ANTHROP\C

Hot off the Presses

AI Agents as Your Organization's Personal Security Newsroom

21 August 2025

Table of contents

Who Am I?

- Tech lead for Incidents, Intelligence, and Investigation (i3) within Anthropic's security team
- 15+ years experience in cybersecurity
    - Experience across the technology space, from startups to cloud providers
- Lots of different flavors of security experience, lending a comprehensive view of threat landscape
    - Red team and blue team and purple team!

What is Anthropic?

- Safety-focused AI research lab
- Started as a US Public Benefit Corporation
- Responsible Scaling Policy
- Developer of Claude

What is Threat Intelligence, Anyways?

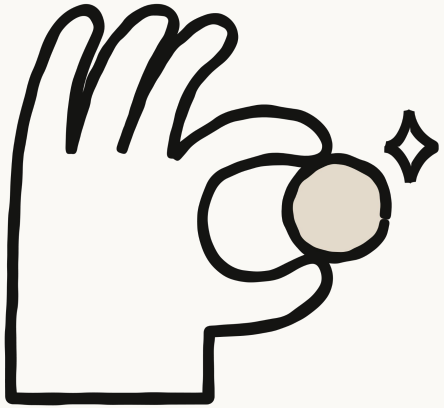
Threat intelligence is a method of optimizing your organization's defenses.

Rather than forcing cybersecurity to be purely reactive against every possible threat, the field of threat intelligence enables defenders to more efficiently focus their attention on the most likely actors and their preferred tactics, techniques, and procedures that would affect them.

CTI Industry Problems

- Cybersecurity market is opaque, challenging for customers to evaluate offerings
- One size fits all - not a good fit for intelligence cycle
- Marketing priority - misaligned incentives for improvement

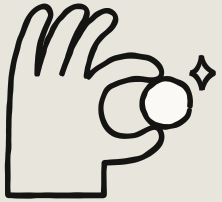
CTI Customer Problems



- Anthropic-specific: Unusual threat landscape
  - Cloud native
  - OS distribution

- Generic: Regional focus is almost exclusively US/Western European customers

So What Can We Do?

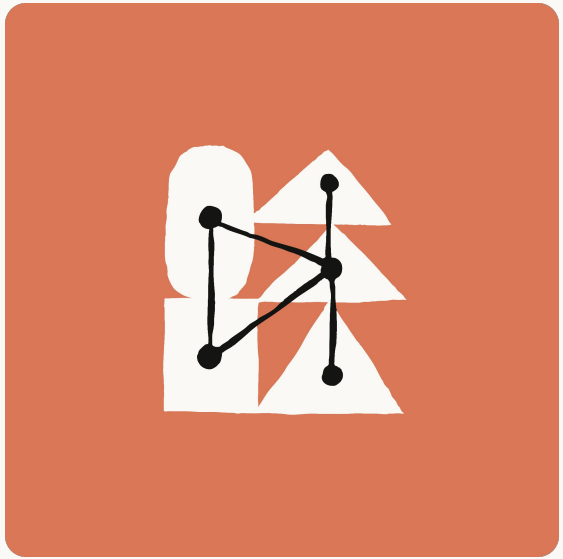


Use the tool(s) at hand!

Inspired by a blog post by Brandon Dixon, got the idea to build a

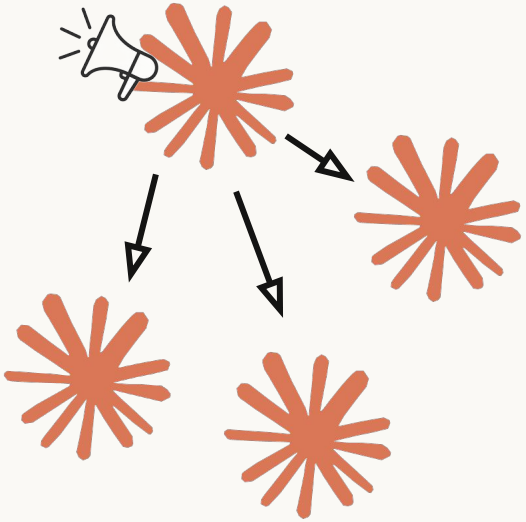
threat intelligence digest

Agents 101



- What is an AI agent? Let's ask one!
- Claude says: **"Autonomous software that perceives, decides, and acts to achieve goals"**
  - Perception: Takes in information (text, data, environment state)
  - Reasoning: Processes information and plans actions
  - Action: Executes tasks through tools, APIs, or outputs
  - Memory: Maintains context across interactions

