Cuckoo Watchtower

Domenic Lo Iacono
Department of Computing Security
Rochester Institute of Technology
Rochester USA
dml3483@rit.edu

1 INTRODUCTION

Cuckoo Watchtower is a Large Language Model reporting and analysis tool utilizing GPT-4 from OpenAI. CW (Cuckoo Watchtower) is perfect for taking in the large amount of data generated from Cuckoo and displaying it dynamically in a chat-bot style. The application does this by gathering data from the json report generated by Cuckoo and creating smaller but still comprehensive text files on the general, signature, network, and static analysis performed. Once the text files have been created they are passed to the RAG pipeline. Retrieval Augmented Generation is the process of adding additional context to the Large Language Model so that it may answer questions more effectively or gain knowledge to new information. In this case, the new information is the text files that contain the various results of the malware analysis. All of this information is stored as vectors inside of a vector database called Weaviate. Weaviate is an open-source, cloud-native vector database that provides advanced search capabilities through the use of machine learning models.

Following the process of the RAG pipeline, the model is then given an important system message.

The message is as follows: "You are an assistant for malware event analysis. If the user asks questions that are completely unrelated to malware event analysis refuse to answer their question and state that your purpose is to answer questions related to malware and analyze the malware event data. If you don't know the answer, just say that you don't know. Question: question Context: context Answer:"

This system message is extremely important as it sets what role the model will take which in this case is an assistant for malware event analysis. Another important aspect of this system message is the inclusion of the question which is the users query, and the context which is the information passed in through the RAG pipeline that is retrieved as a user asks a question.

Finally, the user is prompted to ask a question. For this tools purposes questions that revolve only around the context given, the context being the text files provided, are suitable to be asked. The system message also instructs the model to not answer unrelated questions which keeps the results on track and more consistent.

Conference'17, July 2017, Washington, DC, USA
© 2024 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 USE CASE

The use case for this tool is well defined while also having room for the freedom of expanding the complexity of the queries. The most intuitive use case is asking for a comprehensive report on the program analyzed. CW preforms this task extremely well providing a response that is easy to read and includes specific information passed in through the RAG pipeline. Furthermore, by adding additional stipulations on report format, being clear on the audience the report is meant for, length requirements, and more a report can be created for almost any situation.

Another intuitive use case for this tool is simply asking the model for more information or asking more clarifying questions. For example, if a signature was detected that the analyst/user was unfamiliar with the user could ask the model for more information.

Finally, another use case that can speed up the efficiency of an analyst's job is the ability for the model to assist in pivoting. Because the model can display and search for certain information much faster than a human can pivoting and conducting an investigation can be significantly sped up.

3 LIMITATIONS

As useful as this tool can be there are certain limitations to be aware of. Firstly, the model only has as much information about the program's analysis that is passed into it through the RAG pipeline. Unfortunately, passing all of the data collected during the process of analysis is impossible. In one example the json file that was utilized for the testing of the application had over 700000 lines. OpenAI limits the requests size limit for embedding the information passed through the RAG pipeline and the embedding of information can also be expensive based on which model is utilized. This limitation was dealt with within the application by utilizing multiple functions that take out the most important information.

Another limitation to be aware of relates to if this application were to be used outside of a testing environment. The nature of the sensitive information could require that OpenAI's policies on using chat/training data may require for a different model to be utilized for the application. Utilizing a different model such as Llama, BLOOM, and other open-source models on a private server or system is possible and would still allow for the utilization of the RAG pipeline and similar capabilities.

4 FUTURE WORK

As the accessibility to data is the biggest limitation to the tools usefulness future work for this application would revolve around adding additional data sources from Cuckoo into the RAG pipeline. Providing memory snapshots, adding bson files that relate to each PID observed in the analysis, and more could be added. However, carefully observing what impact the additional information has on the models responses and quantifying if the responses are better, worse, or accurate is a significant challenge.

Another interesting but much more advanced way to increase the accessibility to data would be giving the ability to run Cuckoo scans to the model through integration with a code execution environment. This would then increase CW's usefulness in pivoting and analysis as it would allow for a user to streamline the process of running additional analyses.

5 CONCLUSION

In conclusion, Cuckoo Watchtower is an application utilizing a LLM to accelerate the malware analysis process and has proved to be effective at improving the efficiency of malware analysis. Although dependent on the data collected during the initial analysis, the ability to create readable, accurate reports that can be customized to their audience is highly useful. The ability to assist in the pivoting and investigation related to the malware sample is also an important role of the application. Lastly, the room to grow and improve the tool as discussed in the future work section is clear.

REFERENCES