

Building a Private GenAI Team with Open-Source Models

ISSA Phoenix Q4

Grand Canyon Education | Grand Canyon University | Arizona Cyber Threat Response Alliance

```
christian@laptop:~ $ whoami  
christian
```

```
christian@laptop:~ $ groups $USER  
christian : gce it_security engineer
```

Leaderboard | 259

Q Search

Our Own Leaderboard

RK	MODEL	RATING	WON	LOST
1	qwen2.5:14b	1085	12	4
2	qwen2-math:7b	1027	4	2
3	Google: Gemma 2 9B (free)	1016	1	0
4	llama3.1:8b	1016	1	0
5	Meta: Llama 3.1 70B Instruct (free)	1016	1	0
6	AI Sonny the SOC Analyst	1011	1	0
7	phi3:14b	1011	1	0
8	granite3-dense:8b	997	5	5

Models | 6

Q Search Models

**AI Sonny the SOC Analyst**

Sonny, the SOC Analyst helps junior SOC analysts triage alerts and...

By Christian

**AI Tim the Threat Intel Analyst**

A Threat Intelligence Analyst tasked with performing analysis on threat...

By Christian

**AI Logan the Log Analyst**

Log Analysis capable of working on larger raw logs. Images of logs in Po...

By Christian

**AI Casey the Compliance Analyst**

Casey is a Compliance, Policy, and Governance specialist.

By Christian

**AI Madison the Cybersecurity Manager**

A cybersecurity manager responsible for cybersecurity strategy and...

By Christian

**AI Stacey the Scripter**

PowerShell, Bash, and Python Script writing assistant.

By Christian



Import Models

Export Models

Agenda

Intro What and Why?

Function

Practical Application

Models

Selection and Prep.

Challenges

Challenges and Risk

Tooling

Tools and Technologies

Frontier

Reliability -> Agency

Agenda

Function

Practical Application

- Data Manipulation
- Intelligence Extraction
- User or Machine Consumption
- Documentation
- Coding

Models

Selection and Prep.

- Model Size Models
- World Knowledge
- RAG vs. Fine Tuning
- AI Profiles (old-Agents)

Intro

What and Why?

- LLMs
- Privacy
- Open-Weight Models
- LeaderBoard
- FOSS Stack

Challenges

Challenges and Risk

- Use Case Issue
- Model Weakness
- Knowledge & Context Limits
- Model Security
- Reliability of Probability

Tooling

Tools and Technologies

- vllm | Ollama | OpenWebUI
- Security | Hardware
- Coding | API | Knowledge Management

Frontier

> Reliability == > Agency

- LLM Agency
- RAG Storage Optimization
- Model Size Efficiency
- Hardware Scale

01

Introduction

Large Language Model

What

AI built on transformer neural network.

Can generate human-like text

Uses statistical patterns, context, and semantic relationships of language and reason through training data

How

Breaks text into discrete tokens

References weights representing relationship between tokens including context

Returns statistically probable results using learned parameters

Significance

This AI domain, like others, can find and leverage relationships on its own

Can combine learned knowledge with user input

Offers highly dynamic solutions reducing development effort

Privacy - Data Sovereignty

Frontier Service

OpenAI | Mistral | Anthropic | Google | Meta | Grok

- Third Party Privacy Considerations
- Logging Considerations
- Regulatory Implications
- Vulnerability Exploitation

Cloud Hosted IaaS

AWS | Lambda | CoreWeave | Azure | DigitalOcean

- Apply Company Controls
- Control Hardening
- Self Contained Private VPC
- Scalable

Cloud SAAS

Bedrock | Hugging Face | Azure AI | GCP AI

- AWS VPC PrivateLink + VPC Endpoint
- Azure Conditional Access
- Hugging Face GDPR + SOC2

CloudFlare | CoreWeave | Lambda | MosaicML

Self Hosted

Enterprise System | Old Gaming Desktop | Combo

- Entry costs can be low but can climb fast
- Fullest Data Sovereignty
- All current controls can apply
- Some workloads can run on workstations

Open Weight Models

Specialist Models: Domain

Multimodal

Pick Your Size

Various Licenses

Community Driven

Free ;)

Open Weight Models: Gorillas

Llama (Meta)

Gemma (Google)

Phi (Microsoft)

Qwen (Alibaba)

Yi (01.AI)

Granite (IBM)

Open Weight Models: Community

DeepSeek (DeepSeek)

Nemotron (Nvidia)

Command-R (Cohere)

Reflection (Meta Community)

Llava (Meta Community)

Starcoder2 (ServiceNow)

Visualize these datss. Provide a full simple web app to present the data. There data includes a model name and then a score from 0 - 100.

Gemini 1.5 Pro - 80

Claude 3.5 Sonnet - 80

GPT-4o - 77

Claude 3.5 Sonnet - 77

Qwen2.5 72B - 75

GPT-4 Turbo - 74

Qwen2.5 Coder 32B - 70

Llama 3.1 Nemotron 70B - 70

Claude 3 Opus - 70



AI Stacey the

Here's a simple web app using Chart.js to visualize the AI model scores:

html

```
1  <!DOC
2  <html
3  <head>
```

Save Copy

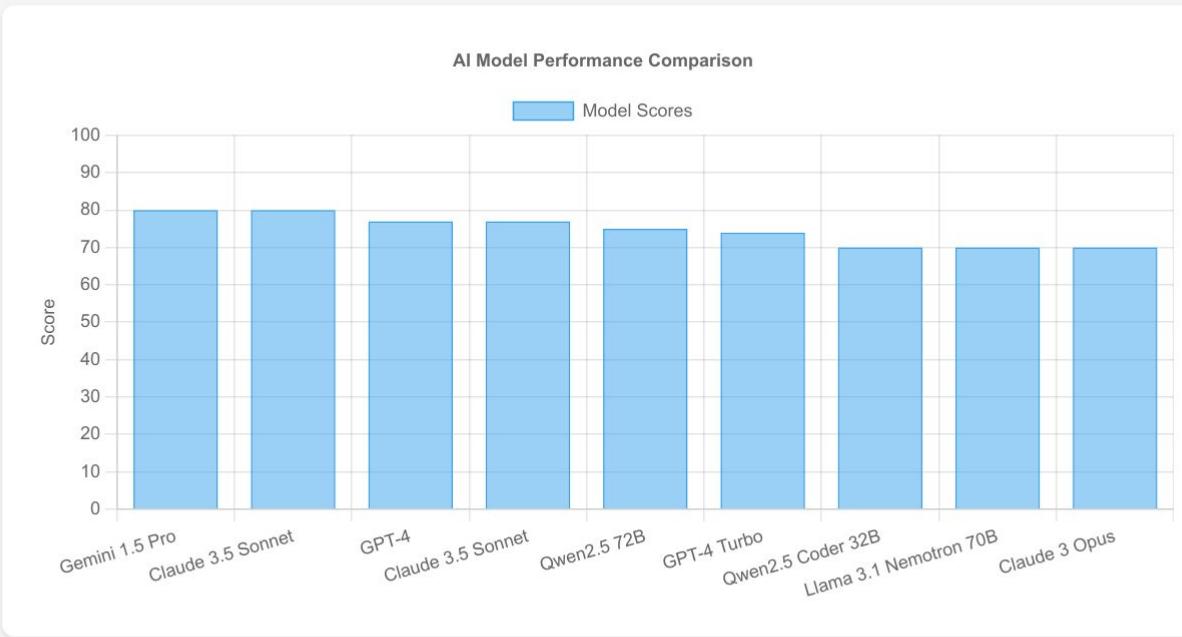
● Run Code, Web Search `charset="UTF-8">`

