cyberhaven

Inside a CISO's Playbook

What Agentic Security Looks Like *in Practice*

By Aman Sirohi, David Phillips, Grant Sowards and Brock Talbott

Cyberhaven's Office of the CISO is setting a higher standard for cybersecurity by embracing a new paradigm: autonomous, agent-driven security. Faced with an ever-widening gap between engineering velocity and security capacity, we made a deliberate shift from traditional, human-in-the-loop processes to a model where specialized AI agents act as force multipliers for our team.

We recognized that incremental improvements to legacy workflows would not be enough. Instead, we have invested in autonomous, agent-based systems that can reason across code, infrastructure, and workflows in real time. These agents don't just assist, they reason, collaborate and execute, reducing manual overhead and embedding security seamlessly into every stage of our development lifecycle. Through agentic workflows, we are transforming security from a reactive bottleneck into a proactive, scalable, and resilient orchestration layer, empowering practitioners to meet today's cyber threats with unprecedented agility.

The 200:1 Reality

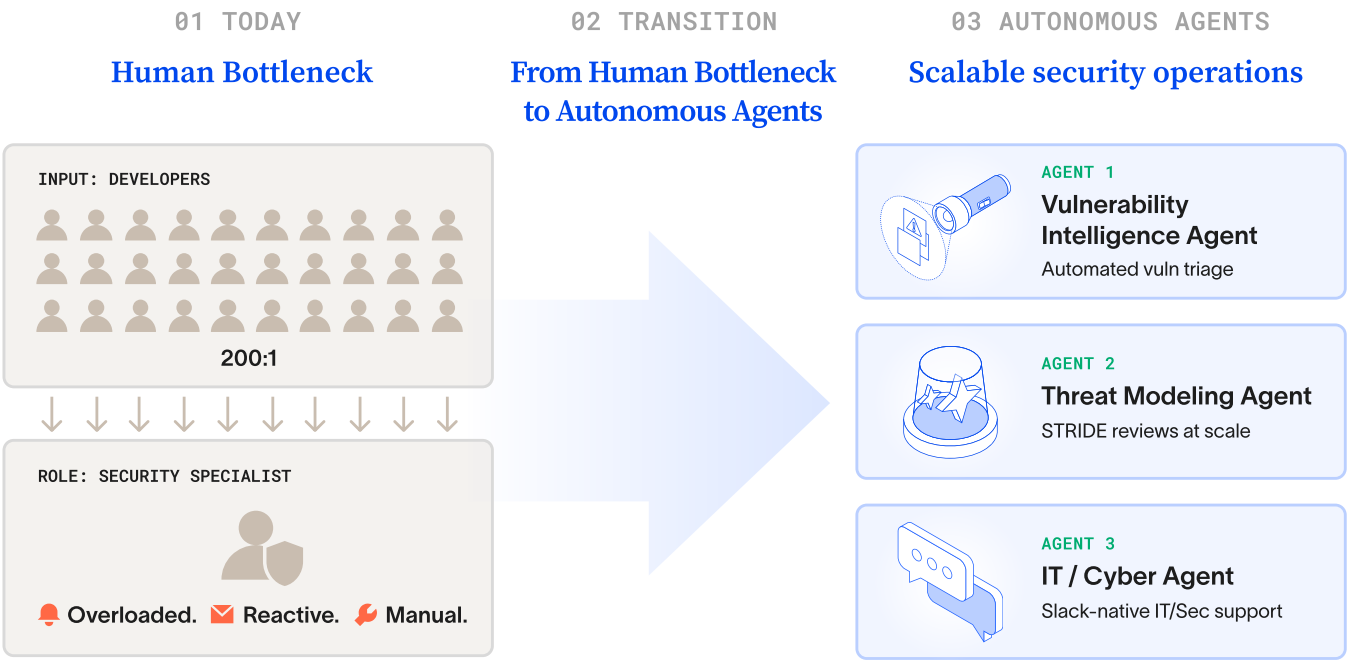
In the modern enterprise, security teams are struggling to keep pace with the sheer volume of security alerts, vulnerabilities, and infrastructure changes demanding review.

As software development velocity accelerates, the gap between those building and those securing has widened to a breaking point. Industry data suggests a staggering 200:1 ratio of engineers to AppSec professionals.

Traditional security workflows are fundamentally linear: a vulnerability is found, an analyst triages it, a developer is notified, and a fix is deployed. This manual "triage-tax" consumes up to 40% of a security engineer's week, leaving little room for proactive threat hunting or architectural hardening.

