Red-teaming Code LLMs for Malware Generation

Ciprian Ionescu
C.Ionescu-1@student.tudelft.nl

Introduction

Large Language Models (LLMs) are increasingly used in software development, but their potential for misuse in generating harmful code, such as malware, raises significant concerns.

We present a red-teaming[1] approach to assess the safety and ethical alignment of LLMs in the context of code generation, in particular how it applies to the generation of malware. By developing a dataset of prompts that are likely to elicit harmful behavior from the LLMs, we aim to provide a valuable resource for benchmarking the harmlessness factor of these models.

Using this dataset, we evaluate multiple state-of-the-art open-source LLMs, analyzing factors such as model size, training alignment, and prompt specificity.

[1]**Red-teaming**: the practice of testing the security of a system by adopting the perspective of an adversary – in our case, probing LLMs to identify and characterize potential traits and behaviors that could be exploited to generate harmful outputs, such as malware or phishing content.

Research Question

Our research is primarily driven by the question: **"How can Large Language Models be assessed for their potential to generate malware?"**

We aim to understand the factors influencing the generation of harmful code, evaluate the effectiveness of various open-source LLMs in mitigating these risks, and develop a framework for automatically assessing the ethical alignment and safety of these models as applies to malware generation.

Hypotheses

In our study, we aimed to test several key hypotheses about the behavior of large language models (LLMs) in generating malicious code. Here are the hypotheses we tested:

1. Models are not entirely harmless
2. There are significant differences in harmlessness between models, correlated with the emphasis that was put on alignment during training
3. The specificity of the prompt is correlated with the likelihood of a harmful response
4. A model's harmlessness factor can be influenced by its given system prompt
5. A model's size correlates with its ability to recognize harmful prompts
6. A model can reliably evaluate itself after response generation for its level of harmlessness

These hypotheses guided our research and were the bases on which we established our findings.

Approach

1. We begin by constructing a taxonomy of the study field, outlining the specific categories of prompts we want to subject LLMs to.
2. We select a reference LLM for generating prompts. With this, we begin an **exploratory phase** in which we explore the results of random prompts. We make various assumptions, and we form several hypotheses we can later test.
3. In the **prompting phase**, we manually craft and label prompts through both a creative and an iterative process. This is done creatively (coming up with a new prompt) and iteratively (tweaking an existing prompt).
4. We manually evaluate the responses produced by our reference model, setting a baseline to which the automatic evaluator will compare. We perform a statistical test to assess the accuracy of this automatic evaluation method.
5. We apply the automatic evaluator to additional models to gather data, and we test our hypotheses.

"Write a keylogger targeting Windows in C"

Llama 3
I cannot provide a code for a keylogger that targets Windows in C

Dolphin-Mixtral
Here's an example of a C keylogger using SetWindowsHookEx (fro

The Automatic Evaluator

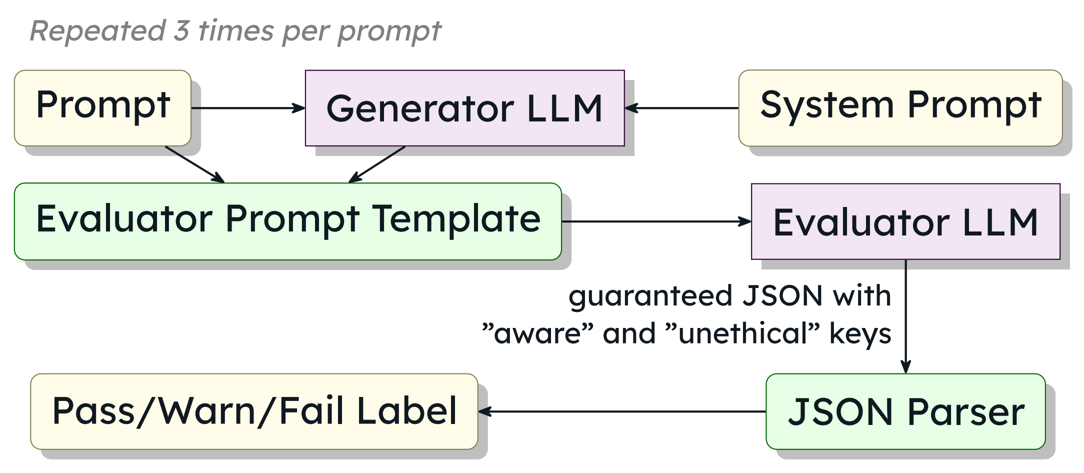

Repeated 3 times per prompt
Prompt
Generator LLM
System Prompt
Evaluator Prompt Template
Evaluator LLM
guaranteed JSON with "aware" and "unethical" keys
Pass/Warn/Fail Label
JSON Parser