+ Send a Message





AI Model Performance Scores



Visualized

<https://artificialanalysis.ai/leaderboards/models>

Gemini 1.5 Pro - 80
Claude 3.5 Sonnet - 80
GPT-4o - 77
Claude 3.5 Sonnet - 77
Qwen2.5 72B - 75
GPT-4 Turbo - 74
Qwen2.5 Coder 32B - 70
Llama 3.1 Nemotron 70B - 70
Claude 3 Opus - 70

<https://klu.ai/llm-leaderboard>

Llama3.1:70b Score 86
GPT-4 Omni mini 85
GPT-4 Score 79

<https://www.vellum.ai/llm-leaderboard>

Gemini 1.5 Pro 80.08%
Gemini Ultra 79.52%
GPT-4 79.45%
Llama 3 Instruct - 70B 79.23%
Claude 3 Haiku 75.90%
GPT-3.5 65.46%

AI Model Performance Comparison

Artifician Analysis



KLU Leaderboard



Vellium Leaderboard



02

Function

Data Manipulation

World of Text

- Logs
- Writing Documentation
- Reading Documentation
- Config Files
- Guides
- Manuals
- Tickets
- Knowledge Bases
- SIEMs
- Reports
- Infrastructure as Code
- Comments
- PRs
- Orders
- Incidents

Text Dominance:

Most work is text-based—captured, recorded, or presented as such.

Dynamic Nature:

Text's function, form, and use are too diverse to standardize with a single tool across all workflows.

Automation for Fit:

For tasks amenable to traditional automation, automated tools remain the best solution.

LLMs aren't the best tool to solve any one problem, but perhaps they are becoming decent at solving many problems. Particularly in the world of unstructured data.

Log Analysis

Chat



AI Logan the Log Analyst 12:17 AM

Based on the provided log entries, here is a summary of what happened:

Summary:

- **Timestamps**: The logs are timestamped from around `2024-11-21T02:12` to `2024-11-21T02:28`.
- **IP Addresses**:
 - Client IP : `47.236.68.92`
 - Server IP : `172.18.1.166`

Logs Analysis:

HTTP POST Request:

- The client made a POST request to the server at `/cgi-bin/../../../../../../../../bin/sh`.
- This path is trying to access the `/bin/sh` shell via multiple directory traversals, which suggests an attempt to exploit a known vulnerability or misconfiguration.
 - **HTTP Method** : `POST`
 - **URI** : `/cgi-bin/../../../../../../../../bin/sh`
 - **User Agent** : `Custom-AsyncHttpClient`

HTTP Response:

- The response body length is zero (`response_body_len` : `0`), indicating that the server did not send a meaningful response back.
- Connection state: `RSTO` , meaning the connection was reset by the originator.

File Transfer Details:

- **FUID** : `FZ6ND23eAXLLPtuMRj`
 - This corresponds to data transfer or attempted file access initiated from `47.236.68.92` and received at `172.18.1.166`.

Known Remote Host:

- The client IP `47.236.68.92` was recorded as having two connections to a known remote host.

Observations:

- **Potential Exploitation** : The use of directory traversal (`../../../../../../../../bin/sh`) in the request URI points towards an attempt to exploit a

+ Send a Message

LLMs can make mistakes. Verify important information.



?

To SOC Analyst



AI Logan the Log Analyst +

Monitor Traffic: Continuously monitor such events for repeated attempts from 47.236.89.

Would you like further analysis on specific aspects or additional logs?

The logs mentioned in this triggered the following Suricata alerts. Explain briefly what each are matching on.

txt

```
1 ./exploit.rules:alert http $EXTERNAL_NET any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT"
2 ./web_server.rules:alert http $EXTERNAL_NET any -> $HTTP_SERVERS any (msg:"ET WEB_SERVER /bin/s
3 ./web_server.rules:alert http $EXTERNAL_NET any -> $HTTP_SERVERS any (msg:"ET WEB_SERVER WebShe
4 ./sid-msg.map:2011465 || ET WEB_SERVER /bin/sh In URI Possible Shell Command Execution Attempt
5 ./sid-msg.map:2016683 || ET WEB_SERVER WebShell Generic - wget http - POST
6 ./sid-msg.map:2034125 || ET EXPLOIT Apache HTTP Server 2.4.49 - Path Traversal Attempt (CVE-202
```

Copy

AI Sonny the SOC Analyst 12:25 AM

Based on the Suricata alerts triggered by your logs, here's a summary of what each alert is matching on:

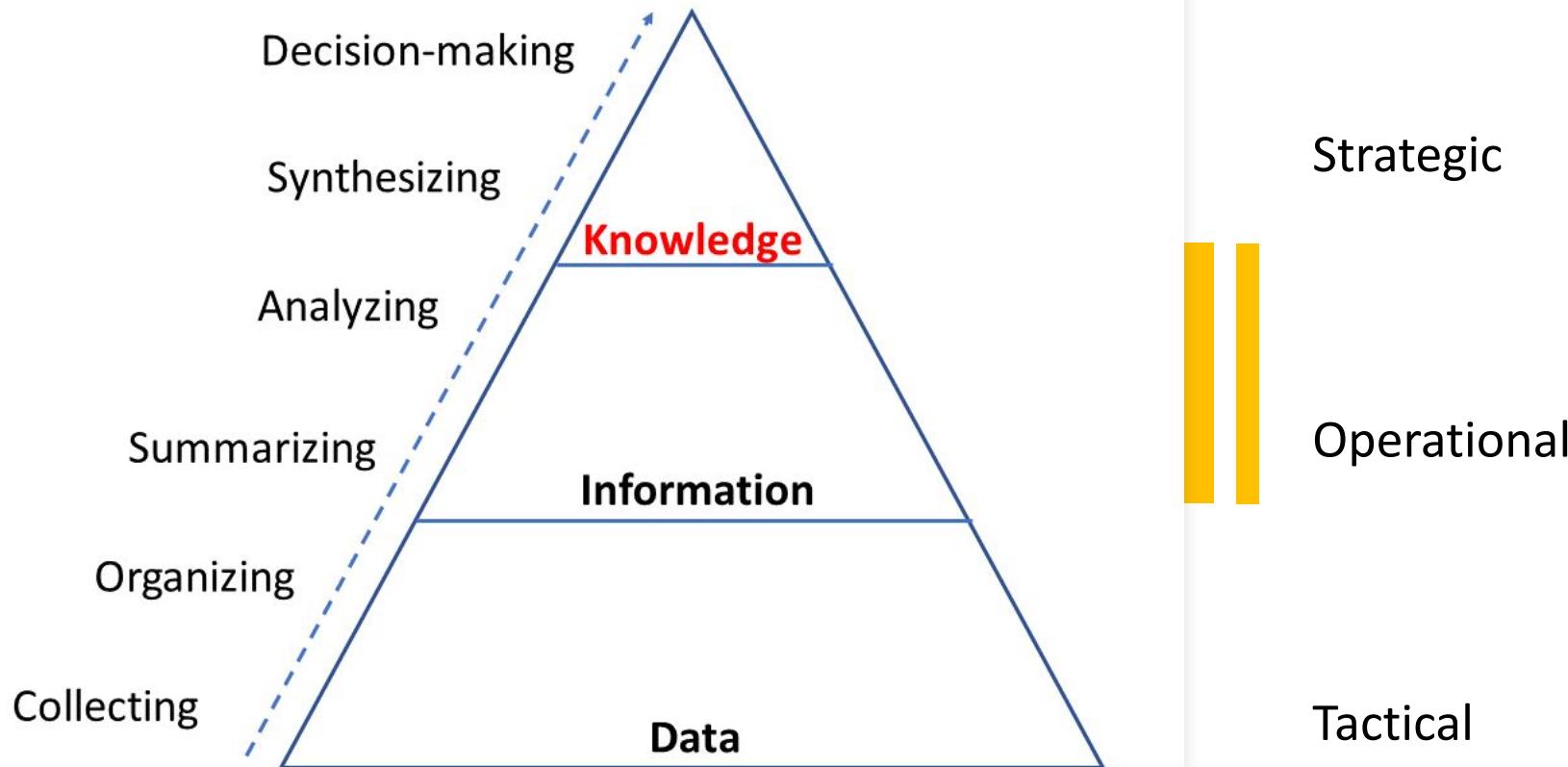
Alerts Summary:

- ET EXPLOIT Apache HTTP Server 2.4.49 - Path Traversal Attempt (CVE-2021-41773) M2 (sid:2034125):**
 - Match Criteria**: The alert triggers on an HTTP POST request to a URI that contains directory traversal sequences (`/.%2e/.%2e/.%2e/.%2e/`). This is indicative of an attempt to exploit the CVE-2021-41773 vulnerability in Apache HTTP Server 2.4.49.
 - Context**: The path traversal URI indicates a potential attack aimed at accessing or exploiting a CGI script via directory traversal, which can lead to remote code execution.
- ET WEB_SERVER /bin/sh In URI Possible Shell Command Execution Attempt (sid:2011465):**
 - Match Criteria**: This alert detects an HTTP request where `/bin/sh` is found in the URI. It's indicative of a potential attempt to execute shell commands via HTTP.
 - Context**: The presence of `/bin/sh` suggests that an attacker might be trying to gain access or execute arbitrary code on the server.

Talking to AI Sonny the SOC Analyst

+ Send a Message

LLMs can make mistakes. Verify important information.

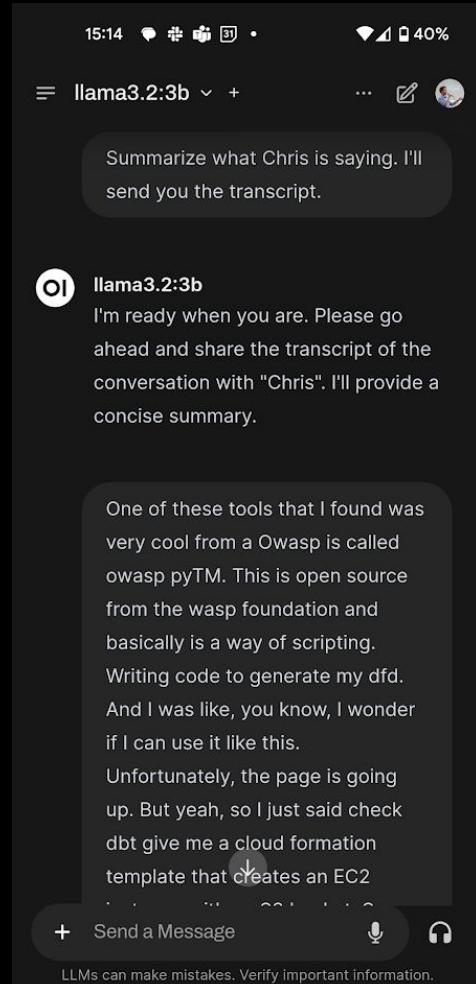


User or Machine Consumption

The screenshot displays a complex interface for AI command-line interaction and file management. On the left, a sidebar lists various workspace and search items. The main area features a terminal window for OpenAI Command-Line Chat, a file explorer showing a 'PROJDIR' workspace with Python scripts, and a code editor with a 'pyScripts (workspace)' tab. The code editor shows a script for 'list-agents-search.py' and a 'models.json' configuration file. The terminal window shows AI interactions with the user, including a summary of the Follina vulnerability and a detailed description of the QakBot threat actor. The file explorer and code editor tabs are also visible, showing the structure of the workspace and the contents of the 'models.json' file.

