

Interacting With The RAG UI

Introduction

We've developed an application that lets you use all the [Cyber Science Lab](#) RAGs (Retrieval-Augmented Generation) from one platform. At the current time there are 4 RAGs available: [Penetration Testing](#), [CVE](#), [Malware](#) and [Threat Intelligence](#). The process works by processing the user requests and retrieving context based on the selected RAG model and then uses the chosen LLM (Large Language Model) to generate accurate responses. The frontend allows you to select the LLM model, configure RAG settings, upload files, and view the results in real-time, making it easy to work with our cybersecurity RAGs and get the information you need.

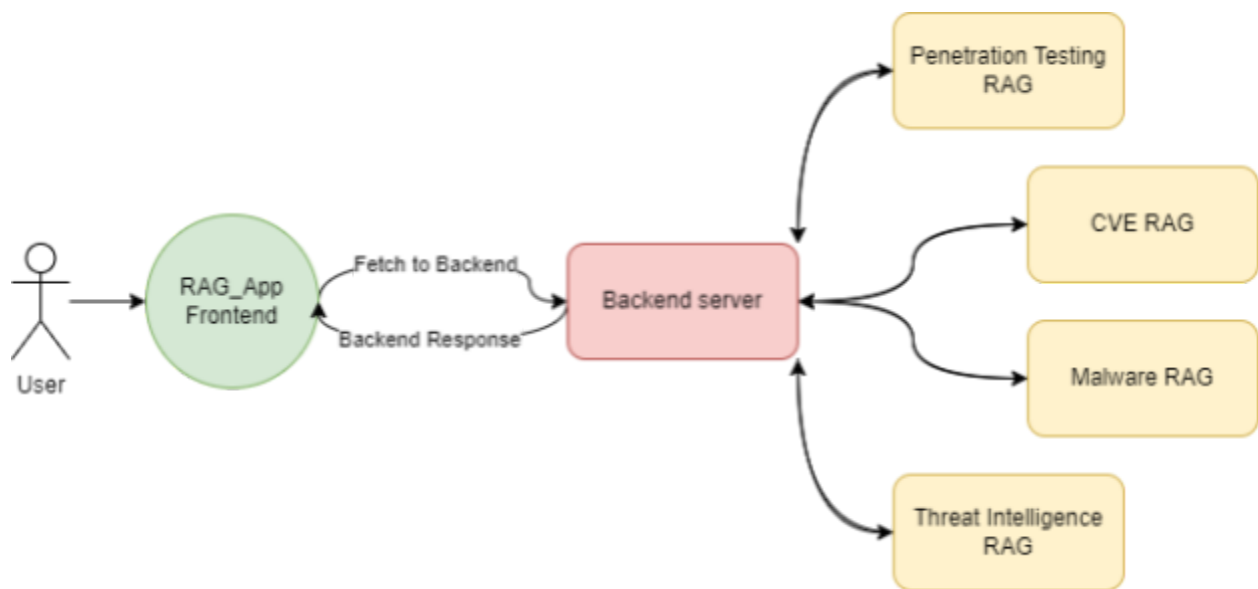


Diagram of Penetration Testing RAG

The user enters all their information on the RAG App UI like the prompt, which model, RAG type, # of context to retrieve and can upload a file. That information is bundled up and sent in the fetch request to the backend server. The server then requests the user selected RAG and completes some additional processing before sending the response back to the frontend to be displayed to the user.

Project Setup

1. Download the required programs: [Docker](#), [Anaconda](#), [Git](#), [Node](#)
2. Clone project from the [Cyber Science Lab Github](#)
> **git clone https://github.com/CyberScienceLab/RAG_App.git**
3. Change to the RAG_App directory
> **cd RAG_App**
4. Setup Frontend
 - a. Create a new terminal session
 - b. Change to frontend directory
> **cd RAG_App/frontend**
 - c. Install all frontend dependencies
> **npm i**
 - d. Update **VITE_BACKEND_URL** in .env file. Replace '**REPLACE_WITH_SERVER_ADDRESS**' with the address of the device running your backend
 - e. Start frontend
> **npm run dev**
5. Setup Backend
 - a. You must have at least one of the following RAGs successfully running on your device to use the backend. Any RAG that hasn't been setup yet will still show on UI, but will not work.
 - b. Create a new terminal session
 - c. Change to backend directory
> **cd RAG_App/backend**
 - d. Clone all RAG projects and install all dependencies
> **./setup.sh**
 - e. Add your Google API Key to the environment
 - 1) Go to <https://aistudio.google.com/app/apikey>, login and click 'Create API Key'
 - 2) Copy your API key
 - 3) Create ENV variable for the Google API Key
> **export GOOGLE_API_KEY=yourGoogleAPIKey**
 - f. Start server
> **python server.py**

Using RAG App

Navigate to `http://SERVER_ADDRESS:3000` and you should see the following

The screenshot shows the RAG App interface. On the left, there is a 'RAG Config' sidebar. It contains a 'Model' dropdown menu with 'Llama3' selected. Below it is a 'RAG Types' dropdown menu with 'CVE', 'Threat Intelligence', 'Pen-Testing', and 'Malware' options. The '# of Relevant Context Used' is set to '5'. At the bottom of the sidebar is a 'File Upload' section with a 'Choose File' button and a 'No...sen' button. The main area on the right is a large light gray rectangle. At the bottom of this area is a text input field with the placeholder 'Enter a prompt here' and a circular button with an upward arrow.