Agents 202

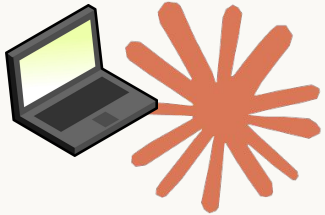


- Agents can be recursive - Claude calling Claude(s)
- One possible Agent architecture: orchestrator calling subagents for specific tasks
  - Each subagent can have bespoke system prompt instructions
  - Ex: architect/designer => coder => code reviewer => security review => deployer
  - Enables parallelization and can help with context window size constraints

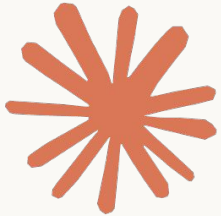
Our Agentic Project

Oops! All Claudes!

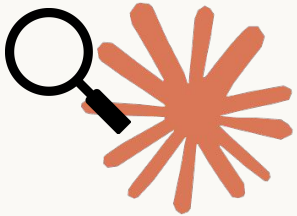
Analyst Claude



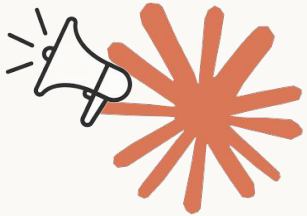
Developer Claude



Extra Claude (never know when you might need one more)



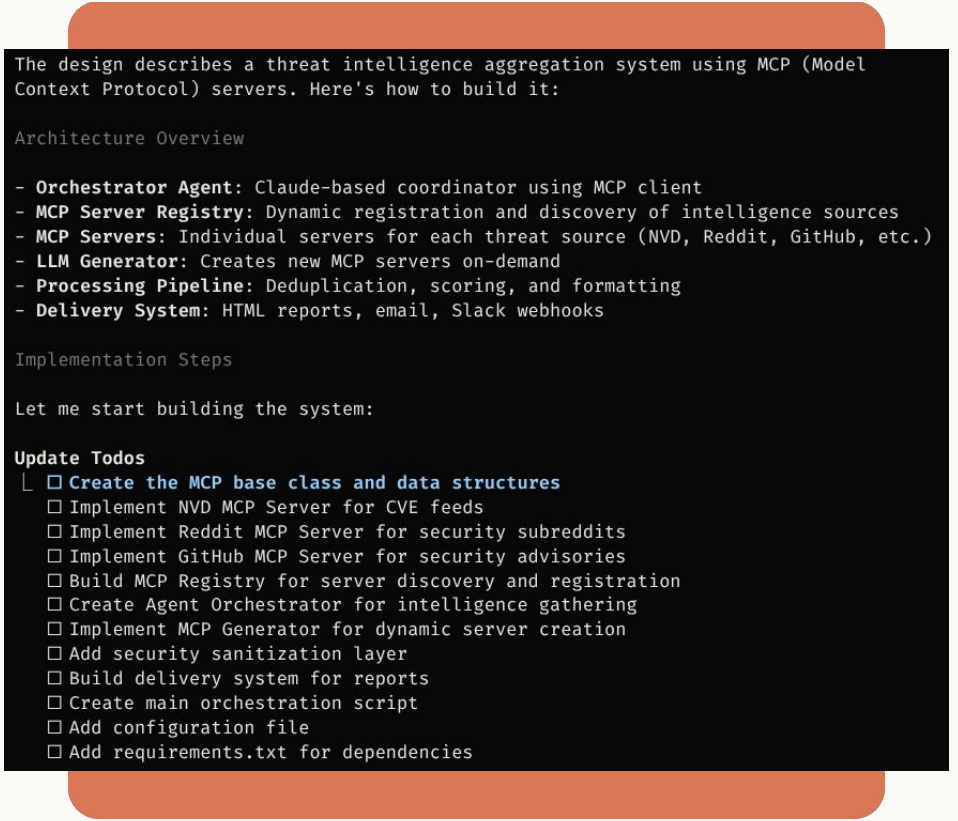
Investigator Claude



Supervisor Claude

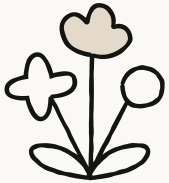
Design Documentation with Claude

- Let's have Claude draft a design doc
- Takes 3 tries - too much, too little, finally just right
- Design doc can then be fed into Claude Code and it'll build a TODO list and go start building it!



