LLM-Based Method to Generate Vulnerability Exploitation Steps from NVD Vulnerability Descriptions

Introduction

This project aims to automate the generation of exploitation steps for cybersecurity vulnerabilities using Transformer-based language models (LLMs). By using the National Vulnerability Database (NVD), the project focuses on mapping CVE descriptions to execution flows. The proposed solution reduces manual efforts, mitigates risk, quickly understands the exploitability of vulnerabilities, and makes systems safer and more secure.

What Has Been Done So Far

1. Initial Dataset Mapping

- Started with two files:
  - A CSV file containing CAPEC-to-CWE mappings (one-to-many).
  - A JSON file containing CVE-to-CWE mappings (one-to-many).
- Used CWE as a common link to map CVEs to CAPECs, resulting in multiple CAPECs for a single CVE.
- To make each row unique, combined CVE IDs (e.g., `CVE-2023-0001`) with CAPEC IDs (e.g., `391`), resulting in unique identifiers like `CVE-2023-0001_391`. These were not used further.
- Extracted all techniques that could exploit the vulnerability and used placeholders (`t1`, `t2`, `t3`, etc.) for each technique. These placeholders were later replaced with actual techniques in the execution flow column of the dataset but were not utilized further.

2. Relevant CAPEC Mapping

- Reviewed research papers to identify techniques for mapping CVEs to CAPECs.
- Learned about text preprocessing techniques to implement the approach from the paper *"Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques"*.

- The paper used cosine similarity for matching CVEs to CAPECs across their entire corpus.
- Adapted this approach by mapping the best possible CAPEC ID for a given CVE ID based on the highest similarity score from multiple possible CAPEC IDs.
- Also mapped CVEs to CPEs (Common Platform Enumerations) for additional context.

3. Learning

- Studied the basics of:
  - Deep learning.
  - Transformers.
  - Artificial Neural Networks (ANNs).
  - Natural Language Processing (NLP) for better understanding of model implementation.

4. Model Selection

- Used T5 Small model because it is pre-trained on tasks like document summarization, text-to-text generation, and question answering.
- Chose T5 Small as the initial model for generating execution flows from CVE descriptions.

5. Fine-Tuning T5 Small Model

- Fine-tuned the T5 Small model on the dataset created.
- Used BLEU score to validate and check the training direction.
- Generated two graphs:
  - **Loss vs. Epoch:** Monitored the reduction in training loss over epochs.
  - **BLEU Score vs. Epoch:** Measured how well the model performed in text generation over the epochs.
- Achieved a BLEU score of **0.34** after training T5 Small for 50 epochs.

6. Fine-Tuning T5 Base Model

- Experimented with the larger T5 Base model to improve performance.
- Fine-tuned T5 Base for 50 epochs and achieved a BLEU score of **0.54**, a significant improvement over T5 Small.
- This BLEU score indicates good alignment between the generated execution flow text and the reference text.