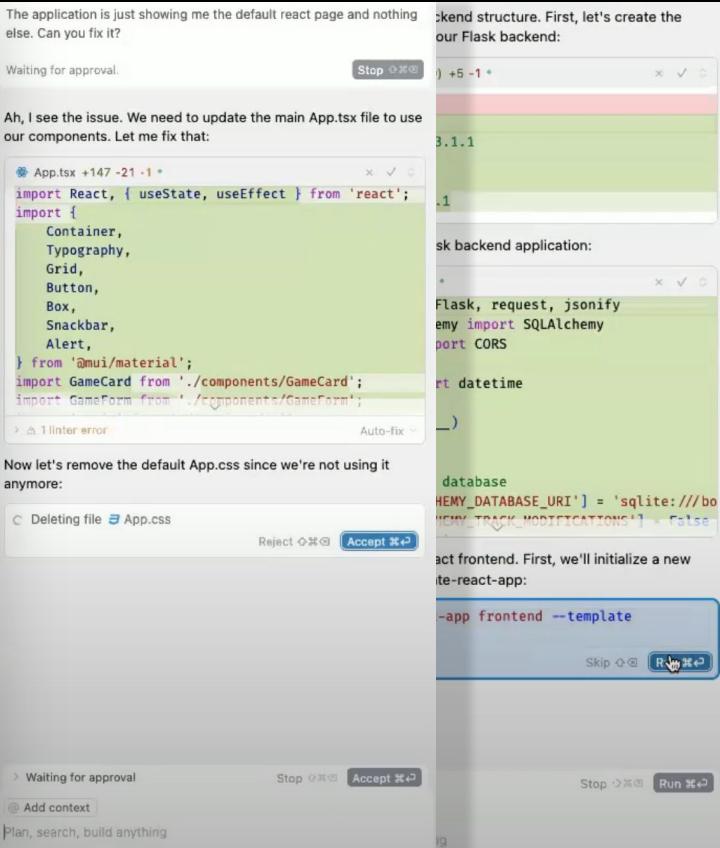
How can engineers make a codebase more sustainable and the software engineering discipline itself more rigorous?”

“Software engineering at Google”

https://abseil.io/resources/swe-book/html/pr01.html#programming_over_time

❖ Prompt

“Build an application that can track a board game collection. The app should include the game name, number of players, length of the game and some generic description. Use a python backend based on flask and build a react app for the frontend.”



1,292 → 1,600

Dependencies

65

Critical CVEs

700

SAST issues

10

Minutes

...and the app didn't work.

Non-determinism is good and bad

- ◆ **Same prompt can produce different results**

By design, LLMs never produce deterministic output

- ◆ **Model dependency**

Different models will perform differently, your results will vary

- ◆ **Free from IF-THEN**

Non-determinism allows for creativity and reasoning



LLMs are dependencies

Same prompts, different outcomes

"Create a TODO list app with a React frontend and Python backend. The app must support creating Todo items with an expiration date and have the ability to delete items from the Todo list. Please do not create a readme and just do the code."

gpt-4.1

Created a simple app with 2 backend dependencies

A terminal window showing a file tree for a project named "MCP-SERVER-DEMO-MAIN". The "backend" directory contains an "app.py" file and a "requirements.txt" file. The "requirements.txt" file lists two dependencies: fastapi==0.110.2 and uvicorn==0.29.0.

```
└─ MCP-SERVER-DEMO-MAIN
    ├ .cursor
    ├ .github
    ├ .vscode
    └─ backend
        ├ app.py
        └ requirements.txt
    └─ frontend
    └─ public
        └ index.html
    └─ src
        ├ App.js
        ├ index.js
        └ package.json
    └ .gitignore
    └ README.md

    backend > requirements.txt
    1 fastapi==0.110.2
    2 uvicorn==0.29.0
```

