Sentoru – Secure Coding for the AI Era

Sentoru is the AI agent that embeds an entire security team—an analyst, a fixer, and a pen-tester—directly into your pull request workflow.

Inspiration

The rise of "vibe coding" and AI-assisted development highlighted a major security gap, but it also revealed a universal truth about modern software engineering: speed often comes at the expense of security. The pressure to ship quickly means that vulnerabilities can slip through, regardless of whether the code is written by a human, generated by an AI, or a mix of both.

We were inspired to build a universal safety net for this new era of development. We envisioned an intelligent agent that could act as a tireless security partner, scrutinizing every pull request with the same rigor. Our goal was to automate the tedious but essential task of secure coding, making robust security a seamless and effortless part of the development cycle, not a bottleneck. Sentoru was born from this vision—to secure *all* code, for *every* developer, in *every* PR.

What it does

Sentoru is an AI-powered security agent designed to ensure every line of code is secure, no matter how fast—or auto-generated—it is. In technical terms, Sentoru is a **Hierarchical Multi-Agent System** architected to be **git-provider agnostic**, integrating directly into the pull request workflow of modern development platforms like GitHub, GitLab, or Azure Repos.

Once triggered by a new pull request, Sentoru's orchestrated workflow automatically:

1. **Gathers contextual intelligence** through a Search Agent that retrieves relevant security knowledge from a custom RAG knowledge base.
2. **Analyzes** code changes for vulnerabilities using both the retrieved security context and direct vulnerability scanning.
3. **Generates fixes** and proposes them in a developer-friendly, native format.
4. **Generates penetration tests** (`pytest` code) to validate that the suggested fixes are effective against the original attack vector.

How we built it

Sentoru's journey began as an idea at a previous hackathon where we first explored Google's Agent Development Kit (ADK). Inspired, we envisioned an agent that could automatically secure modern code.

A significant part of our development involved mastering the Google Cloud platform, which was a new and incredibly rewarding experience. To bring our vision to life, we built Sentoru on a modern, serverless **Google Cloud stack**, leveraging these key technologies:

- **Hierarchical Multi-Agent Architecture:** We designed a sophisticated orchestration system where an **Orchestrator Agent** intelligently coordinates the entire security workflow. This master agent conditionally invokes a **Search Agent** for RAG-based knowledge retrieval, then delegates to a **Review Agent** that manages the sequential pipeline of specialized security agents (Analyst, Fixer, Pentester).
- **Vertex AI RAG Engine:** We configured a powerful RAG engine on Vertex AI. Our knowledge base, a corpus of security documents, is stored in **Google Cloud Storage** and ingested by Vertex AI Search. The system then uses **Gemini 2.5 Flash**, chosen for its optimal balance of performance and cost-efficiency, to intelligently parse the documents and `text-embedding-005` to create vectors for high-quality semantic search.
- **Agent-as-a-Tool Pattern:** We implemented a pattern where the Search Agent operates as a specialized tool within the orchestrator, allowing for modular, configurable intelligence gathering.
- **Cloud Run & Probot:** A serverless function runs our bot (built with Probot), which listens for PR webhooks from git providers to trigger the agent's security review **in a scalable, auto-scaling environment.**
- **GitHub Actions for CI/CD:** We established a full DevOps pipeline on GitHub to automate the deployment of our agent to Google Cloud. This effort paid off immensely, making our process repeatable, reliable, and ready for future iterations.

Challenges we ran into

- **Perfecting the Developer Experience:** Our biggest challenge wasn't just finding vulnerabilities, but presenting the fixes. We invested significant prompt engineering and a deep dive into the GitHub API to ensure our agent's code suggestions were delivered in the correct patch format, allowing developers to approve and commit them with a single click.
- **Orchestrating a Secure Event-Driven Architecture:** Building a robust, real-time system was a new experience. We navigated the complexities of securing webhooks, ensuring reliable event delivery to our Cloud Run service, and managing the stateful interaction with the Vertex AI Agent Engine, all while upholding high security standards.

Accomplishments that we're proud of

One of our standout achievements is creating a system that directly tackles a critical challenge in modern software development. In an era where AI-generated code is accelerating development, we're proud to have built an intelligent, automated solution that addresses the essential need for security.

Our key accomplishments include:

- **Developing a True Hierarchical Multi-Agent System:** We successfully designed and deployed a sophisticated orchestration system where an intelligent coordinator manages specialized agents across multiple tiers. From the **Orchestrator Agent** that makes dynamic decisions about tool invocation, to the **Search Agent** that provides contextual intelligence, to the sequential **Review Agent** pipeline (Analyst, Fixer, Pentester)—this multi-layered agentic architecture is not just a concept; it's a functioning solution to a real-world problem that demonstrates advanced AI coordination patterns.
- **Building a Seamless GitHub Bot Integration:** We are particularly proud of the orchestration between our agents and the pull request workflow. We developed a custom GitHub bot that brings the agent's insights directly to the developer, highlighting vulnerabilities and suggesting fixes. This bot is the crucial component that **closes the feedback loop** instantly and effectively.
- **Empowering Developers to Embrace AI Securely:** Ultimately, our biggest accomplishment is not just the technology itself, but what it enables. By automating the security review process, Sentoru empowers developers to innovate and leverage AI tools with confidence, knowing they have an intelligent partner watching their back.