What is MCP, Exactly?

The Model Context Protocol (MCP) is an open protocol developed and shared by Anthropic that lets AI assistants connect to any data source or tool through standardized interfaces

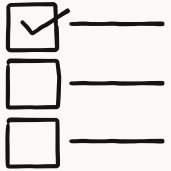


Open Source

Contributed to the AI community to improve all users' ability to develop novel AI capabilities

Standardized

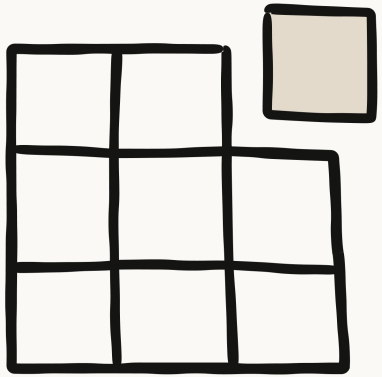
Exposes consistent interface for LLMs to leverage external tools



Extensible

Enables rapid scaling of an AI implementation to numerous independent tools

Building Block – MCP Server for Input Sources

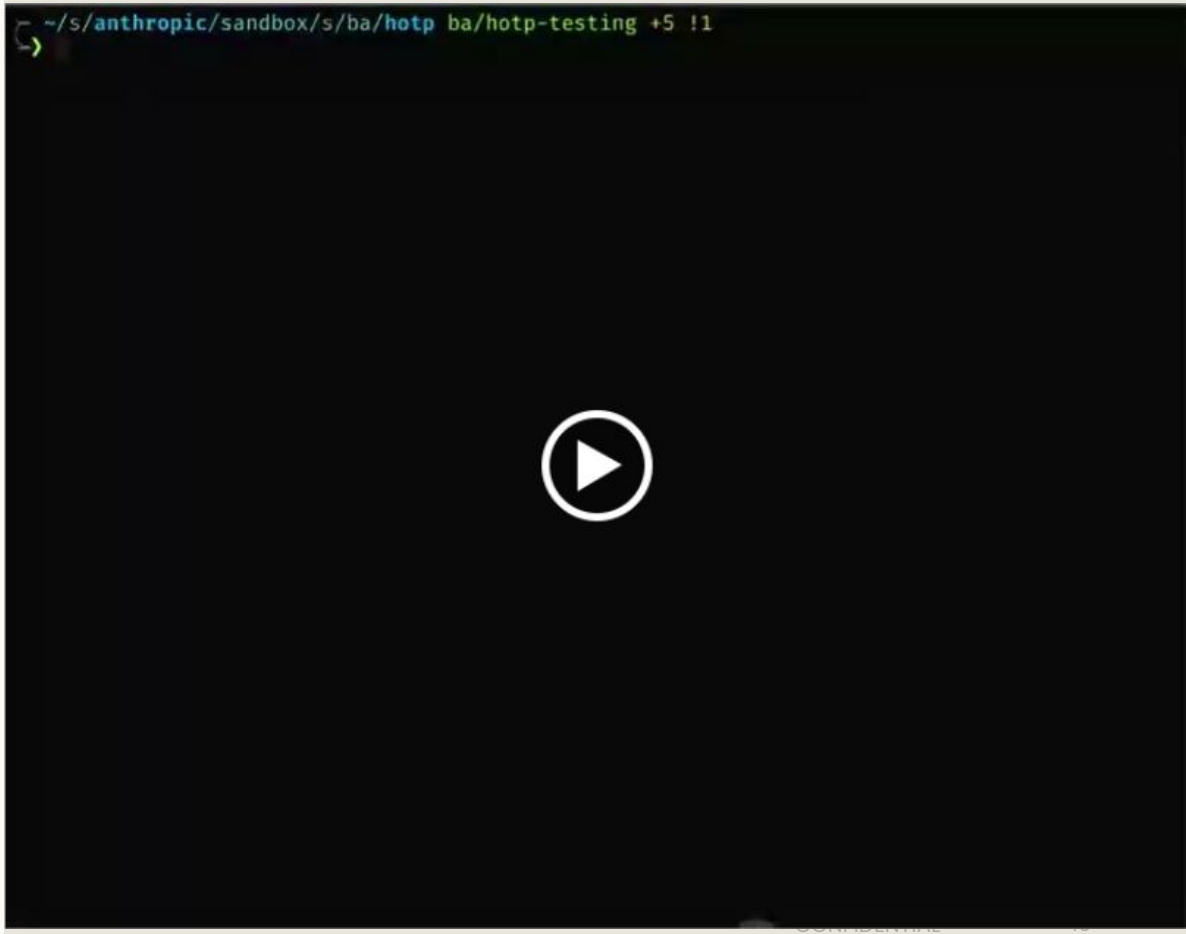


- First step is build a single POC
  - Take a given intelligence source, craft an MCP server
- Claude selects a simple public feed: NIST NVD
- Mocks up an MCP server implementation
  - Zero shot fail, Claude builds a fake MCP server that doesn't actually run MCP :(
  - Some prompting gets it to actually build a functioning MCP server