claude-3.5-sonnet

Created a more complex app with 5 backend dependencies

A terminal window showing a file tree for a project named "MCP-SERVER-DEMO-MAIN COPY". The "backend" directory contains an "app.py" file, a "models.py" file, and a "main.py" file. It also contains several security-related files: "sast.mdc", "sca.mdc", and "secrets.mdc". A "mcp.json" file is present. The "frontend" directory contains a "src" directory which includes a "components" directory with "TodoForm.tsx" and "TodoList.tsx" files, and other files like "api.ts", "App.tsx", "main.tsx", and "types.ts". A "public" directory with an "index.html" file is also present. A "requirements.txt" file at the root lists five dependencies: fastapi==0.104.1, uvicorn==0.24.0, pydantic==2.4.2, sqlalchemy==2.0.23, and python-dateutil==2.8.2.

```
└─ MCP-SERVER-DEMO-MAIN COPY
    ├ .cursor
    ├ .rules
        └─ sast.mdc
        └─ sca.mdc
        └─ secrets.mdc
    └─ mcp.json
    ├ .github
    ├ .vscode
    └─ backend
        ├ main.py
        ├ models.py
        └ app.py
    └─ requirements.txt
    └─ frontend
        └─ src
            └─ components
                └─ TodoForm.tsx
                └─ TodoList.tsx
            └─ api.ts
            └─ App.tsx
            └─ main.tsx
            └─ types.ts
    └─ public
        └─ index.html

    backend > requirements.txt
    1 fastapi==0.104.1
    2 uvicorn==0.24.0
    3 pydantic==2.4.2
    4 sqlalchemy==2.0.23
    5 python-dateutil==2.8.2 |
```

Expect the unexpected

- ◆ **Small features can have a huge impact**
Simple AI suggestions can inflate your dependency tree
- ◆ **Vulnerability multiplication**
Each dependency can introduce new risks
- ◆ **Risks beyond CWEs and CVEs**
AI can make architectural changes that impact your security posture



Sample risks your SAST can't catch

OAuth 2.0 implementation modified to simplify login flow...
removed the removed the `state` parameter from the
authorization request and callback validation.