How Sentoru Stands Out

Sentoru combines the strengths of multiple security tool categories into a single, open, and developer-centric agent.

Tool/Agent	Multi-Agent Architecture	Security Analysis (Static)	Suggests & Commits Fixes	Generates Penetration Tests	PR Integration (GitHub)	Explainability (Comments, Docs)	LLM-Generated Code Focus	Open Source & Customizable
Sentoru	✅	✅	✅	✅	✅	✅	✅	✅
Snyk Code + AI Fix	❌	✅	✅	❌	✅	✅	⚠️	❌
GitHub CodeQL	❌	✅	❌	❌	✅	✅	❌	✅
Argusee (Google Research)	✅	✅	❌	✅	❌	⚠️	❌	✅
Generic LLMs (ChatGPT, etc.)	❌	⚠️	✅	❌	❌	✅	✅	✅

What we learned

1. **LLMs are Force Multipliers for Research and Development:** We knew Gemini would be the core of our agent, but we were amazed by its utility as a development partner. We used it daily as a co-pilot to debug cloud configurations and write infrastructure code. More profoundly, we leveraged its Deep Research capability to create our custom Python security knowledge base. Since no synthesized guide existed, we tasked it to research and consolidate information from over 260 sources into comprehensive, up-to-date documents with real-world code examples, a feat that would have been manually impossible.
2. **Mastering a New Cloud Ecosystem from Scratch:** As professionals who don't use Google Cloud in our daily work, we embraced the challenge of learning a new, complex platform. This project taught us how to go from zero to a fully functional, secure, and serverless application on GCP. We're incredibly proud of this steep and successful learning curve, which proves our team's ability to rapidly adapt and build on any modern stack.
3. **The "Last Mile" of Developer UX is Everything:** An agent's intelligence is only valuable if it's usable. The effort to get the GitHub patch format perfect taught us that a seamless user experience is just as critical as the underlying AI. It doesn’t matter how smart the agent is if its suggestions are difficult to use.
4. **You Don't Need to Be a Domain Expert to Solve a Real Problem:** We are not cybersecurity professionals; this project was born from a need we identified as developers. It taught us that by combining a sharp focus on the user's problem with powerful tools and proactive consultation with knowledgeable colleagues, a passionate team can build a valuable solution even in a highly specialized domain.
5. **Security is Inherent Value, and Automation is How You Deliver It:** This project crystallized a core belief for us: security is not an optional feature, but an inherent component of quality software. While every good developer knows this, the process is often manual and time-consuming. We learned that the most profound value Sentoru offers is its ability to automatically embed this crucial layer of security into every pull request, making the *right way* the *easy way*.

What's next for Sentoru

Looking ahead, our vision is to evolve Sentoru from a powerful agent into an indispensable, autonomous security partner for any development team. Our roadmap is focused on strengthening its capabilities and thoughtfully expanding its reach:

- **Achieving True Autonomy with a "Closed-Loop" Agent:** Our top technical priority is to implement the final piece of our vision: a secure execution environment for the generated penetration tests. If a test fails, the system will loop back, re-engaging the Fixer Agent with the results. This creates a recursive, self-healing cycle that continues until the vulnerability is verifiably patched and the code is truly secure.
- **Becoming a Universal Tool for Developers Everywhere:** While we currently focus on Python, we see immense potential for Sentoru to become a versatile tool for all developers. We will expand support to other major languages like JavaScript, Java, and Go, ensuring that any team can benefit from automated security, regardless of their tech stack.
- **Enabling Trusted Enterprise Adoption:** We recognize that large corporations are rightly hesitant to share their intellectual property with external services. To meet them where they are, we envision Sentoru as a solution that can be deployed entirely within a company's own cloud environment. This makes Sentoru a trusted and easily adoptable piece of technology, allowing enterprises to leverage our agent with their own models and infrastructure, and embrace AI with both security and confidence.
- **Re-enabling Full RAG Capabilities in Cloud Deployments:** While our RAG-powered Search Agent works flawlessly in local development, we encountered a deployment limitation where `VertexAiRagRetrieval` tools return empty results when deployed to Vertex AI Reasoning Engine—a known issue in the current ADK version. We're actively monitoring ADK developments and will re-enable our full RAG capabilities in cloud deployments as soon as this issue is resolved, allowing our deployed agents to leverage the complete knowledge base for enhanced security analysis.
- **Fortifying for Enterprise-Grade Reliability:** To build the trust required for enterprise adoption, we will systematically review and harden all external service integrations. Our goal is to ensure that every component in our toolchain meets the stringent security, scalability, and reliability standards required for a mission-critical development tool.