- To make this extensible, Claude builds a template base class to use as foundation

Demo – NVD MCP Server Implementation

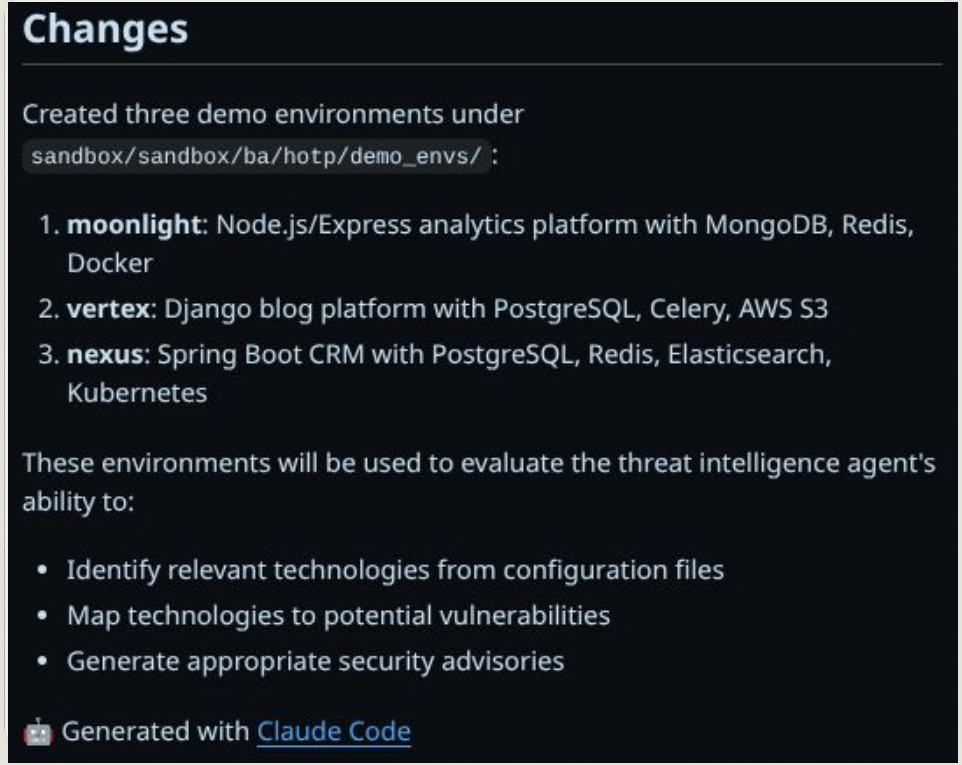
Once Claude has built the MCP server, we can install it into Claude Code to test it!

Next up we'll scale this to multiple servers



Mock Tech Stacks

In order to have our agent evaluate a series of tech stacks, we ask a different Claude to help us build some:



Changes

Created three demo environments under `sandbox/sandbox/ba/hotp/demo_envs/`:

1. **moonlight**: Node.js/Express analytics platform with MongoDB, Redis, Docker
2. **vertex**: Django blog platform with PostgreSQL, Celery, AWS S3
3. **nexus**: Spring Boot CRM with PostgreSQL, Redis, Elasticsearch, Kubernetes

These environments will be used to evaluate the threat intelligence agent's ability to:

- Identify relevant technologies from configuration files
- Map technologies to potential vulnerabilities
- Generate appropriate security advisories

