Chapter 4
Revolutionizing Malware Detection With LLMs

Attila Magyar
Capitol Technology University, USA

Marwan Omar
Capitol Technology University, USA & Illinois Institute of Technology, USA

ABSTRACT

Malware detection remains a critical challenge in cybersecurity, necessitating innovative approaches to identify and mitigate threats. Recent advancements in natural language processing (NLP) offer promising avenues for improving malware detection systems. This chapter explores the application of GPT-2, a state-of-the-art generative pre-trained transformer model, to enhance malware detection. The authors propose a novel methodology leveraging GPT-2's capabilities to analyze and classify opcode snippets and textual features associated with malware. The approach involves fine-tuning GPT-2 on a diverse dataset of malware and benign software to learn distinctive patterns and characteristics indicative of malicious behavior. The experimental results demonstrate that GPT-2 achieves significant improvements in detection accuracy and reduces false positives compared to traditional methods. This study highlights the potential of integrating advanced NLP models with cybersecurity practices, providing a robust framework for future research in malware detection.

DOI: 10.4018/979-8-3693-9311-6.ch004

INTRODUCTION

In the ever-evolving landscape of cybersecurity, the identification and mitigation of malware stand as critical endeavors (Wang, et al., 2021). The presence of malware not only compromises the integrity and confidentiality of systems but also poses significant threats to data privacy and operational stability.

Malware detection has traditionally relied on signature-based methods and heuristic analysis (Santos et al., 2020). Signature-based methods, which involve matching known malware signatures against files or behaviors, are effective for known threats but fail to detect novel or polymorphic malware (Hafner et al., 2021). Heuristic methods, which use rules and behavioral patterns to identify malware, can be more adaptive but may suffer from high false positive rates (Alazab et al., 2017). As malware authors employ techniques like code obfuscation and polymorphism to circumvent these traditional methods (Kolb et al., 2019), there is a growing need for more robust detection mechanisms.

Recent advancements in machine learning (ML) and NLP offer new avenues for addressing these challenges. Models such as GPT-2, developed by OpenAI, have demonstrated impressive capabilities in understanding and generating human-like text (Radford et al., 2019). Although GPT-2 was originally designed for natural language generation, its underlying architecture and training paradigm may provide valuable insights into malware detection. GPT-2's ability to learn complex patterns and representations from large datasets could be leveraged to identify subtle anomalies and patterns indicative of malicious activity.

Preliminary research suggests that NLP techniques, including those used in models like GPT-2, can be adapted for cybersecurity purposes. For instance, recent studies have explored using language models for analyzing and classifying malware based on code snippets and execution traces (Li et al., 2021; Zhang et al., 2022). These studies indicate that NLP models can capture intricate patterns in opcode strings that traditional methods may overlook, potentially leading to more accurate and adaptive malware detection systems.

Malware detection methods have previously relied solely on signature databases, including malicious instruction patterns. The signature databases are used for matching against a signature generated from a newly encountered executable. Recently, the detection of malicious lines of code has been critical for the development of efficient malware detection. A significant challenge is that the source codes for executable files are not usually accessible in compiled form. Therefore, the assembly instructions are the best candidate to unveil malicious functionality in a suspected file (Demirci et al., 2022). In the present study, we focus on the disassembly of the executable file into opcodes, and feed the opcode strings into our GPT-2 model.