**A detailed report on NETWORK SECURITY CS 5342**

**PROJECT ROUND 2**

# **OnDemand Professor Q&A Bot**

**Ying Liu**

**(Associate Professor)**

Submitted by

**R11916865**

**Cherukuri Navyasri**

**GitHub Url :** [**https://github.com/cherupallybhanuteja/Q-A\_Bot\_NetworkSecurity**](https://github.com/cherupallybhanuteja/Q-A_Bot_NetworkSecurity)

# **Project Objectives:**

Our goal is to develop a Q&A Bot designed to provide information and answer questions about network security courses. To ensure data privacy, we focus on using open-source alternatives to ChatGPT that can run directly on users' devices. This approach allows all data, including course information and the Q&A model, to remain securely stored and processed locally. By handling everything on-device, we avoid the need to transmit information over the internet, thereby minimizing privacy risks.

# **Tools & Technologies:**

**Programming Languages**:

**Python**: It is one of the most widely used backend languages today, largely due to its strong compatibility with AI libraries and data processing tools.

**JavaScript, HTML, and CSS**: Owed to the need of an attractive and responsive front end, this is also used for the development of web applications.

# **User Interface:**

**Flask**: It handles web server configurations, requests, and backend processes, while managing frontend routing to facilitate smooth question-and-answer interactions with the bot.

**Vector Database:**

**Chroma Database**: Well indexed documents stored as embeddings in a database to facilitate fast document search based on similarity and to ensure relevant course material is retrieved.

# **Natural Language Processing (NLP):**

**Hugging Face Embeddings**: Used to generate document embeddings that are quite a form of ranking-efficient class of deep convolution dynamics as well as attention based word similarities which can be utilized to assist the bot in retrieving topically related information.

**Groq API (Llama Model)**: Used in creating language model, that is, in creating given questions and providing corresponding answers. There are certain chat settings such as temperature and max\_length aimed at enhancing the output, that is, the output accuracy.

# **Document Handling:**

**PyPDFLoader**: Comprises some functions and kernel classes for Reader. Additionally, it includes functions like ConvertToPDF methods that enable simple reading and indexing of a document.

**PyPDF2**: enhancement for manipulation and parsing of clear text in PDF files. It decodes and retrieves text from a provided source document and prepares to save it in AudioManager.

# **Embedding and Retrieval Framework:**

**Sentence-transformers/msmarco-distilbert-base-v4**: A service implemented by Hugging. Provides a service where it is enabled to convert the textual content of a document into vectors thereby making it easy for the retrieving of similar documents.

# **Error and Performance Management:**

**Logging and Console Monitoring**. Monitor errors with the help of backend splurging and JavaScript console. It becomes more critical when dealing with JavaScript, HTML tags and requests on the net.

**Optimization Techniques**: For instance, in managing the surface of the Chroma database and working with large document sets. It results in effective utilization of data processing practices.

# **Documentation:**

An interactive search bot is trained on documents focused on network security and is designed to deliver fast and accurate responses to users when they ask questions related to network security.

I provided five sample prompts, along with screenshots of the bot’s responses, complete with citations. Additionally, we included snapshots from Wireshark packet tracer to analyze the incoming and outgoing network packets.