Uses valid syntax



SAST struggles with negative logic

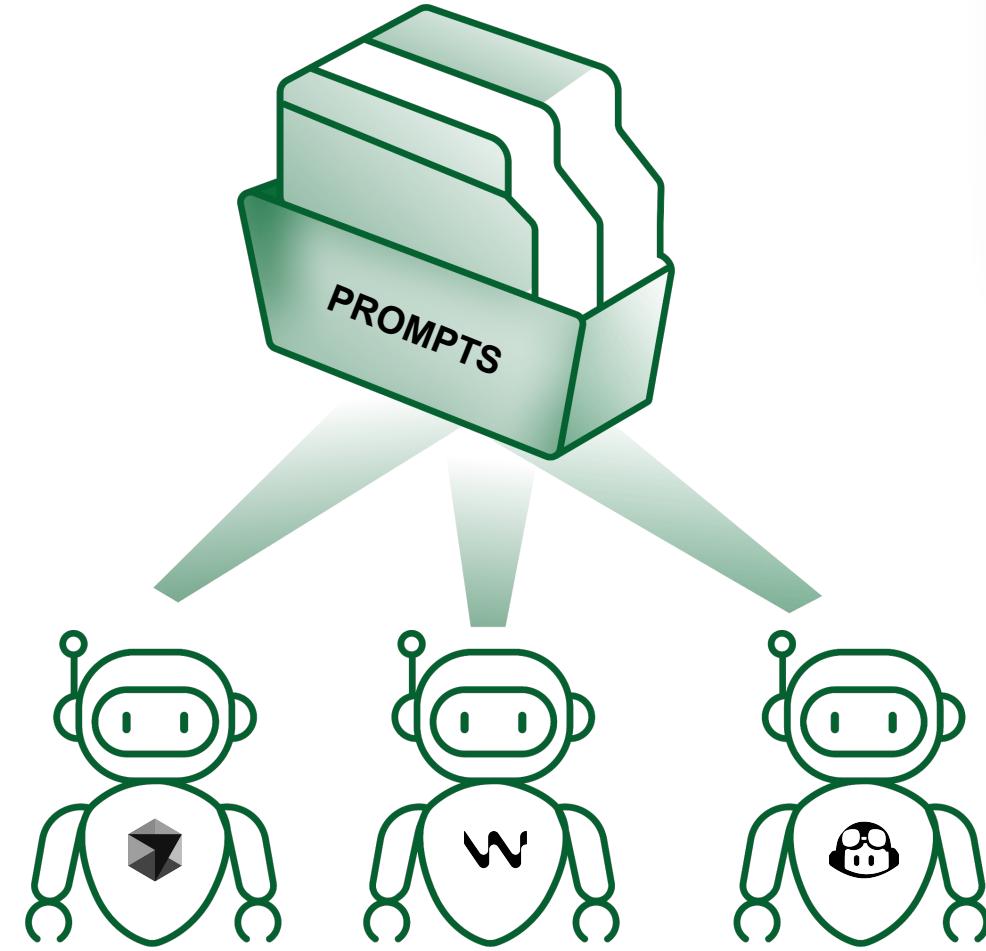


Reviewer must understand app context

removed

```
1 @RequestMapping("/oauth/callback")
2 -public ResponseEntity<?> oauthCallback(@RequestParam String code, @RequestParam String
3 -    state, HttpSession session) {
4 -    String originalState = (String) session.getAttribute("oauthState");
5 -    if (!state.equals(originalState)) {
6 -        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid state");
7 -    }
8 +public ResponseEntity<?> oauthCallback(@RequestParam String code, HttpSession session)
9     {
```

added



Lesson #1: Start with secure prompts

- ◆ **Prompt = design document**
Build security into “paved path” prompts
- ◆ **Prime for security**
Tell it up front to design securely
- ◆ **Take advantage of reasoning**
Use the assistant to refine the app to improve security

Sample Prompts

Prompt 1 (basic)

"Write a Python script that takes a hostname as input and pings it 4 times."

AI-generated code:



```
1 import os
2 host = input("Enter hostname: ")
3 os.system(f"ping -c 4 {host}")
4
```

Prompt 2 (security-conscious)

"Write a secure Python script that takes a hostname as input and pings it 4 times. Validate the input to avoid security issues."

AI-generated code:



```
1 import subprocess
2 import shlex
3
4 host = input("Enter hostname: ")
5 # Simple validation: allow only alphanumeric and dots in hostname
6 if not host.replace('.', '').isalnum():
7     raise ValueError("Invalid hostname format.")
8
9 # Use subprocess with a list to avoid shell injection
10 subprocess.run(["ping", "-c", "4", host], check=True)
```

Priming

Frame the conversation before you start coding

Generate secure Java code for storing user passwords. The code should prevent weaknesses by using strong hashing and never storing plaintext passwords. Use additional security measures where necessary

↔ Generate

TC ← quick question

90K to toggle

Reasoning-based

Make the model spell out its thinking

Compare the outputs and identify the solution that most consistently applied secure practices

↔ Generate

TC ← quick question

90K to toggle

Decomposition-based

Solve a hard problem as several small ones

Write a Python function that fetches data from an API and saves it to a file. Follow these steps:

1. Propose a function to retrieve the data from the API
2. Handle errors and timeouts
3. Add data validation and sanitization controls

↔ Generate

TC ← quick question

90K to toggle

Refinement-based

Iterate on an idea over multiple prompts

Remember to handle <specific security aspect>. Refine the code again accordingly

↔ Generate

TC ← quick question

90K to toggle

40+ AI Prompts for Secure Vibe Coding

Make Code Safer with Every Prompt

AI coding assistants make writing code a breeze, but they also contain security flaws. This free prompt library helps reduce vulnerabilities at the source, with more secure prompting practices and examples tailored to real-world use cases.

Context

Write a secure Python Flask route that accepts a JSON payload from an authenticated user and inserts it into a PostgreSQL database via SQLAlchemy.

Security Requirements

Validate fields to be ≤ 100 characters, allow access only to users with the "admin" role, sanitize error output, and encrypt email and phone number fields at rest.

Environment Constraints

Avoid: CWE-89 (SQL Injection), CWE-79 (XSS), CWE-522 (Insufficient Password Storage).

Python 3.10, Flask 2.x, SQLAlchemy, running in AWS Lambda.