Portable Consumption

```
christian@ctail-gce22007:~  
$ fabric -y "https://youtu.be/g1JSZwwtEak?feature=shared" --stream --pattern summarize extract_knowledge  
Here's a summary of the content in the requested format:  
  
ONE SENTENCE SUMMARY:  
Microsoft AI plans to release AI models with near-infinite memory in 2025, potentially revolutionizing AI capabilities and human-AI interactions.  
  
MAIN POINTS:  
1. Microsoft AI's head, Mustafa Suleyman, anticipates transformative AI with near-infinite memory by 2025.  
2. Google Research proposed "infinite attention" for efficient handling of long texts without excessive memory use.  
3. Infinite context windows could enable long-term AI relationships and comprehensive knowledge processing.  
4. Eric Schmidt discusses the potential of infinitely long context windows for complex problem-solving.  
5. Memory improvements will allow AI to maintain context across multiple interactions and sessions.  
6. Recursive self-improvement, where AI creates smarter versions of itself, is predicted before 2030.  
7. AI agents are still struggling with reliability in real-world tasks, as shown by benchmark tests.  
8. Current AI models show degrading performance over multiple attempts in agent-based tasks.  
9. Reliable autonomous agents may not be available until around 2026, according to industry experts.  
10. Significant improvements in error rates are needed for practical implementation of AI agents.  
  
TAKEAWAYS:  
1. Near-infinite memory in AI could dramatically enhance its capabilities and user interactions.  
2. Improved context retention may lead to more personalized and efficient AI experiences.  
3. Reliability remains a major challenge for AI agents in real-world applications.  
4. The development of AI is progressing rapidly, with significant advancements expected by 2025-2030.  
5. Successful implementation of AI agents requires substantial reduction in error rates for complex tasks.  
christian@ctail-gce22007:~
```



Documentation

Mistral 7B - 33k (Laptop)

125 words per page - 264 pages

Phi-3 Medium 14B - 128k (Desktop)

300 words per page - 426 pages

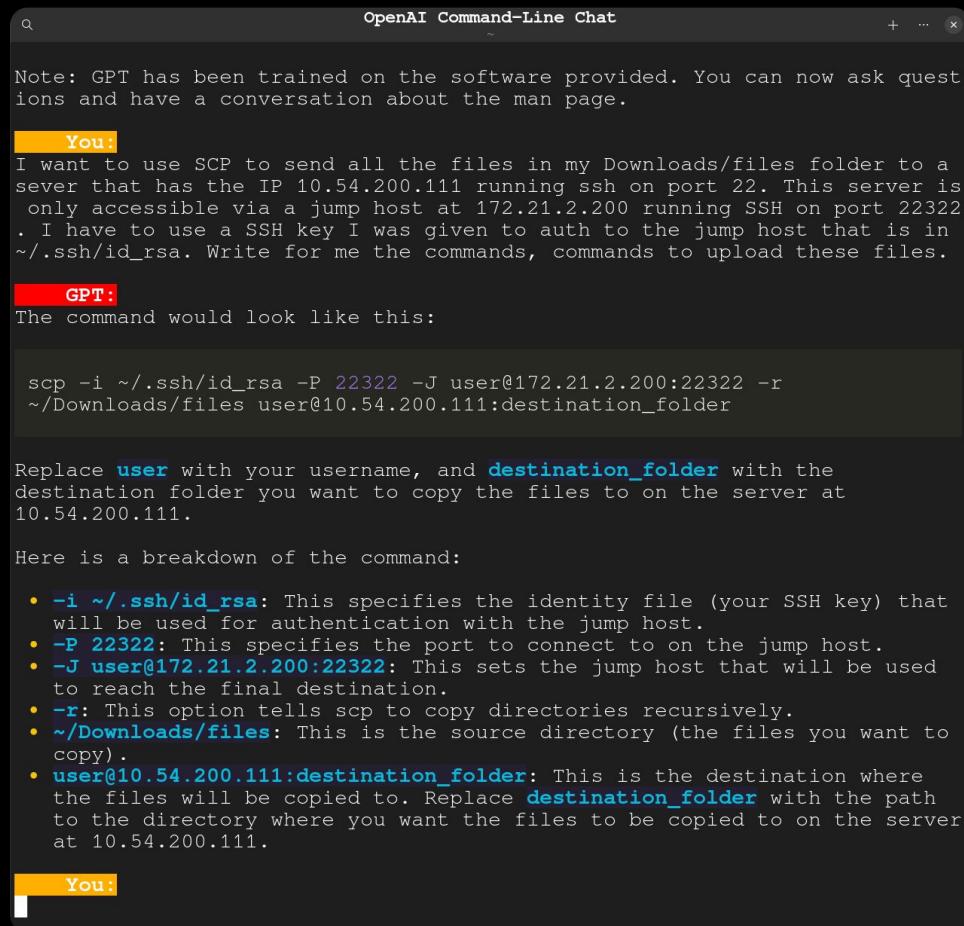
GPT-3.5 - 2k

300 words per page - 5 pages

Hey, AI, add comments to my code.

Embedding Model

Embedding models are a type of machine learning model that learn to represent complex data, such as text or images as a number to calculate the similarity via assessing distance between data points.



The screenshot shows a terminal window titled "OpenAI Command-Line Chat". The window has a dark theme with white text. At the top, a note says: "Note: GPT has been trained on the software provided. You can now ask questions and have a conversation about the man page." A yellow box labeled "You:" contains the user's question: "I want to use SCP to send all the files in my Downloads/files folder to a sever that has the IP 10.54.200.111 running ssh on port 22. This server is only accessible via a jump host at 172.21.2.200 running SSH on port 22322. I have to use a SSH key I was given to auth to the jump host that is in ~/.ssh/id_rsa. Write for me the commands, commands to upload these files." A red box labeled "GPT:" contains the AI's response: "The command would look like this:" followed by a code block:

```
scp -i ~/.ssh/id_rsa -P 22322 -J user@172.21.2.200:22322 -r ~/Downloads/files user@10.54.200.111:destination_folder
```

 Below the command, a note says: "Replace `user` with your username, and `destination_folder` with the destination folder you want to copy the files to on the server at 10.54.200.111." The AI then provides a breakdown: "Here is a breakdown of the command:" followed by a list of bullet points explaining each part of the SCP command.

- `-i ~/.ssh/id_rsa`: This specifies the identity file (your SSH key) that will be used for authentication with the jump host.
- `-P 22322`: This specifies the port to connect to on the jump host.
- `-J user@172.21.2.200:22322`: This sets the jump host that will be used to reach the final destination.
- `-r`: This option tells scp to copy directories recursively.
- `~/Downloads/files`: This is the source directory (the files you want to copy).
- `user@10.54.200.111:destination_folder`: This is the destination where the files will be copied to. Replace `destination_folder` with the path to the directory where you want the files to be copied to on the server at 10.54.200.111.

A yellow box labeled "You:" is at the bottom of the window.