**PROMPT 1: What is the Criteria to Evaluate AES in short.**

A screenshot of a computer

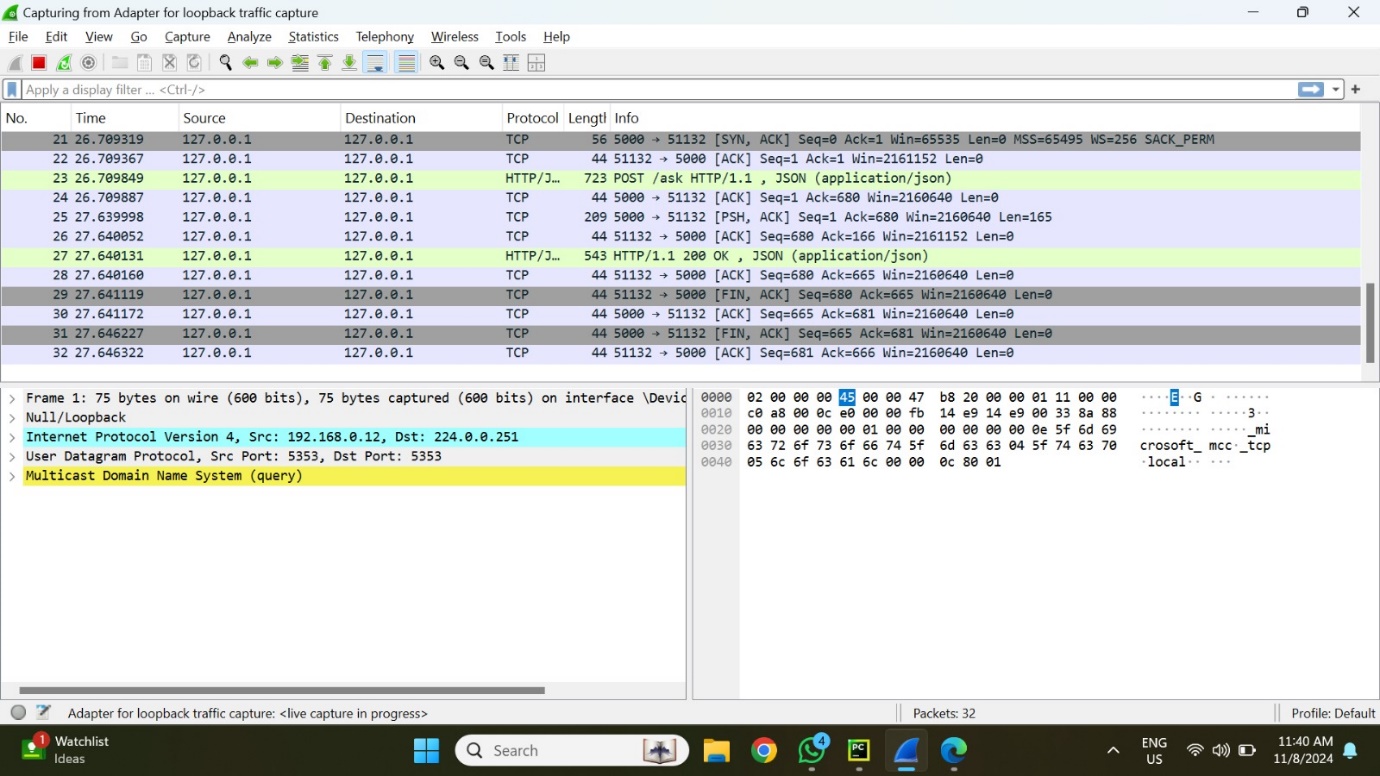
Description automatically generated

Here the prompt given by user is:

**PROMPT 1: What is the criteria to evaluate AES in short?**

When the specified command initiates the process in the web-based client, the chatbot—a software tool designed for conversation—replicates the operation on the local system. In the initial processing stage, the query intended to retrieve patterns from the database pattern base, Chroma DB, is encoded. During the actual retrieval stage, matched vectors activate the language model, ChatGPT.

Throughout this process, Wireshark can be used to observe the interaction, showing source addresses, content, incoming requests, and related TCP packets in its captures. The chatbot operates on localhost (127.0.0.1), with the client application running on port 8000. When the command is sent through local port 5000.



In Wireshark, we can view all details related to the prompt request, including the source and destination ports, header information, and the timestamp of when the packet was sent.

It uses the TCP protocol and sends an ACK packet back to the source.

A screenshot of a computer

Description automatically generated

The wireshark snapshot gives the information of the response sent to the user based on the prompt from the localhost 5000 port. The frame length is 56 bytes and protocol used is TCP.

**Protocol: TCP**

* **Source Port**: 5000
* **Destination Port**: 51132
* **Sequence Number**: 2366114572 (raw), displayed as 0 for relative numbering
* **Acknowledgment Number**: 1473486407 (raw), displayed as 1 for relative numbering
* **TCP Flags**: SYN, ACK (0x012)
* **Window Size**: 65535
* **Checksum**: 0x0d66 (unverified)

TCP Request Packet ------------------- Source IP: 127.0.0.1 Source Port: 5000 Destination IP: 127.0.0.1 Destination Port: 51132 Sequence Number: 0 Acknowledgment Number: 1 Flags: SYN, ACK Window Size: 65535 Data Length: 31 bytes Checksum: 0x0d66 (unverified)

TCP Response Packet -------------------- Source IP: 127.0.0.1 Source Port: 51132 Destination IP: 127.0.0.1 Destination Port: 5000 Sequence Number: 1 Acknowledgment Number: 1 Flags: ACK Window Size: 65535 Data Length: 0 bytes Checksum: 0x0f34 (unverified) # Hypothetical new checksum

The chatbot retrieves data using mapping vectors and feeds it into a large language model (LLM).

The LLM, based on the ChatGPT model, processes the embedding vectors and generates a response, which is then sent back to the user interface through the localhost on port 5000.

**PROMPT 2:Advantages of AES?**

A screenshot of a computer

Description automatically generated

The prompt given by user and response given is shown in the above snapshot.

