#### DATASHEET

# How Legit Secures Vibe Coding

# What Is Vibe Coding?

Always wanted to create software but held back by lack of coding skills? No longer a problem.

Vibe coding is the latest Al-driven software development trend that keeps developers focused on outcomes, not code. When vibe coding, developers use natural language to voice prompt Al assistants to generate code. Without worrying about the technical details, developers are able to "vibe" – or stay highly focused on creating software.

#### Gartner notes that

"By 2028 40% of new enterprise production software will be created with vibe coding techniques and tools."

Google recently <u>reported</u> Al writes over 30% of new code at the company.

How does security fit into this new reality?

How can developers feel confident about the quality and security of vibe coding output if traditional AST no longer plays a role?

How can security keep up with the volume, and questionable quality, of code teams are now creating?

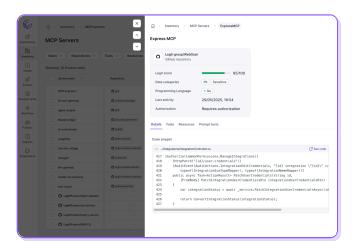
# Legit and Vibe Coding

You keep vibing; we'll worry about security.

Coding is evolving, and Legit is evolving to keep it secure. We offer real-time security insights and remediation embedded directly in AI-powered assistants – no learning curve, no context switching, all natural language.

Through the Legit MCP Server, AI code assistants like Cursor, GitHub Copilot, and Windsurf leverage Legit to determine the security of generated code, enforce guardrails to prevent issues, and drive automated remediation.

# 



### Real-Time Security for Al-Generated Code

#### Legit:

- → Detects vulnerable AI-suggested code snippets.
- → Blocks insecure patterns and secrets introduced by copy/pasted Al output.
- → Flags AI-generated code that violates policy (e.g., use of banned packages, license issues, or risky functions).

Visibility Into	
Al-Generated Coc	le

#### Legit provides:

- → Visibility into code and pipelines introduced during fast-paced AI development.
- → Risk scoring and correlation across repos, pipelines, and development activity.
- → Real-time alerts when insecure behaviors emerge (e.g., pushing unverified Al-generated code to prod).

A large langu	guage Model vulnerable to p age model is being used in an insec	
① Detection time 은 Assignee	06/23/2025 08:51 PM	C LegitPhoenix/lambda_chat Repository
Ticket	Create ticket	Owner Elad
Source tools	Legit	Business impact     ••••     High     Usibility     Private
	Show more	Product units     Production
	Risk Score 77 Remediation Conte	ext Graph
<>> first_function.py 27 chain = Con 28 29 30	nversationalRetrievalChain.f llm=llm,	_ riever(search_kwargs={"k": 3}),

I Powered Remediation Guidance howing how to fix the finding using Legit Al				
The SAST tool is correctly flagging this as a potential security risk. Let's fix it by replacing yam. Full.Loader with yam. SafeLoader, which is the recommended approach for handling untrusted YAML data. Here is a code snippet that can help				
/> load-docs.py الع main				
1 def get_docs_yaml(docs_file):				
2 with open(docs_file) as f:				
- 3 return yaml.load(f, Loader=yaml.FullLoader)				
+ 3 return yaml.load(f, Loader=yaml.SafeLoader)				
4	<u>п</u>			

## The Right Remediation

To keep pace with vibe coding, Legit provides:

- → AI-powered remediation suggestions in the tools developers already use (e.g., IDEs, PRs).
- → Prioritized, contextual guidance that helps developers fix insecure code without slowing down their creative flow.

Want to ensure your Al-generated code is automatically investigated, remediated, and verified?

Contact us for a demo

info.legitsecurity.com/request-a-demo

We integrate with all your vibe coding tools, including: → Cursor
→ Windsurf
→ Co-Pilot
→ Claude

# Build fast with AI. Secure with AI-powered ASPM.

The Legit Security ASPM platform is a new way to manage application security in a world of AI-first development, providing a cleaner way to manage and scale AppSec and address risks. Fast to implement, easy to use, and AI-native, Legit has an unmatched ability to discover and visualize the entire software factory attack surface, including a prioritized view of AppSec data from siloed scanning tools. As a result, organizations have the visibility, context, and automation they need to quickly find, fix, and prevent the application risk that matters most. Spend less time chasing low-risk findings, more time innovating.