At Cyberhaven, we decided that the only way to scale security was to stop acting like a bottleneck and start acting like an orchestrator. We have moved from "AI-assisted" tools to **Autonomous Security Agents**. These aren't just chatbots, they are specialized digital teammates that reason, collaborate, and execute complex security workflows.

The Evolution of Security Operations



Section 01

Cerberus: The Vulnerability Intelligence Ensemble

Lead: David Phillips **Agent:** Cerberus

Is / Does / Means

Is: A multi-model ensemble system that automates the entire vulnerability triaging pipeline.

Does: Fetches findings from cloud and code scanners, performs reachability analysis by inspecting our actual codebase, and conducts a "cross-examination" between different LLMs to reach a verdict.

Means: Our security engineers no longer spend mornings manually dismissing "noise." They only see findings that have been verified as exploitable within our specific environment.

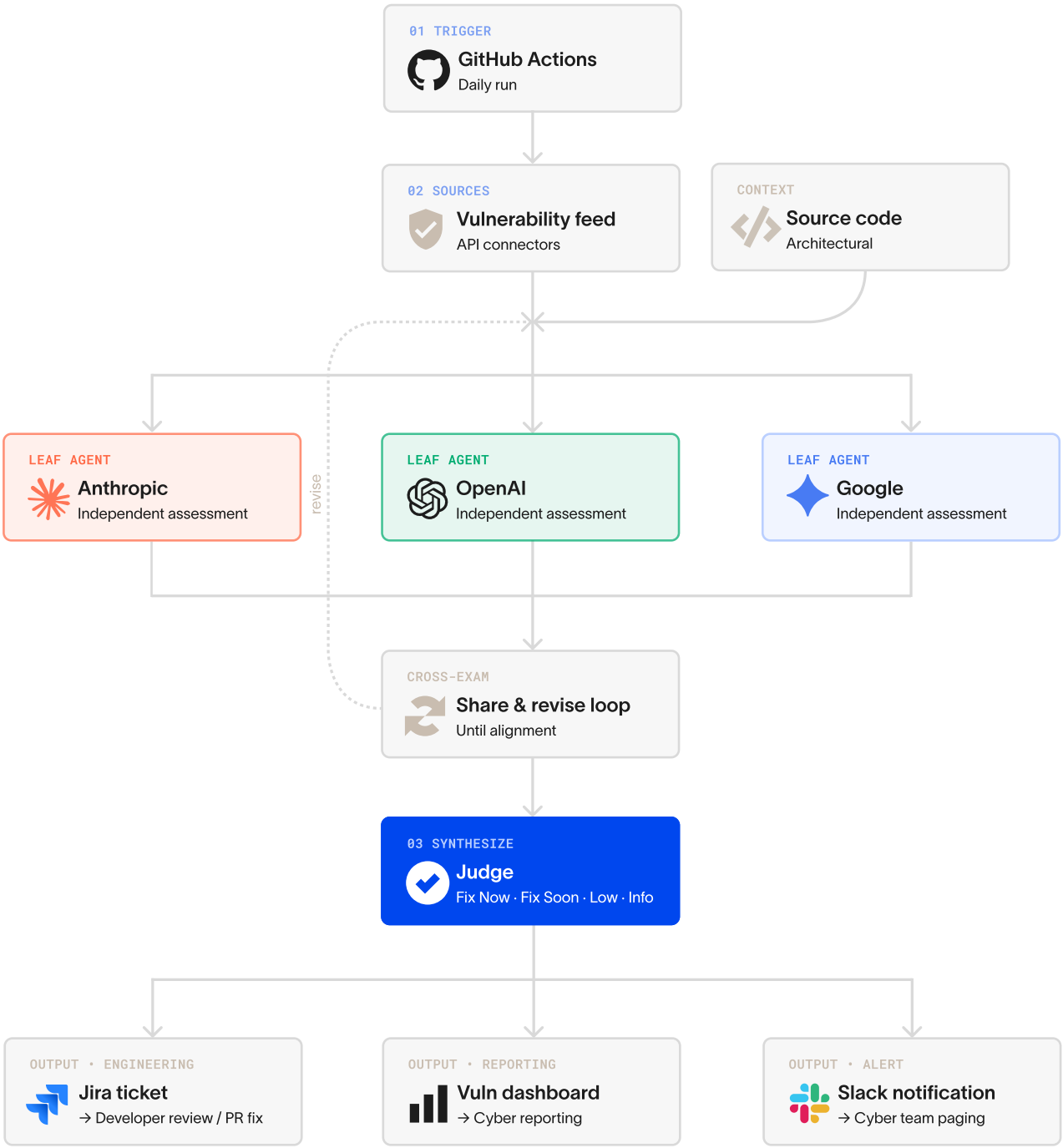
Technical Deep Dive & Architecture

Cerberus is built on the principle of **Adversarial Collaboration**. We realized that no single LLM is perfect. One might be better at understanding Python logic, while another excels at cloud infrastructure context.