The source of the information tells about the resource document which actually contains the prompt information in this case is the Stallings Text book.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The following information can be seen for the packet traced:

This contains the request sent through the user interface.

**PROTOCOL: Transmission control protocol (TCP) details:**

**Stream Information**

* **Stream Index**: 0
* **Stream Packet Number**: 1
* **Conversation Completeness**: Incomplete, with SYN\_SENT status, indicating that this is the initial SYN packet of the TCP handshake, awaiting a SYN-ACK from the destination.

TCP SYN Packet -------------------- Source IP: 127.0.0.1 Source Port: 51148 Destination IP: 127.0.0.1 Destination Port: 5000 Sequence Number: 8 Acknowledgment Number: 0 Flags: SYN Window Size: 65535 Checksum: 0x49c1 (unverified) Options: Maximum Segment Size, No-Operation (NOP), Window Scale, Timestamp

The LLM which uses chatgpt model receives the resultant embedding vector generated using resource documents and the prompt and then it generates the response which is again sent to the localhost 5000 port back to the user interface.

**PROMPT 3:Explain when to use random numbers in short**

A screenshot of a computer

Description automatically generated

The prompt given by user and response given is shown in the above snapshot.

The source of the information tells about the resource document which actually contains the prompt information in this case is the Stallings Text book.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The wireshark snapshot gives the information of the response sent to the user based on the prompt from the localhost 5000 port. The frame length is 56 bytes and protocol used is TCP.

TCP Layer Details

* Source Port: 51158
* Destination Port: 5000
* Sequence Number: 2975800837 (raw), displayed as 0 in relative numbering.
* Next Sequence Number: 1 (relative numbering)
* Acknowledgment Number: 0
* Header Length: 32 bytes
* Flags: SYN (0x002), signaling the start of a new TCP connection.
* Window Size: 65535 (maximum window size)
* Checksum: 0xcc26 (unverified)
* Urgent Pointer: 0

This packet initiates a new TCP connection with a SYN flag. Once a SYN-ACK is received from the destination (port 5000), the TCP handshake will proceed.

The resultant data packet is sent to OPENAI embedding which generates a resultant vector based on prompt and the resource documents vectors present in database. It is then feed into the LLM.

The response is sent by the LLM to the USER INTERFACE from port 61920 port of localhost to 5000 port. The packet information is also given in the form of hexadecimal representation which can be seen above.

**PROMPT 4:what are the type of random numbers?**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The wireshark snapshot gives the information of the response sent to the user based on the prompt from the localhost 5000 port. The frame length is 56 bytes and protocol used is TCP.

TCP Layer Details

* Source Port: 51162
* Destination Port: 5000
* Sequence Number: 3326875457 (raw), displayed as 0 in relative numbering.
* Next Sequence Number: 1 (relative numbering)
* Acknowledgment Number: 0 (as expected for an initial SYN packet, no acknowledgment yet).
* Header Length: 32 bytes
* Flags: SYN (indicated here as exe82)
* Window Size: 65535 (maximum possible window size)
* Checksum: 0xbdf9 (unverified)
* Urgent Pointer: 0

The response is sent by the LLM to the USER INTERFACE from port 61920 port of localhost to 5000 port. The packet information is also given in the form of hexadecimal representation which can be seen.

**PROMPT 5:What are the properties of random numbers**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The prompt is sent to embedding which generates a resultant vector based on prompt and the resource documents vectors present in CHROMA database. It is then fed into the LLM.

The wireshark snapshot gives the information of the response sent to the user based on the prompt from the localhost 5000 port. The frame length is 56 bytes and protocol used is TCP.

TCP Layer Details

* Source Port: 51164
* Destination Port: 5000
* Sequence Number: 4105576165 (raw), shown as B (relative sequence number).
* Next Sequence Number: 1 (relative numbering)
* Acknowledgment Number: 8 (acknowledges nothing yet as this is the SYN stage)
* Header Length: 32 bytes
* Flags: SYN (0x002), indicating a connection initiation
* Window Size: 65535 (maximum window size)
* Checksum: 0x87e9 (unverified)
* Urgent Pointer: 0

The response is sent by the LLM to the user interface from port 61920 port of localhost to 5000 port. The packet information is also given in the form of hexadecimal representation which can be seen above.

Overall, this paper provided a comprehensive analysis of the network security model and essential processes required for establishing secure communication channels. By examining samples of TCP traffic, it explored the role of protocols like the TCP handshake in ensuring data integrity, reliability, and sequential delivery within local networks. Additionally, the study delved into the network security framework, focusing on protocol development, secure data transmission, key generation, and algorithm design. Together, these elements contribute to the implementation of robust security measures that protect data during transmission.