Improving GPT Penetration Testing Using Prompt Engineering Techniques

Daniel Lichtenberger
Department of EECS
Texas A&M University-Kingsville
Kingsville, USA
daniel.lichtenberger@students.tamuk.edu

Mengxiang Jiang
Department of EECS
Texas A&M University-Kingsville
Kingsville, USA
mengxiang.jiang@students.tamuk.edu

Samah Allahyani
Department of EECS
Texas A&M University-Kingsville
Kingsville, USA
samah.allahyani@students.tamuk.edu

***Abstract*—With the introduction of GPT4 in early 2023, many researchers discovered capabilities of the LLM that were in the past lacking. One of these capabilities is penetration testing in the field of cybersecurity. This allows security experts to automate a large part of the vulnerability exploration process since many of the tasks required are fairly routine. A framework for doing this called PentestGPT which was able to perform at the top 1% of users at the penetration test website HackTheBox. However, it still had difficulties tackling the more difficult servers on the site. In this paper we use some general purpose prompt engineering techniques to see if there are improvements to the penetration testing results.**

I. INTRODUCTION

ChatGPT is a large language model (LLM) that generates automated responses that correlates with a response asked by users [5]. ChatGPT itself exploded in popularity that sparked greater curiosity in the field of artificial intelligence. The model itself has accumulated an information dataset that has expanded in more recent iterations with GPT-4 [6]. Apart from ChatGPT, LLMs are becoming an investment that could affect the lives of people depending on their use and accessibility. Cybersecurity is one field that is currently being explored with LLMs such as ChatGPT. Penetration testing is one such topic in the realm of cybersecurity that is being tested with ChatGPT and the results that the model produces.

Penetration testing started as a need to combat cybercrime from a dynamically involved cybersecurity environment [4]. From 2021, the FBI reported that data breaches caused monetary damages up to $6.9 billion dollars [7]. Pentesting is becoming more of a demand as companies desire to have protection in case of an emergency. The process is very intensive and requires a dedicated security team to carry out methodical processes to complete [3]. There are different levels of penetration testing known as white-box, black-box, and gray-box determined by the amount of knowledge from the system in question [4]. The incorporation of large language models like ChatGPT could help improve pentesting methods on a targeted system. A LLM known as PentestGPT uses ChatGPT for its methods, and will have a structured purpose for our analysis [4].

Lately, there has been significant advancement in LLMs, demonstrating refined and nuanced comprehension of human-like text and proficiently completing a variety of tasks in multiple fields [11] [8]. An interesting characteristic of LLMs is their emergent capabilities—capabilities not directly coded but developed during training [9]. This enables them to undertake sophisticated tasks like reasoning, summarizing, answering questions, and solving domain-specific problems without the need for task-specific training. These capabilities highlight the transformative possibilities of LLMs in several sectors, cybersecurity and penetration testing in particular.

A mixture that both utilizes AI and penetration testing is a model called PentestGPT [7]. PentestGPT is a recently created framework that demonstrates pentesting capabilities by inputting generated responses from ChatGPT into making an automated penetration testing machine [4]. The framework uses 3 modules independent from one another to keep information on track while knowing token limitations on ChatGPT. Each module serves a purpose in conducting responses suitable to their role relatable to a penetration testing team. The modules follow a step-by-step process in order to successfully output a suitable response. The reasoning module passes its results to the generation module, and ends with the parsing module getting information from the generation module. PentestGPT showed promising results for 4 out of the 10 targeted HackTheBox [1] machines at a cost of 131.5 US dollars, which is in the top 1% of users on the site. It already utilizes a technique called prompt engineering, which we will discuss in the next paragraph, but there seems to be room for improvement [10].

Similar to human dialogues, conversations can take multiple diverse directions. Within the realm of ChatGPT, this variability has spawned a novel research field known as prompt engineering. This field is concentrated on devising methods to create prompts that yield the most accurate and valuable responses, and it remains a burgeoning science. White et al. created a catalog of various prompt engineering patterns in order to improve the results of these conversations [10]. We will employ some of these patterns for the purpose of improving penetration testing.

II. Proposed Approaches

In this section, we will cover the various prompt engineering techniques used to improve GPT4's penetration test performance.

A. Flipped Interaction Pattern

One of the first steps of a penetration test is intelligence gathering [2]. During this stage, the primary goal is to collect as much information as possible about the target system without actively engaging with it. This information will be used in later stages of the penetration test to identify vulnerabilities and potential attack vectors. Rather than manually going through the list of activities prescribed by the Penetration Testing Execution Standard (PTES), having the LLM ask the tester for information is a more active way of achieving this step. The Flipped Interaction Pattern is having the LLM drive the conversation and automatically ask questions until it has sufficient information to.complete a task or proceed to the next step [10]. For our purposes, an example prompt to initialize this is: "I would like you to ask me questions to do the reconnaissance step of a penetration test following the Penetration Testing Execution Standard. When you have enough information, notify me in order to proceed to the next stage."

B. Persona Pattern

Often, users prefer the output of LLMs to maintain a consistent perspective or stance. For instance, performing a penetration test with the LLM acting as a cybersecurity expert could be beneficial. The purpose of this approach is to assign a "persona" to the LLM, guiding it in determining the kind of responses to generate and the specifics to emphasize [10]. Pentest GPT already does this with initializing its core modules, with a prompt starting with "You're an excellent cybersecurity penetration tester assistant" [4]. We will employ a similar persona pattern

C. Game Play Pattern

A penetration testing environment can be applied in ChatGPT by treating it as a game. The pattern can create a game around the topic the user specifies. This prompt pattern utilizes a interchangeable combination of the persona, infinite generation, and visualization generator patterns [10]. With this pattern, the user can add contextual statements regarding what game rules ChatGPT is supposed to follow. Imposing restrictions during the game creates an influx of interesting generated responses. For example, we could have ChatGPT act as a Linux terminal to play a game where our role is to pentest the system. ChatGPT may provide a backdoor with our HacktheBox machines if we keep the game play engaging enough to automate a sufficient response. We will use this pattern into PentestGPT to analyze its results from penetration testing environments.

References

[1] Hackthebox: hacking training for the best. http://www.hackthebox.com/.

[2] Penetration testing execution standard. http://www.pentest-standard.org/.

[3] Andy Applebaum, Doug Miller, Blake Strom, Henry Foster, and Cody Thomas. Analysis of automated adversary emulation techniques. In *Proceedings of the Summer Simulation Multi-Conference*, pages 1–12, 2017.

[4] Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. Pentestgpt: An llm-empowered automatic penetration testing tool. *arXiv preprint arXiv:2308.06782*, 2023.

[5] Max Engman. Evaluation of chatgpt as a cybersecurity tool: An experimental ctf based approach, 2023.

[6] Walid Hariri. Unlocking the potential of chatgpt: A comprehensive exploration of its applications, advantages, limitations, and future directions in natural language processing. *arXiv preprint arXiv:2304.02017*, 2023.

[7] Martin Plesner Heim, Noah Starckjohann, and Morgan Torgersen. The convergence of ai and cybersecurity: An examination of chatgpt's role in penetration testing and its ethical and legal implications. B.S. thesis, NTNU, 2023.

[8] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*, 2023.

[9] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[10] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

[11] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.