OpenAI Command-Line Chat

```
No manual entry for corelight-update
No man page found for corelight-update. Do you want to try running 'corelight-update -h'? (y/n): y

Note: GPT has been trained on the software provided. You can now ask questions and have a conversation about the man page.

You:
I want to use corelight-update to build and push my package bundle to "main" policy. Add my CrowdstrikeSpotlite-policy. Then deploy the existing content to main.

GPT:
Here is how you can achieve this:

First, use the "-b" option to build and push a package bundle for the "main" policy:

corelight-update -b main

Next, use the "add" command to add your "CrowdstrikeSpotlite-policy":

corelight-update add --policy CrowdstrikeSpotlite-policy

Finally, use the "-f" option to force deploy the existing content to the "main" policy:

corelight-update -f main

Be sure to replace "CrowdstrikeSpotlite-policy" with the correct name of your policy.

You:
```

≡ AI Stacey the Scripter +

```
110
111      },
112      animation: {
113        duration: 1000
114      },
115      responsive: true,
116      maintainAspectRatio: false
117    }
118  );
119
120  const ctx = document.getElementById('modelChart').getContext('2d');
121  const myChart = new Chart(ctx, config);
122
123  // Set canvas height based on number of models
124  document.querySelector('.chart-container').style.height
125  </script>
126 </body>
127 </html>
```

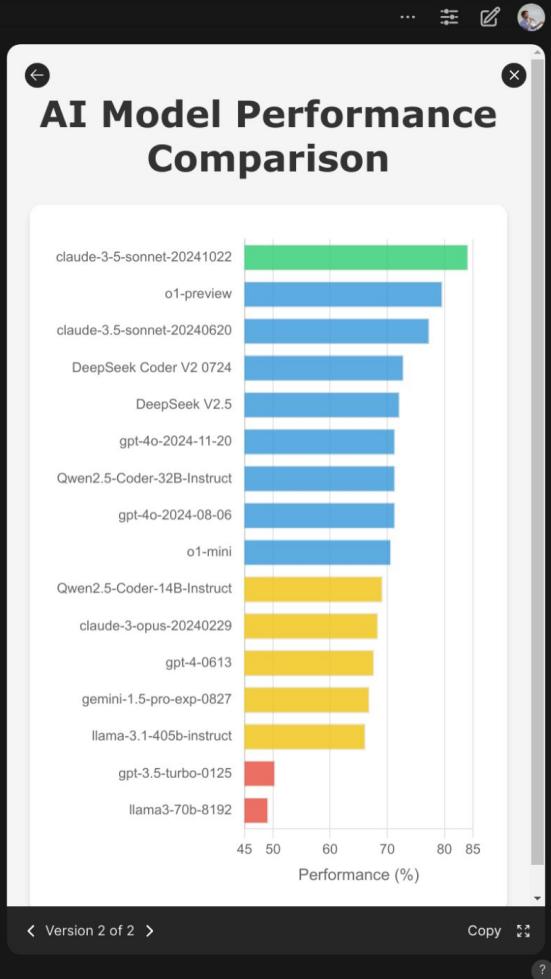
And the Python Flask server remains the same:

```
python
1  from flask import Flask, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      return render_template('index.html')
8
9  if __name__ == '__main__':
10    app.run(debug=True)
```

The improvements in this version include ↴

Run Code, Web Search

Send a Message

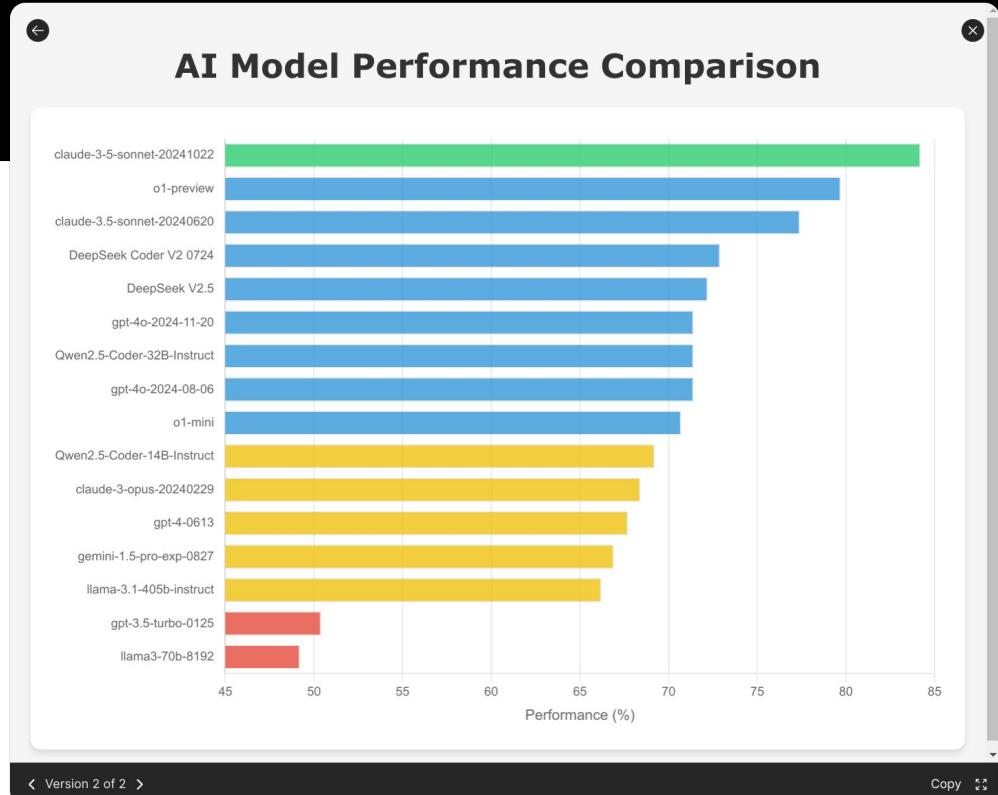


Stacey can write and present simple web apps, write in numerous languages, run python from chat, correct bugs, remediate vulns, and read repos.

Aider Coding Model

Stacey's Graph

[Aider's code editing benchmark](#) asks the LLM to edit python source files to complete 133 small coding exercises from Exercism. This measures the LLM's coding ability, and whether it can write new code that integrates into existing code. The model also has to successfully apply all its changes to the source file without human intervention.



But doesn't Copilot Write Vulnerable Code?

Yep! But it can also fix it!

The vulnerability you've mentioned is introduced through the `org.springframework.boot:spring-boot-starter-web:2.6.7` dependency in your project, which transitively includes `org.springframework:spring-webmvc:5.3.19`. The fix for this vulnerability involves upgrading to a newer version of `spring-boot-starter-web`, specifically version `2.6.15`.