Select the LLM model you would like to use

This screenshot is similar to the previous one, but the 'Model' dropdown menu is open, showing a list of options: 'Llama3', 'Llama3', and 'Gemini'. The first 'Llama3' option is highlighted with a blue background. The rest of the interface remains the same.

Select the RAG type you want to use

This screenshot shows the 'RAG Types' dropdown menu open, displaying the options 'CVE', 'Threat Intelligence', 'Pen-Testing', and 'Malware'. The 'Pen-Testing' option is highlighted with a blue background. The 'Model' dropdown remains closed and set to 'Llama3'. The rest of the interface is identical to the previous screenshots.

Enter your question in the prompt section

RAG Config

Model
Llama3

RAG Types
CVE
Threat Intelligence
Pen-Testing
Malware

of Relevant Context Used
5

File Upload
Choose File No...sen

User prompt can be entered here!!!!


(Optional) Upload a file to use with your prompt

RAG Config

Model
Llama3

RAG Types
CVE
Threat Intelligence
Pen-Testing
Malware

of Relevant Context Used
5

File Upload
Choose File Sh....pdf

User prompt can be entered here!!!!


Press enter on your keyboard or click the button in the bottom right

RAG Config

Model
Llama3

RAG Types
CVE
Threat Intelligence
Pen-Testing
Malware

of Relevant Context Used
5

File Upload
Choose File Sh....pdf

User prompt can be entered here!!!!


Additional Examples

RAG Config

Model

Gemini

RAG Types

CVE

Threat Intelligence

Pen-Testing

Malware

of Relevant Context Used

5

File Upload

Choose File

No...sen

How do I break out of a Docker container?

Gemini -- View Relevant Context

Exploit: Docker - Container escape

Type: Local exploit that targets Linux

Information: Published in 2019 by Dominikczarnotatob

Description: This exploit utilizes the "cgroup" feature to escape a docker container and execute commands on the host. By creating a "release agent" with the ability to access the host filesystem, the container can execute arbitrary commands by utilizing command substitution and piping the results to the host. The steps involved in this exploit include: 1. Mounting the "cgroup" filesystem into the container. 2. Creating a "release agent" file that references the host filesystem. 3. Creating a script that executes commands on the host. 4. Using command substitution ("\$(") to execute the script on the host. This exploit is effective against containers that have access to the "SYS_ADMIN" capability and are running with an unconfined apparmor profile. The code snippets provided in the exploit data include instructions on how to create the necessary files and execute the exploit within the docker container.
<https://gitlab.com/exploit-database/exploitdb/-/blob/main/exploits/linux/local/47147.txt>

Enter a prompt here

↑

RAG Config

Model

Llama3

RAG Types

CVE

Threat Intelligence

Pen-Testing

Malware

of Relevant Context Used

5

File Upload

Choose File

No...sen

can you give me a sha256 hash that is about bruteforce attack

Llama3 -- View Relevant Context

Based on the provided data, I found a hash that is related to a bruteforce attack:
`7c604da002a891ac341f941ed5bae94f2440e07fc44d852dbc467a36eb6c8ede` This is the SHA256 hash of a file named `d811644fc7ac48303ae3bba32cc9e182` which is described as performing a bruteforce attack in the network.

Enter a prompt here

↑

RAG Config

Model

Llama3

RAG Types

CVE

Threat Intelligence

Pen-Testing

Malware

of Relevant Context Used

5

File Upload

Choose File

No...sen

How do I privilege escalate on a Linux server with kernel 2.x?

Llama3 -- View Relevant Context

Exploit: Linux kernel 2.0.37 - segment limit privilege escalation

Type: Local privilege escalation exploit for Linux kernel 2.0.37

Information: Author: Solar, published in 1999

Codes: ["CVE-1999-1166", "OSVDB-12954"]

Description: This exploit takes advantage of a bug in the Linux kernel 2.0.37 that allows a user process to access kernel memory. It can be used to gain root privileges.
<https://gitlab.com/exploit-database/exploitdb/-/blob/main/exploits/linux/local/19419.c>

Exploit: Linux kernel 2.2.x/2.4.x - privileged process hijacking privilege escalation (1)

Type: Local privilege escalation exploit for Linux kernel 2.2.x/2.4.x

Information: Author: Anszom, published in 2003

Enter a prompt here

↑



CANADA CYBER FOUNDRY



CYBERSCIENCELAB

Secure the future with us

cybersciencelab.com