🤖 Generated with Claude Code

Different Threats for Different Stacks

```
🔥 Hot Off the Presses - Custom Tech Stack Demo
==================================================

📂 Scanning repository: demo_envs/nexus
🔍 Creating orchestrator from repository: demo_envs/nexus
🔍 Scanning repository: demo_envs/nexus
📊 Detected 12 technologies
🎯 Generated threat configuration:
   Tech stack: Java Spring Boot, PostgreSQL Database, Redis Cache, Elasticsearch
Search Engine, Maven Build System...
   Critical assets: Customer Data in PostgreSQL, Authentication Tokens (JWT), Redis
 Session Cache...
   Threat actors: Nation State APTs, Cybercrime Groups, Insider Threats...
✅ Configuration saved to: detected_tech_stack.yaml
✅ LLM analysis enabled (API key found)

💾 Configuration saved to: detected_tech_stack.yaml
🎯 You can now run with: python run_with_custom_config.py config
detected_tech_stack.yaml
```

Orchestration Loop - The Agentic Core

Claude is running code to generate new prompts based on data, enabling the perception -> orientation -> execution loop

```
        analysis_prompt = f"""
You are a threat intelligence analyst. Analyze these security items
 for relevance to our environment.

Environment Configuration:
- Tech Stack: {json.dumps(self.config.tech_stack)}
- Critical Assets: {json.dumps(self.config.critical_assets)}
- Threat Actors of Interest: {json.dumps(self.config.threat_actors)}
- Severity Threshold: {self.config.severity_threshold}

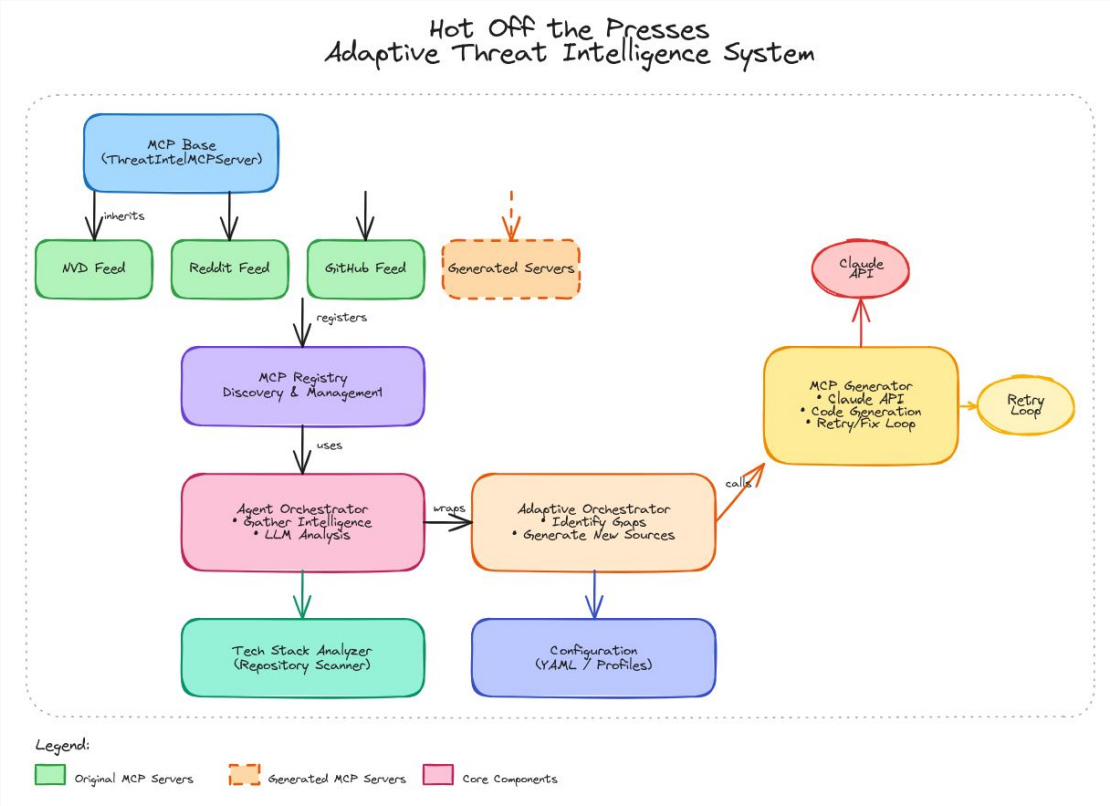
Intelligence Items:
{json.dumps(items_to_analyze, indent=2)}

For each item, provide:
1. Relevance score (0-10) based on our tech stack and assets
2. Brief analysis explaining the relevance (or lack thereof)
3. Recommended actions if score >= 7
4. Related threat actors if applicable
5. Confidence level (0.0-1.0) in your assessment

Return as JSON array with these fields for each item:
- original_title: (copy from input)
- relevance_score: (integer 0-10)
- analysis: (string, 1-2 sentences)
- recommended_actions: (array of strings, empty if score < 7)
- threat_actors: (array of strings, empty if none identified)
- confidence: (float 0.0-1.0)

Focus on practical relevance to the configured tech stack and critical assets.
"""
```

