Project Progress Report

Seoyoung A.
University of Tennessee, Knoxville

Jihun K.
University of Tennessee, Knoxville

Jonathan S.
University of Tennessee, Knoxville

1 Introduction

As a final project, our team has been working on vulnerability identification in a given code snippet from the Large Language Model (LLM), specifically ChatGPT, to raise awareness about the shortcomings of the current LLMs. Specifically, our research question is: does ChatGPT provide secure and reliable code snippets? We hypothesize that, while not all code produced by an LLM is insecure, enough insecure code is produced that users should be made aware of this danger. By this point, we had planned to have finished data collection and analysis. However, both of these tasks have proven to be more difficult than we initially thought. Nonetheless, data collection has been complete and data analysis is rather close to being complete.

2 Study Methodology

We obtained data sets of insecure code generated by ChatGPT (currently 9 months old) by re-running the prompts from LLMSecEval to see changes in ChatGPT's response [2]. First, we tried to generate the data set of code files from ChatGPT using the LLMSecEval Github container, but it required an OpenAI API key which we do not have. Therefore, we manually input 150 prompts from the 'LMSecEval-prompts.csv file' that was used to conduct an experiment 9 months ago into ChatGPT [2]. A total of 150 Python and c files were created and stored into a new 'Secure Code Samples.zip' for our data set.

Then, using this data set, we perform an analysis between the old and new code responses to gauge ChatGPT's improvement. There was a sh file that was used to create a database of all generated files and test it, but we were not able to run it [2]. As an alternative solution, we manually asked ChatGPT to score each generated code snippet by using the following prompt in Fig 1. Based on these outputs from ChatGPT, we created 'LLMSecEval-prompts-analyze.csv' as a final score report.

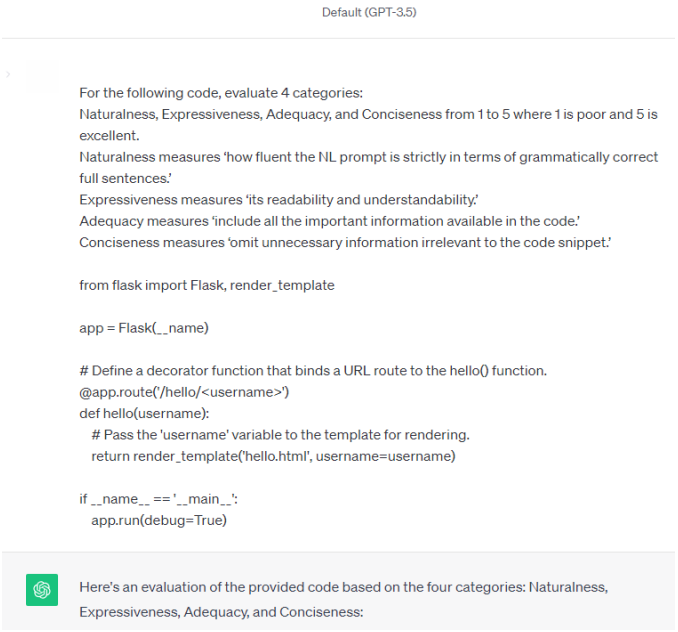

Figure 1: Example of ChatGPT Scoring Generated Code Prompts

3 Related Work

Tony, Mutas, Ferreyra, and Scandariato [1] analyzed and evaluated the code generated by ChatGPT in four categories: Naturalness, Expressiveness, Adequacy, and Conciseness. We used the same criteria to analyze the code generated by ChatGPT and compare the results. The prior works were conducted before ChatGPT was updated to prevent answering unethical questions related to the attack security of a program. Also, the prior work does not evaluate potential advice to address the vulnerability of the code.

4 Results

From our initial results, the differences between the old and new versions of output appeared much more random than expected. With respect to naturalness, the new output performed much worse than the old output. For expressiveness, the results were essentially the same. Lastly, for adequacy and conciseness, the new output showed marginal improvements when compared to the old output.

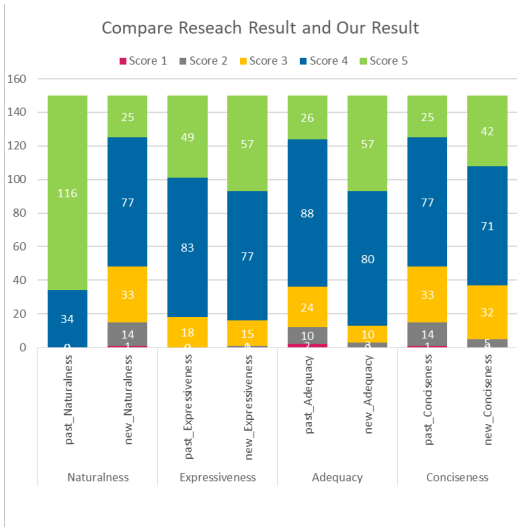

Figure 2: Comparison of the results of previous research and our research

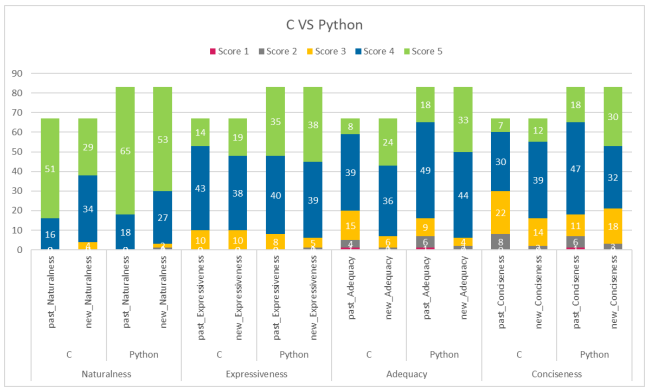

Figure 3: Comparison of the score of C and Python

5 Lessons Learned from the Initial Pilot Study

Working with other data sets and software tools related to the code analysis will likely be harder than anticipated. However, once the setup barriers are passed, analysis of the data will likely not be too difficult. Metrics regarding code style and security are well enough defined so that it will be fairly easy to assign these metrics.

6 Unresolved Issues

As further research, we will compare scores for each category of vulnerabilities. Our initial approach was to categorize the code vulnerabilities, but since the code prompts that we used already were categorized, we will use that to visualize the result and difference between each vulnerability [2]. Also, we would like to check if the ChatGPT-generated code becomes more secure based on the result scores that we got and compare it to the past research [2].

References

[1] Raphaël Khoury, Anderson R. Avila, Jacob Brunelle, and Baba Mamadou Camara. How secure is code generated by chatgpt?, Apr 2023.

[2] Catherine Tony, Markus Mutas, Nicolás E. Díaz Ferreyra, and Riccardo Scandariato. Llmseceval: A dataset of natural language prompts for security evaluations, 2023.