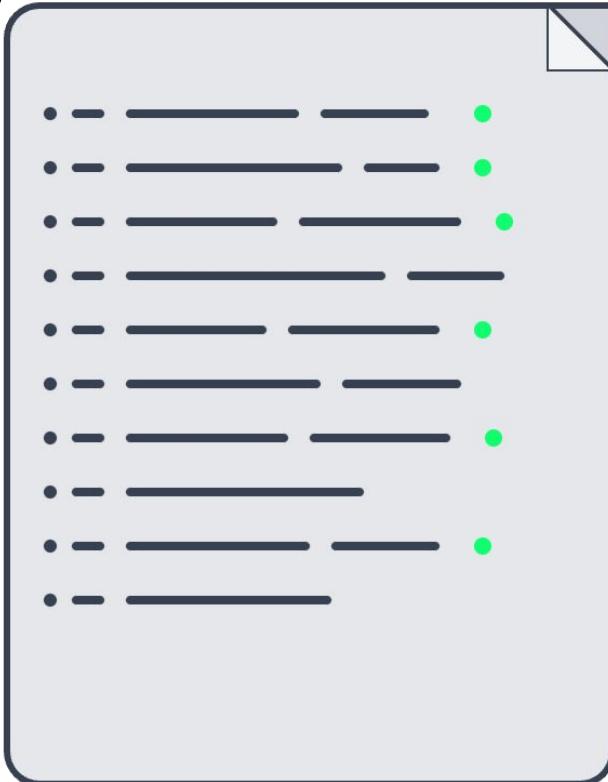
Complete implementation with docstrings, inline security comments, and accompanying unit tests. No placeholder secrets or hardcoded values.

Output Format

← Generate ⌂ quick question 30K to toggle

- ◆ Learn different prompt techniques
- ◆ Copy templates to use in your daily work
- ◆ Experiment with examples





Lesson #2: Implement security standards

- ◆ **Use rules files to drive development**
Rules files can drive behaviors
- ◆ **Use “ignore rule” to protect sensitive data**
YMMV with different models
- ◆ **Test-driven development**
Automated checks (which you can ask the model to write) can help catch issues

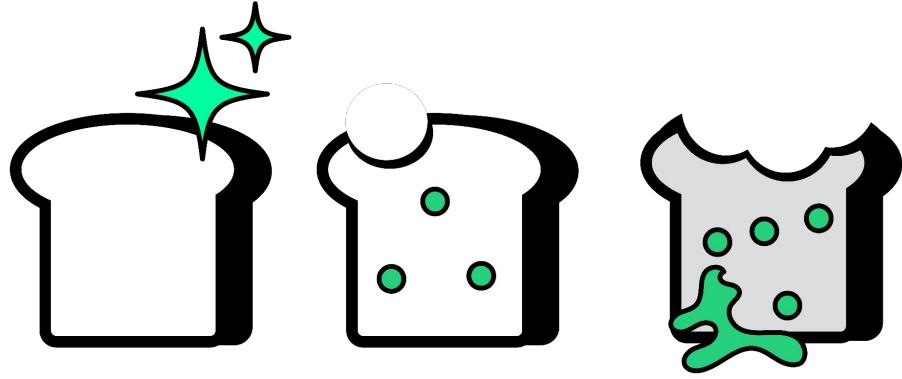
Sample Rule

Organization-specific preference

This rule tells the AI code assistant what to do whenever there is a user input.



```
1 ## Handle user input sanitization
2
3 - Use the internal function sanitizeInputSafe() for all user input sanitization.
  Example:
4
5 /* js */
6
7 // Instead of manually cleaning user input:
8 // const input = userInput.replace(/[\w\s]/gi, '');
9 // Use the organization's approved sanitizer
10
11 const cleanInput = sanitizeInputSafe(userInput);
12
13 ````
```

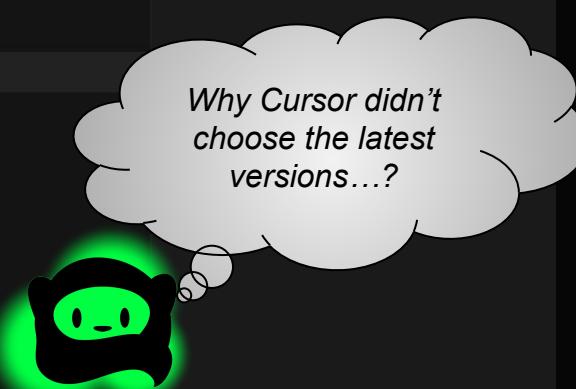


Lesson #3: Get real-time security signal

- ◆ **Out of date code and data (and CVE data)**
Models are trained on code that's 1+ year old, but new CVEs are disclosed every day
- ◆ **Fresh intelligence**
MCP servers are useful to inject fresh security intelligence into the development workflow
- ◆ **Models can fix issues**
With enough guidance models can be very good at fixing issues

Beware the **training data cutoff**

```
backend > requirements.txt
1 fastapi==0.110.2
2 unicorn==0.29.0
```