The Implementation:

1. **Ingestion & Deduplication:** Triggered daily via GitHub Actions, Cerberus pulls raw findings from our security stack.
2. **Context:** Relevant source code and infrastructure repos are cloned and provided to the leaf agents.
3. **The Ensemble Run:** The system spins up three "Leaf Agents" in parallel from Anthropic, OpenAI, and Google. Each agent has a unique system prompt, optimized based on our test results.
4. **Cross-Examination (The "Aha!" Moment):** For each finding, leaf agents share their evidence and judgement with each other. The agents get an opportunity to re-analyze the finding based on each other's independent analysis. If one agent missed a key function, it has a chance to review and correct its initial finding. This helps balance out any weaknesses in specific models.
5. **The Judge (Claude Opus):** A final, high-reasoning model reviews the debate and issues a final verdict (e.g., "Critical — Reachable via Public API").
6. **Visibility:** Vulnerability judgements are posted to a live dashboard for SLA tracking, Jira ticket creation, and fix suggestions. This provides a single pane of glass to review vulnerabilities across multiple tools.

Architecture Diagram · Agent: Cerberus



Why these components?

We chose a multi-vendor stack specifically to combat **model-specific hallucinations**. By forcing a consensus between three distinct training sets, we achieved a **85% reduction in false positives** compared to raw scanner output.

Section 02

The Threat Modeling Agent: Architectural Review at Scale

Lead: Grant Sowards **Agent:** Vektr

Is / Does / Means

Is: A design-review agent integrated into our RFC (Request for Comments) process.

Does: Automatically analyzes architectural documents in Notion, applies the STRIDE methodology, and appends a security review to the design.

Means: We provide consistent and thorough threat models at a pace that matches engineering.

Technical Deep Dive & Architecture

With a 200:1 ratio of engineers to AppSec engineers, traditional threat modeling simply can't keep pace with the volume of new services, features, and architectural changes. The result is either a bottleneck that slows engineering velocity or, more often, threat models that get skipped entirely. Our agent eliminates these challenges by delivering consistent, well-tuned threat analysis on designs in minutes, giving architects near-immediate feedback on the threats their systems will face.

How it's Implemented:

Document Store: Our agent is integrated into Notion and detects RFCs when they are ready for review.

Divide & Conquer Pipeline: We found that giving an LLM a large architecture document leads to "context thinning" that hurts results. Instead, we use a pipeline of specialized sub-agents across four phases.