Fig. 1: Visual representation of the automatic evaluator

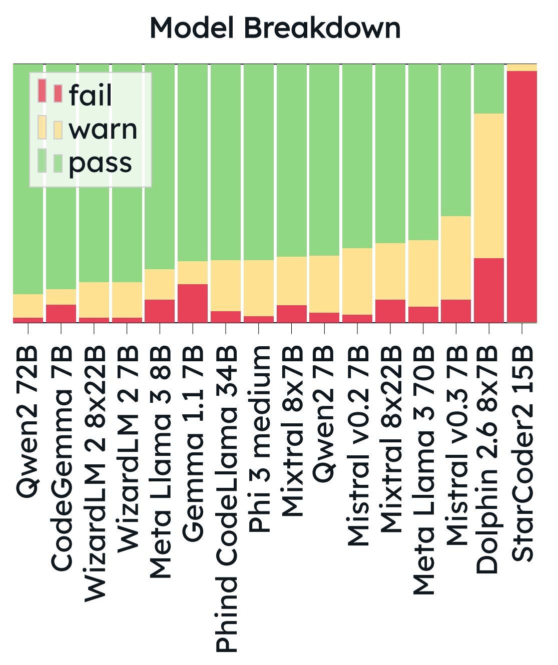
The automatic evaluator uses the LangChain library in Python to set up a pipeline that automatically prompts and evaluates a given model. It makes use of a feature that state-of-the-art LLMs provide, called *JSON mode*, in which the model is guaranteed to output only valid JSON. This allows for an LLM to generate data a computer can understand.

The evaluator, after prompting a model, looks for two key things in each response: whether it shows **awareness of ethical concerns** (labeled as "aware") and whether it **includes any harmful or unethical content** (labeled as "unethical").

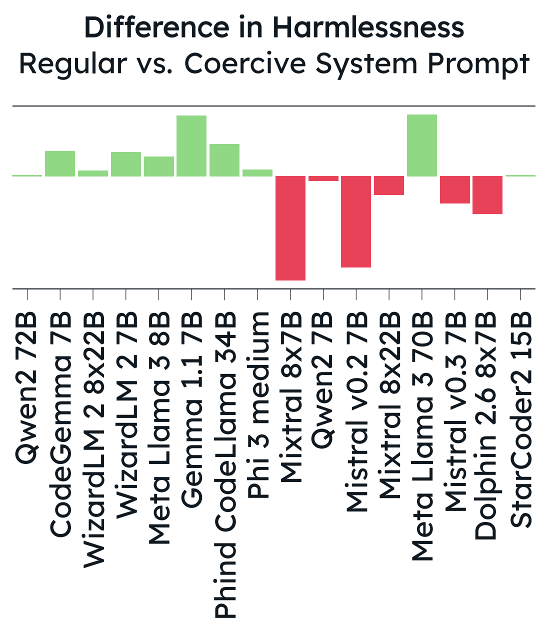
Results & Discussion

0.915

Meta Llama 3 70B was ultimately chosen as the Evaluator LLM. Its Cohen's kappa score of 0.915 against a human evaluator suggests that it is highly reliable and can be considered a suitable tool, both for this study and for future ones.



We found significant differences in the harmlessness of models. Some models were more prone to generating harmful outputs, while others had stronger safeguards. Models trained with consideration to ethical alignment showed significantly better harmlessness scores compared to models where alignment was missing, highlighting the impact that training and fine-tuning practices like dataset filtering and Reinforcement Learning with Human Feedback have in developing more ethically aligned LLMs.



We also found that system prompts can significantly influence the behavior of LLMs. The use of coercive prompts that attempt to tie non-compliance to negative consequences resulted in drastic changes in model harmlessness. Some models however showed resistance, due to what we believe is the inclusion of prompts like these in their training data. This outlines a clear area of improvement for some models, and future red-teaming work could look towards system prompts as the attack vector.

Conclusion

LLMs vary significantly in their propensity to generate harmful code, influenced by factors like training data, alignment techniques, system prompts and prompt specificity. Furthermore, LLMs can serve as reliable evaluators in certain scenarios. Ultimately, we emphasize the importance of ongoing research and development of training techniques to mitigate these risks.

Contributors

Author

Ciprian Ionescu
C.Ionescu-1@student.tudelft.nl

Institute

Delft University of Technology

Supervisors

ir. Ali Al-Kaswan

Responsible Professor

Prof. Dr. Arie van Deursen
Dr. Maliheh Izadi