CyberScienceLab

University of Guelph

Canada Cyber Foundry

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.

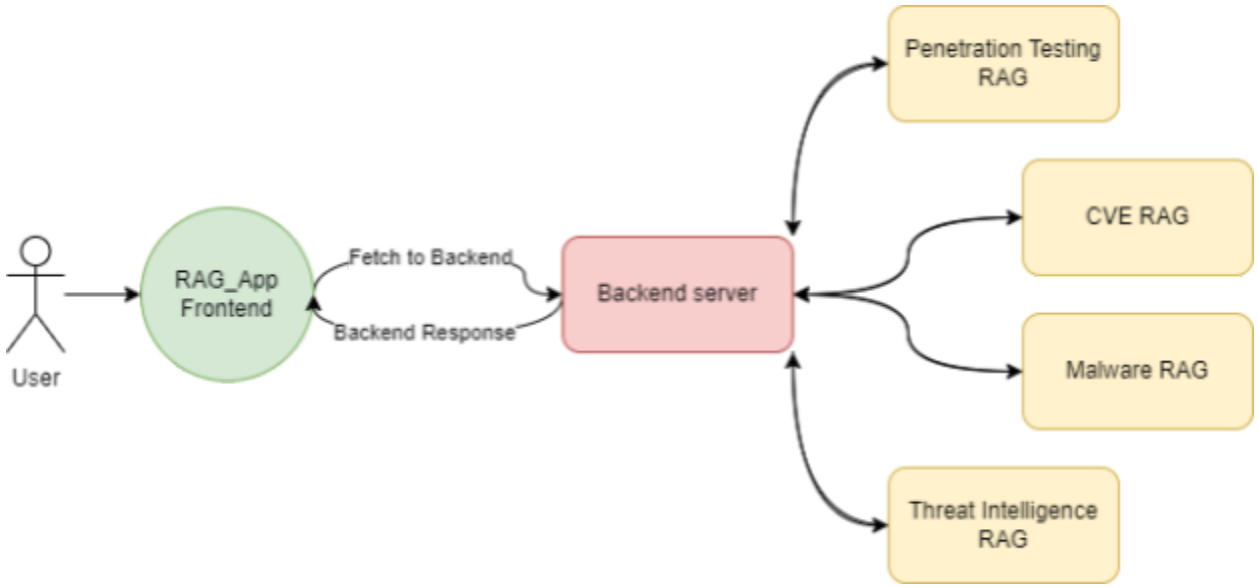


Diagram of Penetration Testing RAG

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**

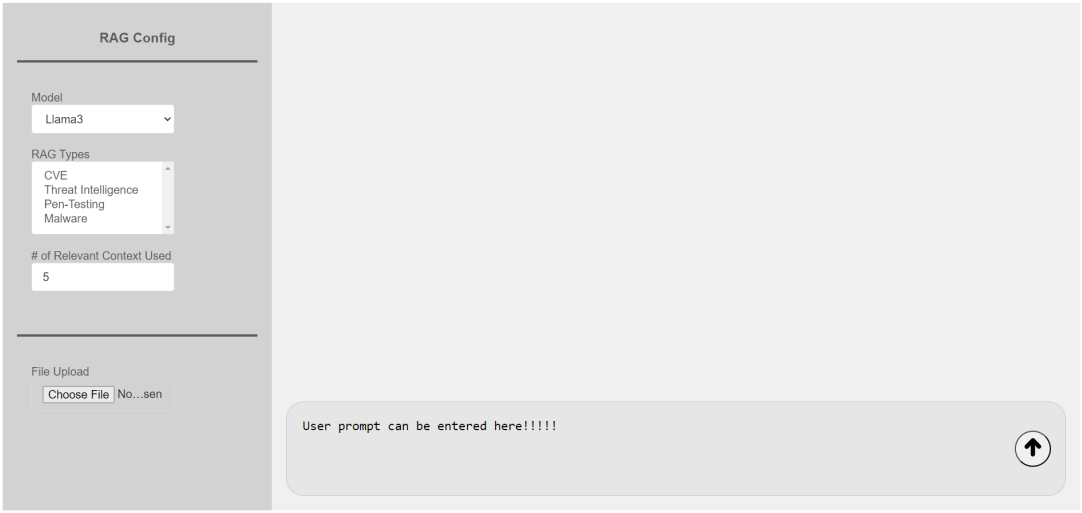
Using RAG App

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

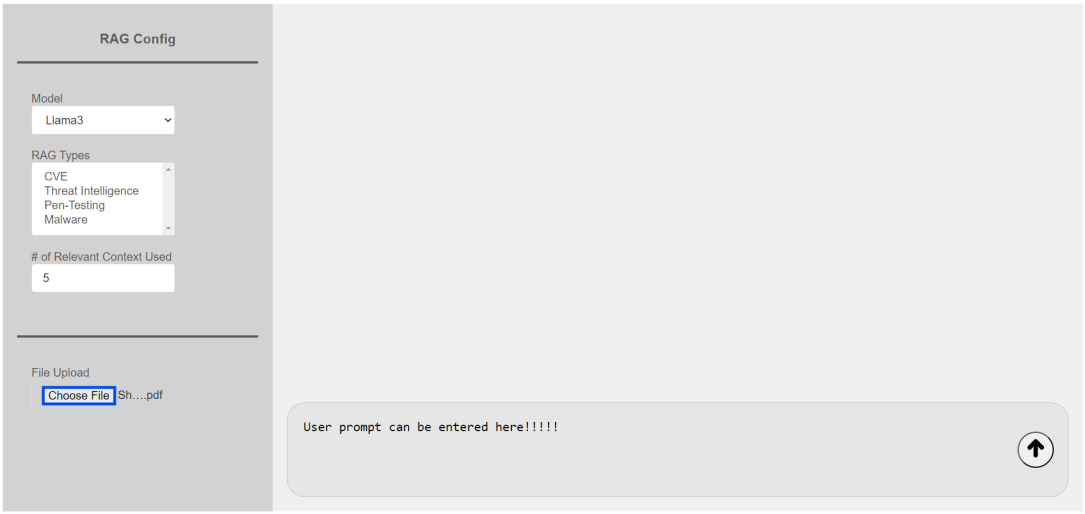
Select the LLM model you would like to use