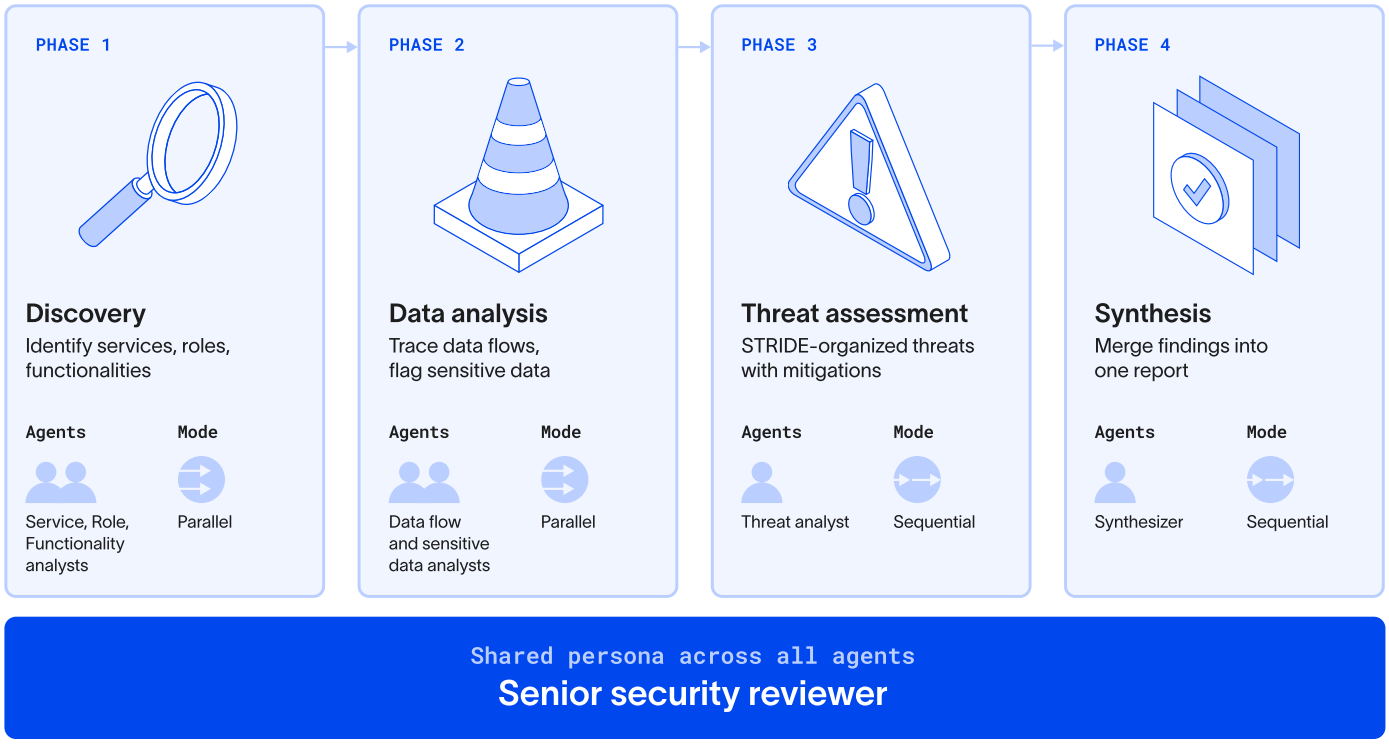
- **Phase 1 Asset Discovery (Parallel):** Identify services, roles, functionalities, and other assets
- **Phase 2 Data Analysis (Parallel):** Map sensitive data and dataflows
- **Phase 3 Threat Assessment:** Perform threat assessment following STRIDE methodology
- **Phase 4 Synthesis:** Compiles final report

Methodology Enforcement: The agent is hard-coded to follow the STRIDE framework (Spoofing, Tampering, Repudiation, Information Disclosure, DoS, and Elevation of Privilege) enumerating each asset and identifying threats based on this taxonomy.

RAG (Retrieval-Augmented Generation): The agent searches our internal document stores for additional context related to the design.

Architecture Diagram · Agent: Vektr

Divide-and-conquer pipeline with 4 phases



Section 03

Jarvis: The IT Support Frontline

Lead: Brock Talbott **Agent:** Jarvis

Is / Does / Means

Is: A Slack-native agent that handles IT helpdesk inquiries, security questions, and takes direct action in third-party systems.

Does: Automates access requests , analyzes suspicious phishing screenshots, answers questions about Cyberhaven's policies, surfaces similar past Q&A from Slack history, and automatically creates tickets for unresolved issues.

Means: Our IT/Security team is "always on." Employees in Europe and India get instant resolutions while the US team is offline.

Technical Deep Dive & Architecture

Jarvis is an **action-oriented agent**. It doesn't just provide answers, it takes steps in third-party systems.

The Implementation

Slack Bolt Framework

Built using Slack's native API to allow for "App Mentions," threaded replies, interactive buttons, and ephemeral messaging for sensitive data like password resets This agent will also query internal Slack channels for similar past questions and referencing them in its response.

