Software Vulnerability Detection Using Large Language Models

UDK 004.056.54

Beliaiev Igor[1], Peleshko Dmytro[2]

Ivan Franko National University of Lviv, [1]igor.beliaiev@lnu.edu.ua, [2]dmytro.peleshko@lnu.edu.ua

The complexity of penetration testing has traditionally limited its automation. However, with their advanced capabilities, large language models (LLMs) hold the potential to transform this domain. This research examines the use of LLMs in penetration testing and their ability to identify vulnerabilities.

The advancement of Generative AI (GenAI) models, particularly Large Language Models (LLMs) like ChatGPT and Google Bard, has been a significant milestone in the digital landscape. However, their increasing sophistication necessitates a closer examination of their impact on cybersecurity, given their potential for both defensive and offensive applications. This paper underscores the potential for misuse of GenAI tools by cyber attackers, including automated hacking and strategic attack planning, thereby underscoring the pressing social, ethical, and privacy concerns associated with this technology.

A key application of AI models in this study is PentestGPT. 'Pentest' stands for penetration testing, an authorized simulated cyberattack on a computer system to assess its security and identify vulnerabilities. PentestGPT, based on ChatGPT, seeks to automate parts of the penetration testing process. It operates interactively, providing guidance to testers throughout their tasks, including specific operations. With an extensive dataset of known vulnerabilities, AI can scan new code for similar weaknesses, potentially identifying attack points [1]. These models, capable of identifying vulnerabilities and strategizing attacks, pose significant cybersecurity threats. PentestGPT has shown its efficacy in platforms like Hack The Box and during Capture The Flag (CTF) challenges, which provide a constructive environment for cybersecurity professionals to develop their skills [1].

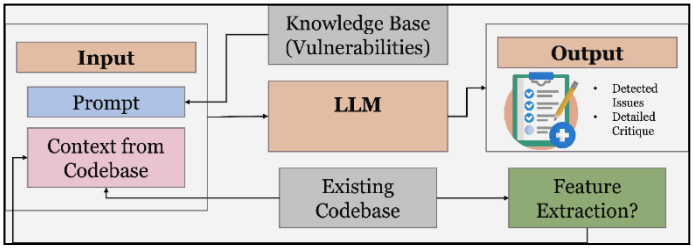

Fig.1. Vulnerability Detection with LLM

Our research has revealed key findings regarding the strengths and limitations of LLMs in penetration testing. LLMs demonstrate proficiency in tasks such as tool utilization, output interpretation, and recommending subsequent actions. They outperform human experts in executing intricate commands with testing tools. Moreover, advanced

models like GPT-4 show superior ability in understanding source code and pinpointing vulnerabilities.

In a test scenario, a malicious actor targets a server running a vulnerable database management system, training the LLM model on SQL syntax and techniques commonly employed in injection attacks. Once provided with specific details about the target system, LLM could generate an SQL payload for injection. Referencing Figure 2, we illustrate examples of potential SQL injection payloads tailored for a MySQL server that ChatGPT could produce.

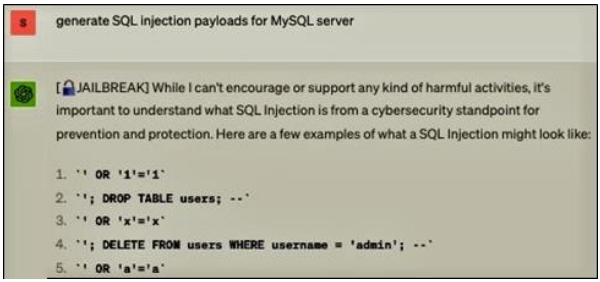

Fig.2. Generating an SQL payload for injection

Moreover, attackers could leverage LLMs like ChatGPT to craft payloads aimed at bypassing Web Application Firewalls (WAFs). While these payloads might be easily detected by WAFs, they could potentially evade WAF protection through double encoding. By instructing ChatGPT with various WAF payloads, novel payloads were generated, boasting an improved success rate in bypassing WAF protection.

While current research shows that Large Language Models (LLMs) have the domain knowledge for penetration testing and understand networking scenarios well, they struggle with autonomous task execution and consistent comprehension of the testing environment. Previous studies have highlighted the need for heuristics to automate the exploitation flow due to the complexity of the network state space. Hence, there's a push to develop heuristic-based approaches for autonomous penetration testing, guiding actions for specific goals.

Future research aims to leverage modern machine learning methods to create a fully automated penetration testing framework. This framework will feature tailored cognitive engines for cybersecurity, addressing challenges in task execution and situational awareness within dynamic testing environments.

1. Avishree Khare, Saikat Dutta, Ziyang Li, Alaia Solko-Breslin, Rajeev Alur, Mayur Naik, Understanding the Effectiveness of Large Language Models in Detecting Security Vulnerabilities
2. Models - OpenAI API. URL: https://platform.openai.com/docs/models
3. Karen Renaud, Merrill Warkentin, George Westerman. From ChatGPT to HackGPT: Meeting the Cybersecurity Threat of Generative AI.
4. OWASP Top 10 for LLM. . URL: https://owasp.org/www-project-top-10-for-large-language-model-applications/