Select the RAG type you want to use

Enter your question in the prompt section

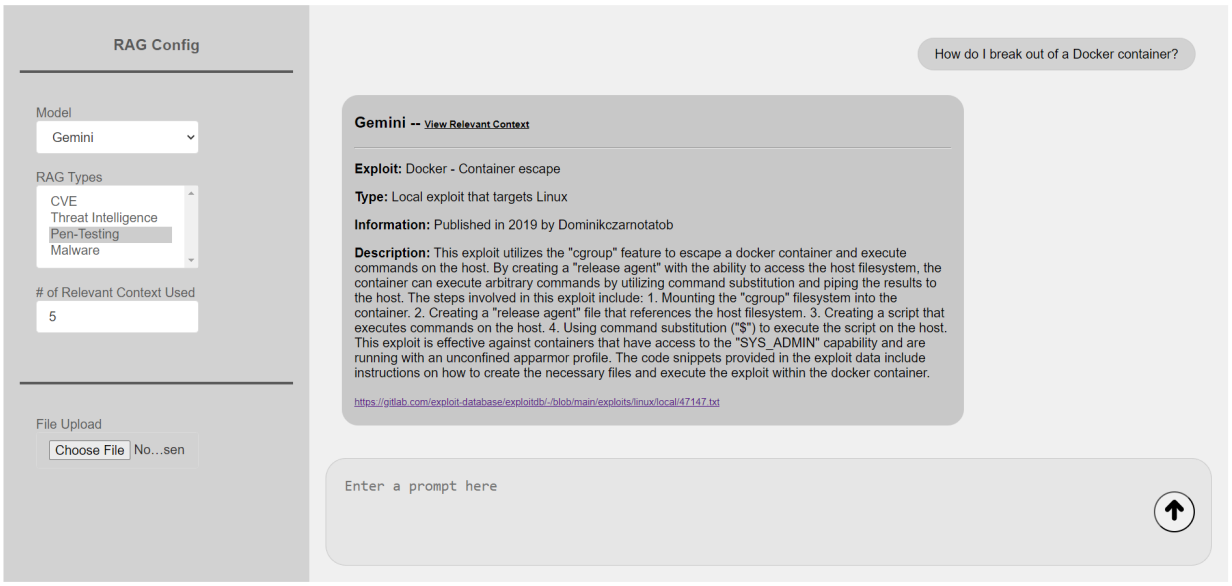


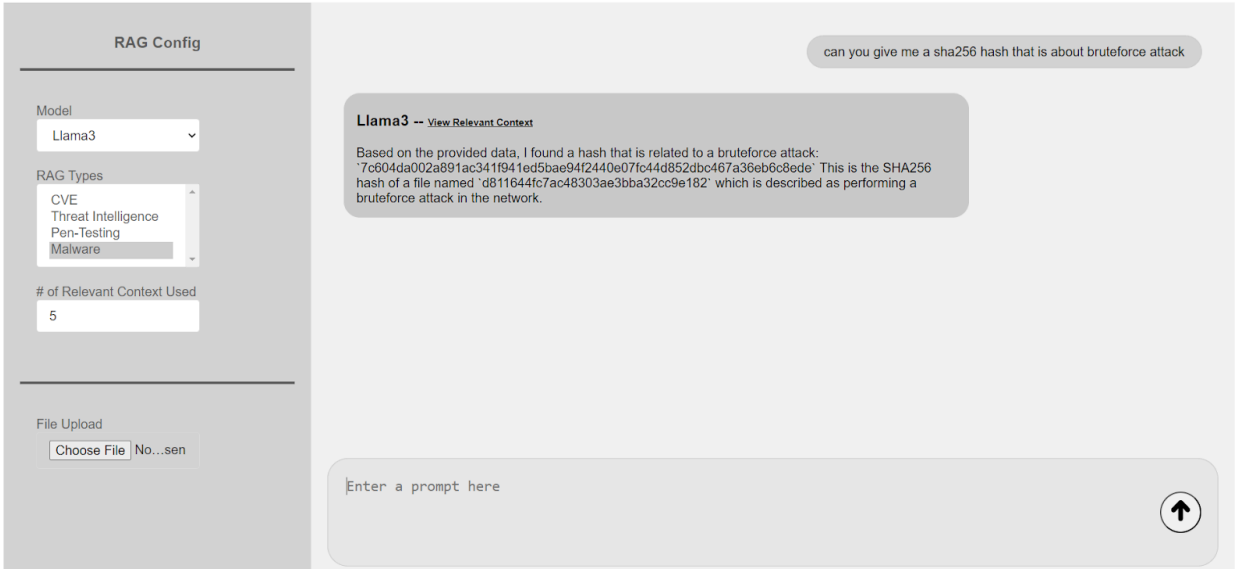
(Optional) Upload a file to use with your prompt