Okta Integration

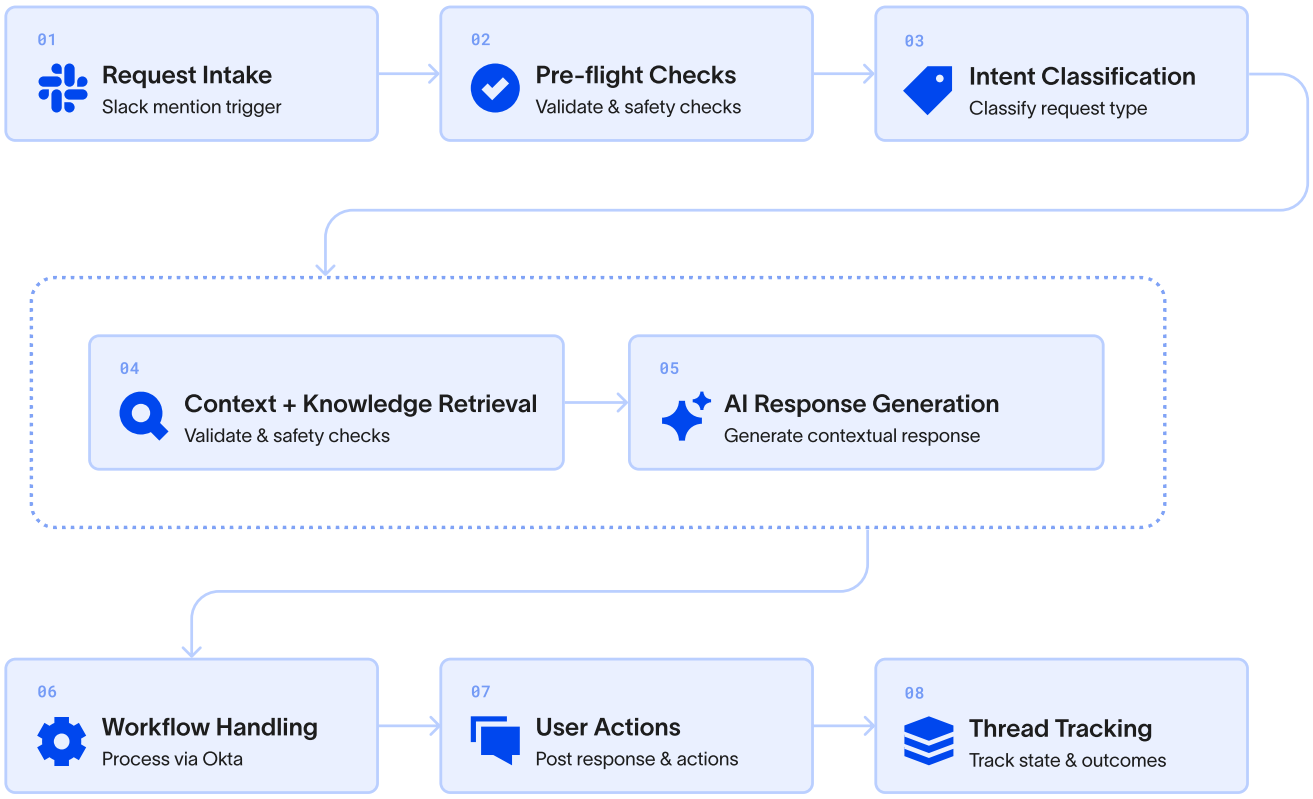
If an employee asks, "Can I have access to Figma?", CyberBot doesn't just say "Go to the portal." It queries the Okta API, checks if the app is in the requestable catalog, fetches required approval questions per-app, prompts the user in-thread, and submits the request on behalf of the end user. Access requests flow through existing approval sequences with no workflow changes.

Vision-Based Phishing Analysis

Using **Claude**, the bot can "see" screenshots of emails, Okta errors, and UI issues. It looks for visual red flags—mismatched sender domains, generic greetings, and suspicious URL structures—and guides the user with actionable next steps.

Architecture Diagram · Agent: Jarvis

Main message handler flow



The Path Forward: Partnerships and Collaboration

The Cyberhaven security team isn't building these agents in a vacuum. We believe that the future of security belongs to organizations that can share and refine these agentic patterns.

We are currently tracking a **significant ROI** across these projects:

- **Arithmos:** 70% reduction in manual triage time.
- **Threat Modeling:** 100% coverage of architectural designs.
- **CyberBot:** Over 70% of routine IT tickets resolved autonomously.

We want to lead the conversation with our customers and peers. Are you experimenting with ensemble agents? How are you handling trust boundaries in autonomous workflows?

Let's build a stronger cyber community — one that thrives on collaboration, shared knowledge, and innovation. Connect with the Cyberhaven security team to explore how, together, we can advance toward a more secure, autonomous future across the cyber domain.

Request a demo

About Cyberhaven

Cyberhaven is the AI-powered data security company revolutionizing how companies detect and stop the most critical insider threats to their most important data. Until now, data security products were limited to scanning data content and looking for specific user actions. Our AI technology analyzes billions of workflows to understand every piece of data within an organization, when it's at risk, and takes action to protect it. It's like nothing that's come before and protects data like nothing else. For more information, visit cyberhaven.com.