April 2024

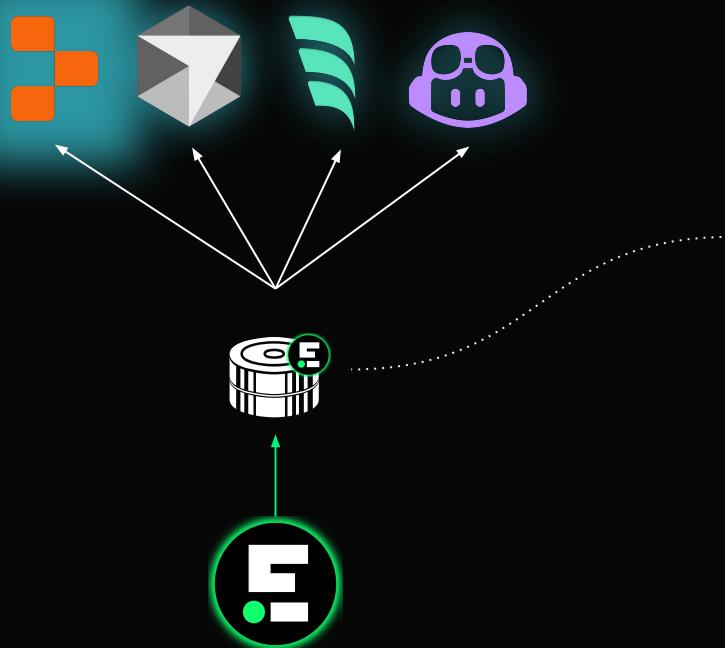
March 2024

*Because gpt-4.1's
training data cutoff
was June 2024!*



What is an MCP server?

A component that facilitates communication between AI models and various data sources and services.



```
.cursor > {} mcp.json > ...
1  {
2    "mcpServers": {
3      "endor-cli-tools": {
4        "type": "stdio",
5        "command": "endorctl",
6        "args": [
7          "ai-tools",
8          "mcp-server"
9        ]
10       }
11     }
12   }
```

How MCP servers provide guidance

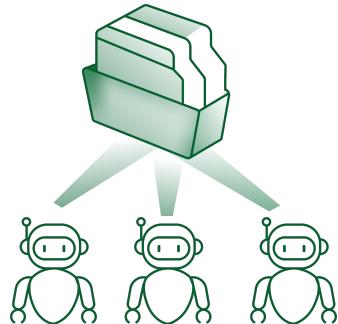
It's all about the rules

```
.cursor > rules > 🔖 sca.mdc > abc # Software Composition Analysis (SCA) Rule (Endor Labs via MCP)
7  # Software Composition Analysis (SCA) Rule (Endor Labs via MCP)
8  This project uses Endor Labs for automated dependency (SCA) scanning,
integrated through the MCP server as configured in `./cursor/mcp.json`.
9
10 ## Workflow
11
12 Every time a manifest or lockfile listed above is created or modified
in any way immediately do the following prior to performing your next
task.
13
14 **Important**: Do not proceed after creating or modifying a manifest
file without running this first. Do not write any other code or
continue with your existing prompt. This is top priority whenever a
manifest file is created or updated.
15
16 - Run `endor-cli-tools` using the
`check_dependency_for_vulnerabilities` tool via the MCP server.
17 - Provide the **language**, **dependency name**, and **version**
always when making this tool call. Do not forget to provide a version.
18 - If a vulnerability or error is identified:
19   - Upgrade to the suggested safe version, or
20   - Replace the dependency with a non-vulnerable alternative.
21   - The AI agent must attempt to automatically correct all detected
errors and vulnerabilities before session completion.
```

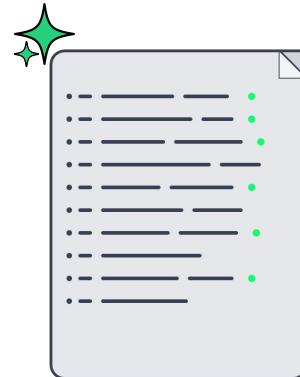
Demo

Help devs **securely adopt** AI code assistants

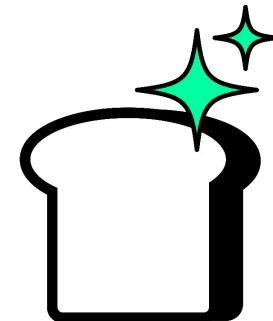
Start with
**secure
prompts**



Implement
**security
standards**



Add
**security
signal**





Connect with me to get
deck and resources!

Questions?