To remediate the issue, you need to update the `pom.xml` file in your project that is located in the root directory of your project. Here's how you can do it:

1. Open the `pom.xml` file.
2. Locate the dependency for `org.springframework.boot:spring-boot-starter-web`.
3. Change its version from `2.6.7` to `2.6.15`.

Here is a snippet of what the updated section in your `pom.xml` might look like:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <version>2.6.15</version>
</dependency>
```

After updating the version, save the changes and rebuild your project to ensure that the updated dependency is included.

If you are using a build tool other than Maven (e.g., Gradle), you will need to update the corresponding configuration file (`build.gradle` for Gradle) in a similar manner.

`pom.xml` Here is the file. Give me the corrected content.

MichaelAngelo - qwen2.5-coder:14b

1 context item

To remediate the vulnerability, you need to update the `spring-boot-starter-parent` version from `2.6.7` to `2.6.15`. This change will ensure that all dependencies managed by Spring Boot are updated accordingly, including `spring-webmvc`.

Here is the corrected content of your `pom.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.15</version>
  </parent>

  <groupId>fi.solita.clamav</groupId>
  <artifactId>clamav-rest</artifactId>
  <version>1.0.2</version>
  <packaging>jar</packaging>
  <name>Simple ClamAV proxy</name>
  <description>Simple REST proxy to put between</description>
  <url>https://github.com/solita/clamav-rest</url>
  <licenses>
    <license>
      <name>GNU LESSER GENERAL PUBLIC LICENSE</name>
      <url>http://www.gnu.org/licenses/lgpl.html</url>
    </license>
  </licenses>
  <developers>
    <developer>
      <name>Antti Virtanen</name>
      <email>antti.virtanen@solita.fi</email>
    </developer>
  </developers>
  <organization>
    <name>Solita</name>
  </organization>
</parent>
```

Ask anything, '/' for slash commands, '@' to add context

MichaelAngelo - qwen2.5-coder:14b

New Session (^ L)

Ln 21, Col 12 Spaces: 2 UTF-8 LF { XML Continue

03

Models

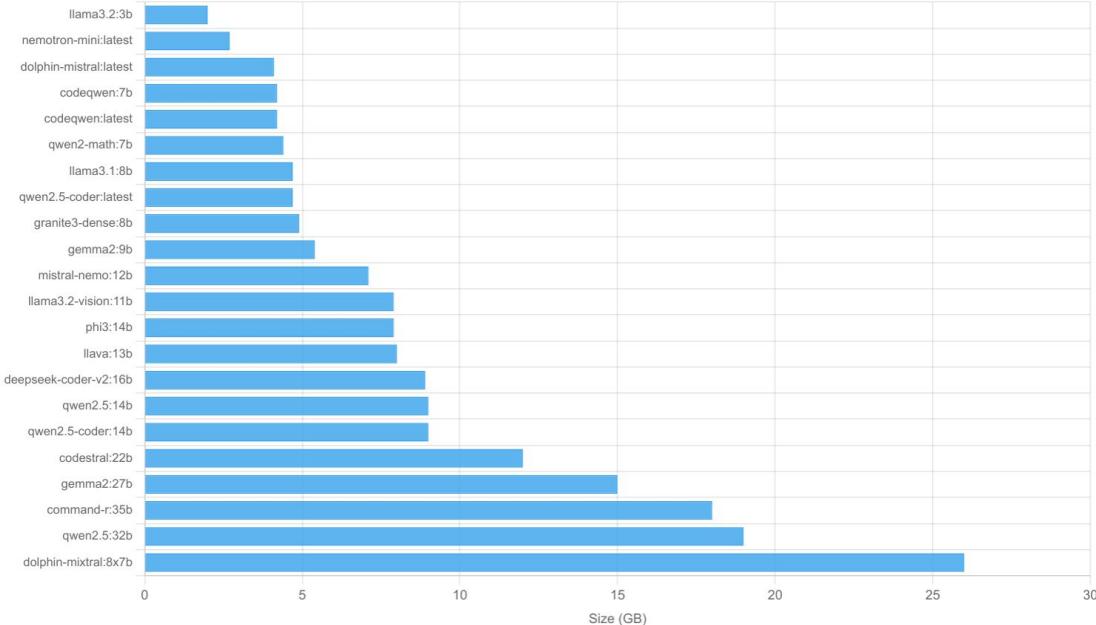
Models

And their sizes



AI Models Size Comparison

Model Size (GB)



Knowledge and Model Size

Larger Models

- Generally more capable
- Expensive to train
- Generally Lack of Specialized Models
- Comes with greater world knowledge
- Greater Zero Shot Logic and Reasoning
- Slower
- Increased hardware requirements

Smaller Models

- Generally less capable
- Cheap to train
- Specialized Models
- Less world knowledge (besides specialization)
- Weaker Zero Shot Logic and Reasoning
- Faster
- Reduced hardware requirements

Quantization to reduce model size at precision cost.

Augment LLM

Fine Tuning

Tune to modify or refine model behavior, give model new skill or capability. Offers the capability to change how the model acts.

Syntax learning, writing style, formatting, are all useful examples of modifications that would benefit from fine tuning.

RAG

Use RAG to modify models knowledge of facts. Provides more reliable consistent fact recall by using AI to call relevant information.

Searching the Internet, knowledge base, or your SIEM are all useful examples of modifications that would benefit from fine tuning.

Machine Learning Profiles/GPTs/Modefiel - Old Agents

GIST

AI Madison the Cybersecurity Manager

Model qwen2.5:32b

A cybersecurity manager responsible for cybersecurity strategy and operations with a focus on security policies, risk management, team leadership, and strategic planning within the cybersecurity domain.

Profile: Prompt + Knowledge + Tools + Actions + Capabilities



≡ Models Knowledge Prompts Tools

Tools

To select toolkits here, add them to the "Tools" workspace first.

Web Search Run Code Web Scrape

Filters

To select filters here, add them to the "Functions" workspace first.

MoEA

Actions

To select actions here, add them to the "Functions" workspace first.

Visualize Data

Capabilities

Vision Usage Citations

04

Challenges

Challenges

Use Case Issue

Incorrect application of this technology because it is “easy”. Many traditional tools are still the best for their job.

Example: Performing anomaly detection, or other big data science function.

Model Weakness

Use statistical correlation to reflect and leverage the patterns and purpose of language – partially inferring logical construct.

Real-world situations involve incredible nuances. Prompts are often the size of a tweet.

Knowledge & Context Limits

Models are trained up to a certain point of time. RAG can accommodate this, but context is limited.

Note: Although context size may be large, models can lose track of information relevance.

Reliability of Probability

We work with risk and probability all the time.

