MOVING BEYOND "VIBE CODING"

Govern the AI Already in Your Supply Chain with Boost Security

The honeymoon phase of AI-assisted development is over. Now you need a way to govern a new class of non-human authors.

THE PROBLEM

The High Cost of "Vibes"

The early enthusiasm for AI coding relied on vibes and intuition. But as AI moves from prototypes to mission-critical systems, vibes just aren't enough.

The Volume Gap

AI generates code faster than human teams can review it.

The Responsibility Gap

Cultural guidance and "prompt engineering" don't scale.

The Context Gap

Static tools flag theoretical issues; security teams need to solve material risks.

THE BOOST SOLUTION

Security AI Governance Framework

Boost Security provides the infrastructure to verify machine-generated code with the same rigor you apply to any other external or internal risk. Boost addresses AI risk through **three critical angles:**

Build More-Secure AI Apps

Govern the AI application your teams build. Boost Security **scanners** and **ASPM** provide deep visibility into your AI stack, from identifying models in use to scanning for vulns unique to LLM-integrated applications.

Leverage Coding Agents Securely

Don't wait for the pull request to find a flaw. Our **Model Context Protocol (MCP)** Server allows you to enforce security policy directly within the coding agent's environment, with safe package lists and approved models in Boost's centralized **policy engine** and **agent guardrails** to suggest secure code patterns and approved dependencies at the moment of creation.

Remediate Using AI

Close the loop by using AI to fix the problems AI (or humans) created. Boost Security provides **context-aware remediation** directly in the IDE and the PR. Fixes are validated against your specific environment and security policies.

Why Boost?

Boost Sees AI as A New Type of Contributor

Boost treats AI as an **untrusted, high-volume contributor.** We don't wait for LLMs to 'improve' or for developers to write better prompts.

We provide the infrastructure to verify machine-generated code with the same skepticism and rigor you apply to any other external risk.

Boost Gives You Security-Owned Guardrails

At Boost, we believe guardrails should be part of security policy (and shouldn't depend on your developers' prompts).

Centralized Control

Define approved models and acceptable dependencies centrally.

Automated Enforcement

Standards are applied automatically as code enters repositories and PRs.

Scale

Governance remains consistent even as AI usage explodes across teams.

Boost Offers Remediation at the Point of Introduction

The most expensive place to fix a vulnerability is in production. Boost's **Model Context Protocol (MCP) server** allows security teams to remediate where (and when) the code is born.

1

Real-Time Risk Prevention

Validate code against known vulnerabilities before the agent provides it to the dev.

2

1-Click Remediation

Deliver context-aware fixes directly into pull requests and IDEs.

3

Traceable Reasoning

Every fix comes with reasoning, allowing security to retain oversight without adding manual review layers.

Boost Is Designed for the Post-Hype Reality

Boost is built for the **technical debt reality of AI.** We provide a permanent, defensible audit trail for every AI-assisted decision, ensuring that 'the AI suggested it' is never the only line of defense in a post-incident review.

Boost Helps You Know What Really Matters

Unlike standalone AI security tools, Boost Secure AI Development integrates with **Boost ASPM.** By combining AI governance with deep environmental context, we distinguish between theoretical flaws and material exposure and offer context-aware fixes that keep your environment in mind.

Stop reacting to AI risk. Start governing it.

Check out boostsecurity.io to learn more.