Assignment Report – Linux Groq Chatbot

Course: Ethical Hacking

Name: ALLAN PREM VARGHESE

Roll Number: 2462030

Index:

1. Introduction:

Linux is an extremely prevalent operating system in development and cybersecurity environments. Though it is full of capable command-line powers, new users tend to find Linux commands confusing because of technical terms. This project employs the Groq API to develop a Command-Line Interface (CLI) chatbot that translates Linux commands into simple, beginner-accessible language. The chatbot serves as an interactive tutor, lessening the necessity for finding documentation or tutorials online.

2. Objective:

- The primary goals of this project are:
    - ➢ To create a CLI chatbot in Python and Groq API that describes Linux commands.
    - ➢ To simplify complex commands so they are easier to understand for new users.
    - ➢ To incorporate error handling for seamless user experience.
    - ➢ To investigate how AI can be used for cybersecurity education.

3. Tools & Technologies Used:

- Programming Language: Python 3
- API Service: Groq API (Model: llama-3.1-8b-instant)
- Libraries: requests (for HTTP requests)
- Operating System: Kali Linux (using virtual environment)
- Hardware: 2 GB RAM minimum, Internet connection

4. Implementation Steps:

- **Step 1:** Get Groq API Key

  - Go to: https://console.groq.com/keys

  - Login and create your API key

- **Step 2:**

  Install Dependencies:-

  pip install requests

- **Step 3:**

  Write the Python CLI Bot (≤ 15 lines):-

```
import requests
GROQ_API_URL = 'https://api.groq.com/openai/v1/chat/completions'
headers = {'Authorization': 'Bearer YOUR_API_KEY'}
system = 'Explain any Linux command in plain English for beginners.'
while True:
cmd = input(' Enter Linux command: ')
    if cmd.lower() in ['exit', 'quit']:
        break
    data = {
        'model': 'llama-3.1-8b-instant',
        'messages': [
            {'role': 'system', 'content': system},]}
{'role': 'user', 'content': f'What does {cmd} do in Linux?'}
        ]
    }
    res = requests.post(GROQ_API_URL, headers=headers, json=data)
    try:
        print( res.json()['choices'][0]['message']['content'].strip())
    except Exception as e:
print(' Error:', e)
print(' Full Response:', res.text)
```

5. Problem Faced:

The bot initially employed the `mixtral-8x7b-32768` model, which got decommissioned. This led to a 'model_decommissioned' error. The fix was to swap it with `llama-3.1-8b-instant` as suggested by Groq documentation.

6. Output Examples:

Example Interaction:

-> enter Linux command: ls -l

-> Lists files with information such as permissions, size, and modification date.

-> Type Linux command: chmod +x script.sh

-> Adds execute permission to the file 'script.sh'.

7. Educational Benefits:

- Describes technical commands in simple English
- Inspires hands-on discovery of Linux
- Saves time by preventing long documentation searches
- Assists cybersecurity students in learning tools such as nmap, tcpdump, and netstat

8. Conclusion:

By doing this assignment, I gained knowledge about how to incorporate a third-party AI service (Groq API) into a Python command-line tool to provide a real-world solution simplifying Linux commands for new users. I also achieved hands-on experience in:

- Working with APIs: Making HTTP requests, authenticating, and processing JSON responses.
- Model selection & updates: Learning how model depreciation impacts projects and how to update to unsupported models.
- Error handling: Creating code that gracefully handles API errors and unforeseen responses.
- Practical Linux use: Consolidating knowledge of typical Linux commands during testing of the chatbot.
- Cybersecurity relevance: Observing how AI can be utilized as a training tool for ethical hacking and system administration.
- This project not only enhanced my coding and API integration knowledge but also provided me with experience of using AI as an educational resource for cybersecurity.