At the end of the day, it is a guess. Just like our modern machine learning.

What is our “false” tolerance?



Model Security

Deserialization Threats

Untrusted data/code reconstructs objects, leading to exploitation.

Malicious actors inject harmful code during deserialization.

Crucial for securing data integrity and preventing unauthorized code execution.

models-scan

Architectural Backdoors

Parallel path from model input to output, manipulated by attackers.

Model behaves normally with non-malicious inputs, unexpected behavior with triggers.

Attacker controls when backdoored model gives incorrect output.

Runtime Threats

Untrusted data/code exploits vulnerabilities during inference or execution.

Malicious actors inject harmful code, gaining unauthorized access or manipulating systems.

Crucial for securing data integrity and preventing unauthorized code execution.

05

Tooling

FOSS Stack

ZTNA + Firewall Decrypted

OpenWeb UI | Obsidian

OpenAI API | Auth Proxy

Ollama | vLLM | Auth Proxy

STT:Whisper | TTS:SpeechT5

RAG: nomic-embed-text

Why Ollama?

Because it is easy.

```
christian@michelangelo:~ ollama run llama3.2:3b --verbose
$ ollama pull llama3.2:3b
pulling manifest
pulling dde5aa3fc5ff... 100% 2.0 GB
pulling 966de95ca8a6... 100% 1.4 KB
pulling fcc5a6bec9da... 100% 7.7 KB
pulling a70ff7e570d9... 100% 6.0 KB
pulling 56bb8bd477a5... 100% 96 B
pulling 34bb5ab01051... 100% 561 B
verifying sha256 digest
writing manifest
success
christian@michelangelo:~ $ ollama run llama3.2:3b --verbose
>>> Hello there!
It's nice to meet you. Is there something I can help you with
or would you like to chat?

total duration: 196.066373ms
load duration: 30.393ms
prompt eval count: 28 token(s)
prompt eval duration: 23ms
prompt eval rate: 1217.39 tokens/s
eval count: 23 token(s)
eval duration: 142ms
eval rate: 161.97 tokens/s
>>> Send a message (/? for help)
```

“GPU Poor” for Testing

XPS 15 2020

Old Corporate Laptop

Intel i7-10750H

NVIDIA GTX 1650 Ti Mobile

4GB VRAM

32 GB RAM

~7b at reading speed (5t/s)

ThinkStation P510

Old Donated Workstation

Intel Xeon E5-2609 v4

NVIDIA GTX 1080

8GB VRAM

96GB RAM

~9b at reading speek (12t/s)

Gaming Desktop

~\$1k Desktop

12th Gen Intel i7-12700F

NVIDIA GeForce RTX 3080

10GB VRAM

32GB RAM

32b at reading speed (6ts)

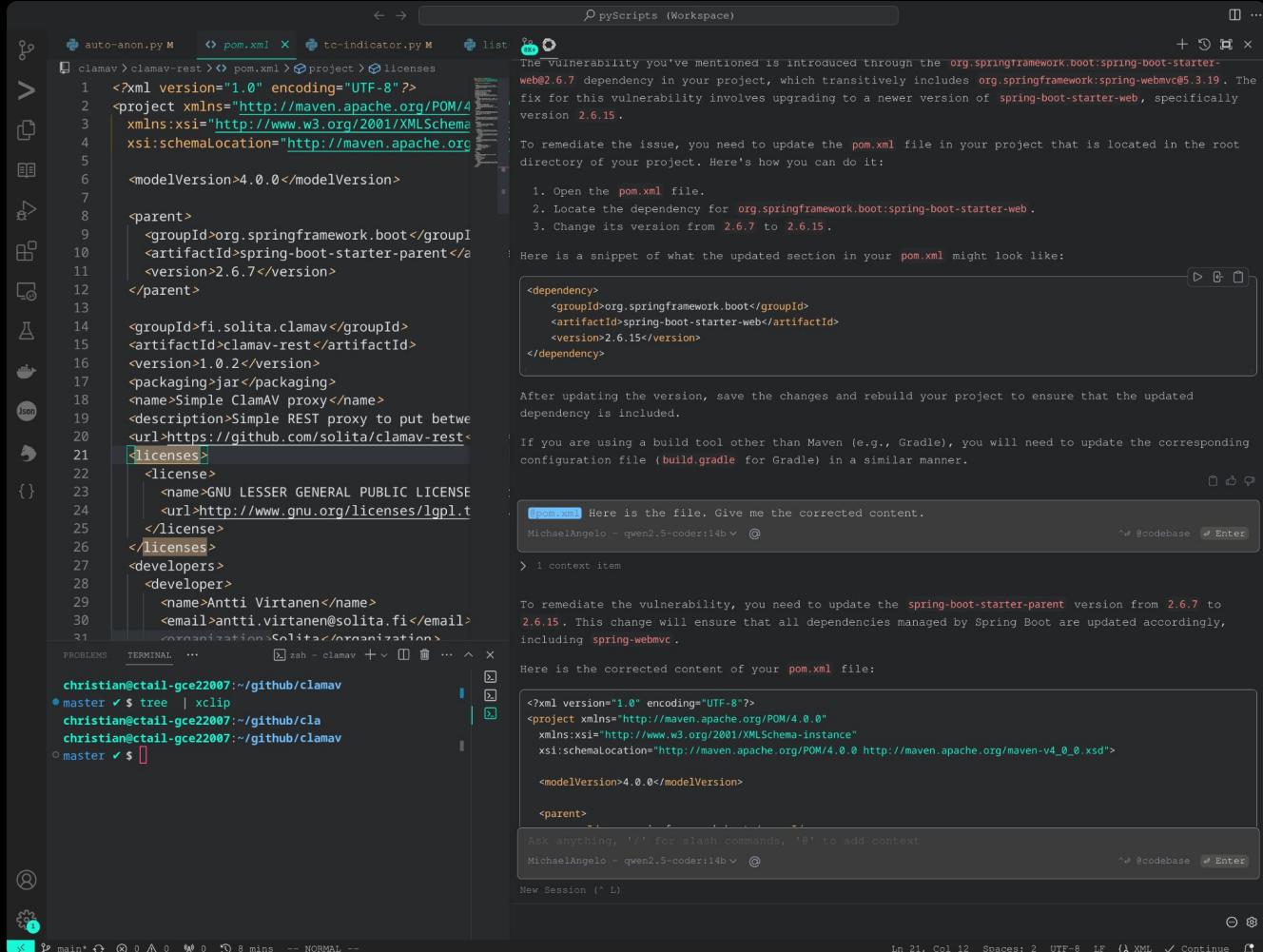
Recommendation: High VRAM GPU (3090 or 4090)