Claude's Eye View



Generatively Addressing Gaps

Claude looks at the tech stack, checks the registry, sees if there's tech not well represented, and generates a new MCP server to cover the gap in intel

```
        prompt = f"""The generated MCP server code failed with this error:

Error: {error_msg}

Here's the code that failed:

```python
{code}
```

This is attempt {attempt} of 3. Please fix the code so it works correctly.

Remember the ThreatIntelItem dataclass expects these exact fields:
- title: str
- description: str
- source: str
- url: str
- severity: str
- published: str
- tags: list[str]

Common issues:
1. Wrong field names (like 'id', 'source_url', 'timestamp' instead of
correct names)
2. Wrong data types (published should be str, not datetime)
3. Missing required fields
4. Using attributes that don't exist on feed entries

Return ONLY the fixed Python code (or "CANNOT_FIX" if unfixable):
"""
```

Building It Securely

Using LLMs can save us time and resources, but does it introduce novel security concerns?

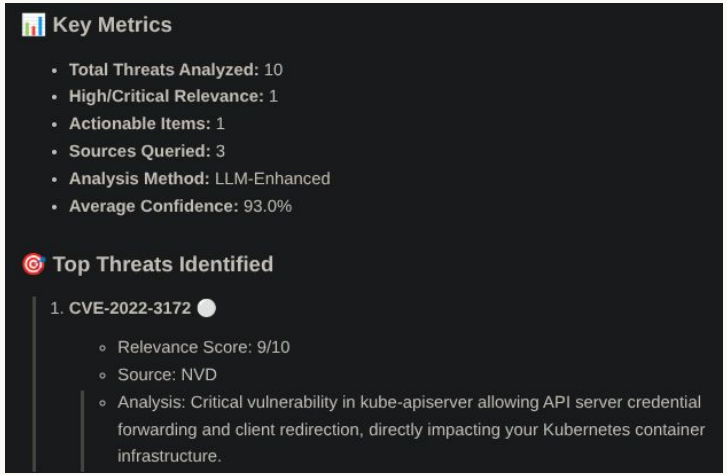
Prompt Injection

We're taking in a lot of random text and parsing it with LLMs, which is a recipe for prompt injection

Future Risks + Controls

If we expand this, risks might be more concerning. But could add classifier filters over the input corpus, and MCP server/sub-agent architecture might limit blast radius in some circumstances

Reporting - The Deliverable Newsletter



📊 Key Metrics

- **Total Threats Analyzed:** 10
- **High/Critical Relevance:** 1
- **Actionable Items:** 1
- **Sources Queried:** 3
- **Analysis Method:** LLM-Enhanced
- **Average Confidence:** 93.0%

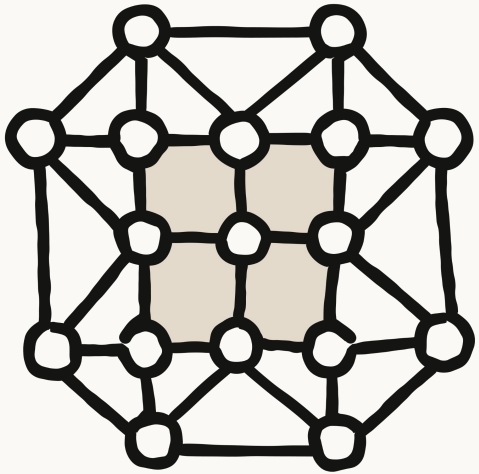
🎯 Top Threats Identified

1. **CVE-2022-3172** ⚪
   - Relevance Score: 9/10
   - Source: NVD
   - Analysis: Critical vulnerability in kube-apiserver allowing API server credential forwarding and client redirection, directly impacting your Kubernetes container infrastructure.

