

Cisco DefenseClaw is an open-source security, governance, and observability layer designed to protect autonomous AI agents and runtimes like OpenClaw. DefenseClaw is a security layer for AI agents. It scans what an agent can use, watches what it tries to do, blocks unsafe actions by policy, and creates an audit trail.

Capabilities:

Skills Scanner: Every AI agent relies on “skills” capabilities like web search, CRM access, or email. The Skills Scanner reviews skills and maps findings to policy actions such as blocking installation, disabling a skill, or quarantining it. OpenShell is a separate feature that can isolate agent execution.

MCP Scanner: Model Context Protocol (MCP) evaluates tool descriptions and schemas. It can also scan prompts, resources, and instructions when configured. Policy can block installation or disable runtime, and configured servers can be rescanned on a schedule.

A2A Scanner: As AI deployments mature, agents increasingly interact with other agents (agent-to-agent communication). This introduces a malicious or compromised sub-agent that can corrupt the behavior of an orchestrating agent. The A2A Scanner (a separate opensource tool) can monitor and validate these inter-agent communications, ensuring that no unauthorized agent is injecting instructions or data into a trusted agent’s workflow.

CodeGuard: AI agents can generate code on the fly, creating a major security risk. CodeGuard analyzes agent-generated code before execution, enforcing security policies and flagging threats like unchecked file writes, open network sockets, and privilege escalation.

AI Bill of Materials (AI BoM): Just as a software bill of materials (SBOM) inventories software dependencies, the AI BoM automatically inventories every AI asset—including agents, skills, MCP servers, and models. DefenseClaw continuously updates this inventory as agents evolve, giving security teams a live, auditable view of their agentic workforce.

Features (layers):

Admission control: scans skills, MCP servers, plugins, and generated code before they run.

Runtime inspection: checks prompts, completions, OpenAI-format tool calls, and selected tool results in the execution path.

Policy enforcement: uses Rego/OPA-style policy and rule packs so findings can become allow, block, quarantine, or audit decisions.

Agent inventory: AIBOM can inventory skills, plugins, MCP servers, agents, tools, models, and memory from a live OpenClaw environment.

Sandbox and egress control: integrates with NVIDIA OpenShell for isolated execution and narrower permissions where supported.

Observability: emits durable audit evidence through SQLite, JSONL, OTLP, Splunk HEC, webhooks, and operator-facing views.

FAQs:

Q: What is MCP?

A: Model Context Protocol (MCP) is an open standard for connecting AI agents to external tools, data, and services. MCP servers act as bridges to systems like CRMs, file systems, and email, making them a critical security surface that DefenseClaw targets with its MCP Scanner.

Q: What is NVIDIA [OpenShell](#)?

A: NVIDIA OpenShell is a sandboxed runtime for AI agents that isolates skills, plugins, and tools from the host system. DefenseClaw integrates with OpenShell to enforce security policies at runtime.

Q: Is DefenseClaw truly free?

A: Yes. DefenseClaw is an open-source project being published on GitHub, free to use, modify, and contribute to. Cisco’s commercial revenue opportunity lies in the surrounding offerings (e.g. Duo IAM, Cisco Secure Access, AI Defense, and Splunk) rather than in DefenseClaw itself.

Skill-scanner

- LLM-assisted semantic analysis
- Cisco AI Defense inspection workflows
- VirusTotal analysis

MCP Scanner

- Evaluate MCP tools, prompts and resources
- Signature-based detection
- Security evaluation by AI Defense

Runtime Guardrails

- Pattern matching signatures
- Semantic interpretation (LLM)
- Behavioral analysis

CodeGuard

- Best practices & guidance (OWASP, CWE, ...)
- Rule translator for common AI coding agents
- Prevent introduction of vulnerabilities

DefenseClaw Capabilities

AI Agent Discovery & BOM

- Asset discovery
- Categorization of code constructs
- Asset dependency graph

Resources:

[OpenClaw Learning Lab](#)

[NVIDIA OpenShell](#)

[Github](#)

[Demo](#)

[OpenClaw](#)

[Blog](#)