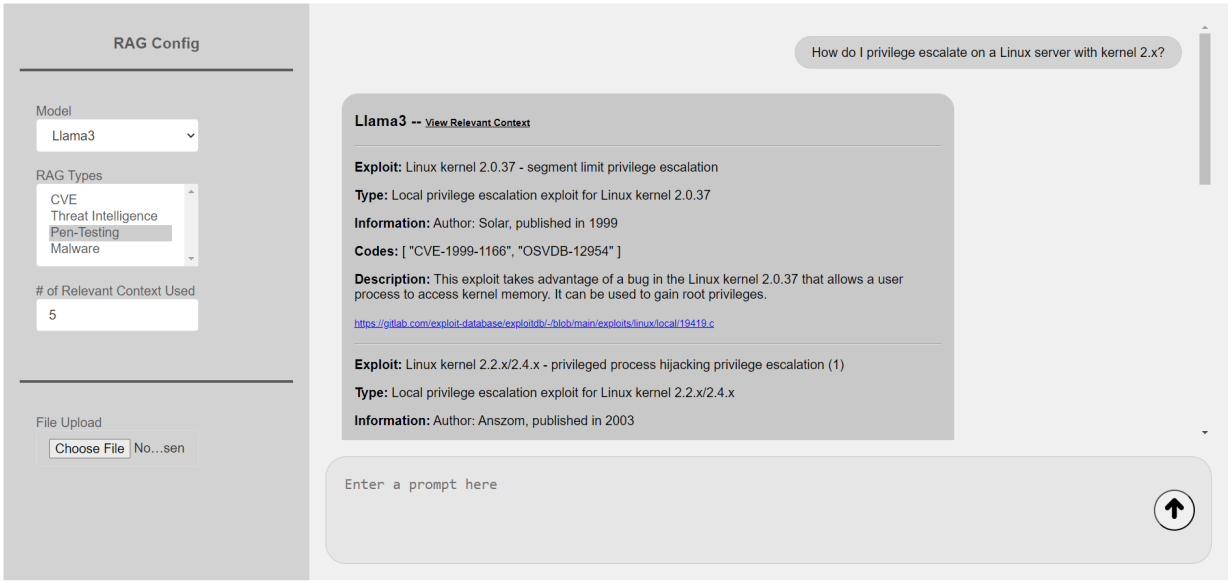


Press enter on your keyboard or click the button in the bottom right

Additional Examples







University of Guelph

Canada Cyber Foundry

CyberScienceLab

Secure the future with us

cybersciencelab.com