- Markdown report in current version for ease/flexibility
- Could be expanded in the future easily into more flexible output (daily Slack updates via bot, etc)
  - Could make something more interesting for exec consumption like a podcast using TTS!
  - Also would be interesting to tailor reporting to different departments
    - Focus on app/library/supply chain issues for appsec
    - Focus on infra layer issues for networking or infra team

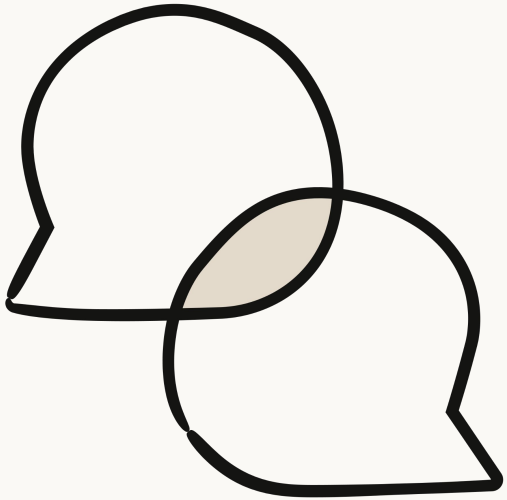
So now what?
Let’s look into the future

Expanded Feature Set – Less Structured Sources



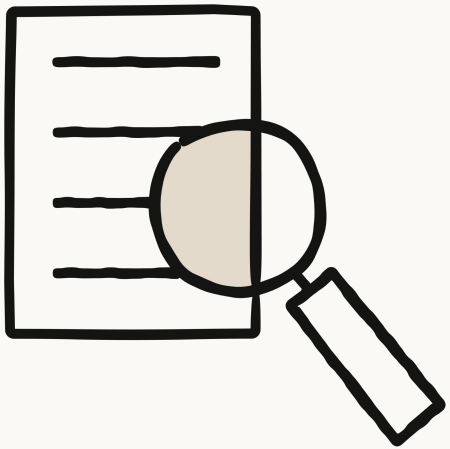
- Time and resource constraints for now kept this limited to more structured feeds focused on vulnerability intelligence
- Would be great to expand this out
  - More detailed articulation of threat actor profiles (currently all APT groups treated equally, etc)
  - Incorporation of less overt sources where LLM interpretation could provide differentiator

Expanded Feature Set - Multimodal Sources



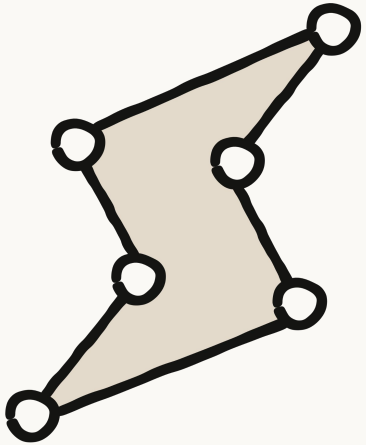
- Current sources are all structured/unstructured text
- Could expand to:
  - Audio - Podcasts via automated transcription
  - Video - Recorded con talks (like this! meta!), youtube videos, etc
    - Can pull relevant intel into text via tools like OCR of frames, etc

Expanded Feature Set - Virtual Persona Development



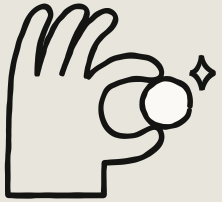
- Threat intelligence can involve scraping or monitoring of hacking group communications platforms
    - Forums, IRC, Discord, Telegram
- Could expand sources by enabling an agent-based network of subagents with tailored personae

Expanded Feature Set – Output MCP Integrations



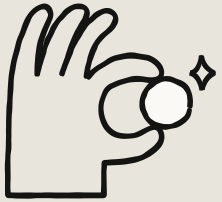
- Our MCP architecture is currently designed to standardize inputs and provide a lovely daily/weekly/etc journal
- Could add MCP standardization to the output side as well
- Enables integration with internal defensive ecosystem
    - Vuln management
    - SOAR

F.A.Q.



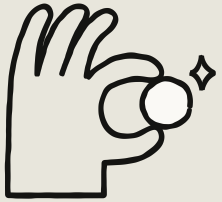
Will this be released?

Not in the current rudimentary form. That said the architecture from this presentation is not challenging to replicate for yourself with the help of Claude!



What alternative solutions or methods were considered?

In our case, the main alternative was “do nothing”



How does the system handle non-English sources?

So far so good, but you can check out Anthropic's official docs page for more granular info about Claude's multilingual support

Thank you

ANTHROP\C

ANTHROP\C