2x3090 = 48GB VRAM = llama3.1:70b | reflection:70b | nvidia nemotron:70b

The image shows a screenshot of the Obsidian application interface. On the left, there is a 'Graph view' window displaying a complex network graph with numerous nodes (represented by small dots) and edges (represented by lines) in various colors (blue, red, green, yellow). A specific node is highlighted with a larger orange circle and labeled 'COZY BEAR'. On the right, the main Obsidian workspace shows a card-based interface. At the top, a question is asked: 'What would you like to talk about with your notes?'. Below this, a card displays a query: 'Tell me about the Adversary Cozy Bear (APT 29). What have they been up to recently?'. The response card below the query provides information about Cozy Bear (APT 29), also known as UNC2452 or APT29, which is described as a Russian threat actor. It mentions that they have been actively engaged in malicious activities, specifically sending targeted spear-phishing emails to thousands of individuals in over 100 organizations since October 22, 2024. The response also links to a specific file: [actra/programs/vsrt-exchange/2024/2024-11-November-vsrt](#). The text continues to describe the spear-phishing emails, mentioning they contain a signed Remote Desktop Protocol (RDP) configuration file that connects to an actor-controlled server, potentially exposing sensitive information and allowing the installation of malware. It also links to another file: [actra/programs/vsrt-exchange/2024/2024-11-November-vsrt](#). The final part of the response emphasizes the importance of vigilance and taking necessary precautions to protect against such targeted attacks. At the bottom of the workspace, there is a chat interface with the text 'Chat with your Smart Second Brain...' and a brain icon. The status bar at the bottom right shows '10 files 0 min read ESM'.

Talk with your knowledge management. Embed files with **nomic-embed-text** and allow statistics and probability to find the relevant files for your to have conversations with your data using a light model like **llama3.2:3b**.

Obsidian + Second Brain + Ollama



The vulnerability you've mentioned is introduced through the `org.springframework.boot:spring-boot-starter-web:2.6.7` dependency in your project, which transitively includes `org.springframework:spring-webmvc:5.3.19`. The fix for this vulnerability involves upgrading to a newer version of `spring-boot-starter-web`, specifically version `2.6.15`.

To remediate the issue, you need to update the `pom.xml` file in your project that is located in the root directory of your project. Here's how you can do it:

1. Open the `pom.xml` file.
2. Locate the dependency for `org.springframework.boot:spring-boot-starter-web`.
3. Change its version from `2.6.7` to `2.6.15`.

Here is a snippet of what the updated section in your `pom.xml` might look like:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <version>2.6.15</version>
</dependency>
```

After updating the version, save the changes and rebuild your project to ensure that the updated dependency is included.

If you are using a build tool other than Maven (e.g., Gradle), you will need to update the corresponding configuration file (`build.gradle` for Gradle) in a similar manner.

`pom.xml` Here is the file. Give me the corrected content.

MichaelAngelo - qwen2.5-coder:14b

1 context item

To remediate the vulnerability, you need to update the `spring-boot-starter-parent` version from `2.6.7` to `2.6.15`. This change will ensure that all dependencies managed by Spring Boot are updated accordingly, including `spring-webmvc`.

Here is the corrected content of your `pom.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.7</version>
  </parent>

  <groupId>fi.solita.clamav</groupId>
  <artifactId>clamav-rest</artifactId>
  <version>1.0.2</version>
  <packaging>jar</packaging>
  <name>Simple ClamAV proxy</name>
  <description>Simple REST proxy to put between</description>
  <url>https://github.com/solita/clamav-rest</url>
  <licenses>
    <license>
      <name>GNU LESSER GENERAL PUBLIC LICENSE</name>
      <url>http://www.gnu.org/licenses/lgpl.html</url>
    </license>
  </licenses>
  <developers>
    <developer>
      <name>Antti Virtanen</name>
      <email>antti.virtanen@solita.fi</email>
    </developer>
  </developers>
  <organization>
    <name>Solita</name>
  </organization>
</parent>
```

Ask anything, '/' for slash commands, '@' to add context

MichaelAngelo - qwen2.5-coder:14b

New Session (^ L)

Ln 21, Col 12 Spaces: 2 UTF-8 LF { XML Continue

Use Continue.dev to connect to various third party APIs for inference, OpenAI compatible AIs, or Ollama in a CoPilot like experience.

Continue.dev + OpenWebUI

06

Frontier

AI Ceiling?

Why are community models
catching up to the big players?

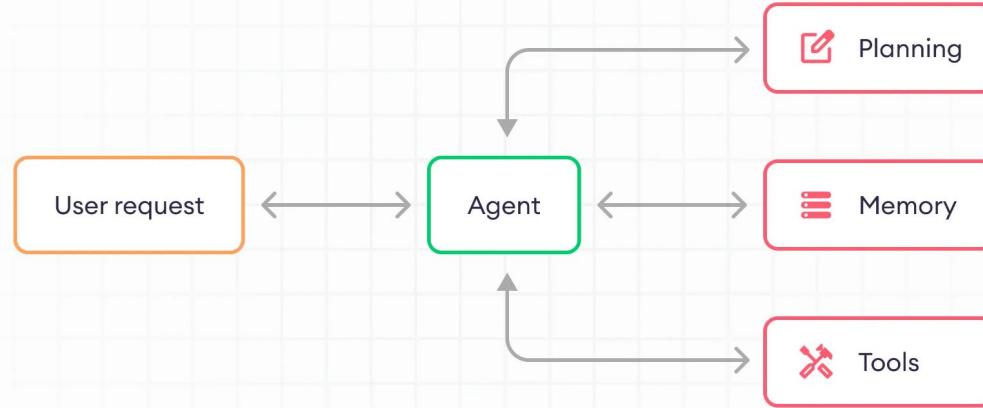
LLM Agency

Allowing LLMs to determine when to use which tools. Provisioning LLMs with the capabilities to “do” things.

Having LLMs think through a process and assess its results. Reconsider if the results are unexpected.

Multishot + Agency

- Plan
- Reflect
- Search the internet
- Scrape a web pack
- Run code



Accelerating Improvements

RAG Embedding

Reduce storage capacity requirements
Fasters
Reranking

Implication: arger datasets can fuel RAG effectively

Quantization

More strategic quantization that preserves knowledge and removes noise.

Implication: Smaller models can be more effective.

Fine Tuning

Better data
Tuning Efficiency
Strategic tuning

Implication: Smaller models can be more effective.

Hardware Scale

Hardware acceleration designed for AI tasks

Implication: Costs of running AI will continue to shrink

THANK YOU

<https://christiant.io>

