Web Application Firewall (WAF) using Natural Language Processing

Ajay Sai Moturi
Georgia Institute of Technology
amoturi@gatech.edu

Shriyanshu Kode
Georgia Institute of Technology
skode3@gatech.edu

Jayson Fong
Georgia Institute of Technology
jayson.fong@gatech.edu

1 Introduction

In the domain of cybersecurity, the rapid advancement and sophistication of cyber threats necessitate the development of equally advanced, adaptive, and intelligent defense mechanisms. Intrusion detection and prevention systems are hardware devices or software programs that continually monitor a network's activity, scanning for malicious or anomalous activity indicative of a cyber threat.

In practice, the most common IDS frameworks (NIDS, NNIDS, HIDS[1]) employ a combination of pattern matching techniques (searching for known attack patterns using regular expressions) and blacklisting of known malicious IP addresses. While these methods proved effective for many decades, recent years have given rise to zero-day attacks like Drive-by Download (DbD) and Spear Phishing, which utilize Exploit Kits to hide malicious content under standard communication protocols, going unnoticed due to the aforementioned traditional IDS practices.

Natural language processing techniques propose a unique opportunity to expand the capability of IDS operations beyond primitive string matching. Specifically, deep learning techniques have the potential to uncover behavioral patterns in malicious network activity over a time scale (LSTM), so HTTP requests can be processed and analyzed for anomalous or exploitative efforts in series (as opposed to one-at-a-time).

Most work combining ML methods with malicious traffic detection employ network logs, which summarize activity of HTTP requests that have already been allowed/disallowed. For example, (Das et al., 2020) utilized the word2vec (Mikolov et al., 2013) techniques of CBOW and skip-gram to generate word embeddings from network and proxy server logs; these embeddings were then used with traditional supervised binary classification methods (logistic regression, SVM, Naive Bayes, neural networks) to determine whether a given request was benign or malicious. Similarly, (Zolotukhin and Hämäläinen, 2013) utilized the N-gram model and k-means or single-linkage clustering techniques to detect anomalous HTTP requests based on network logs.

While both of these studies were able to produce precision and accuracy scores exceeding 0.99 on their respective test and training datasets, the produced classification models are far too large and slow for real-time use with modern broadband networks. When running on routers and edge servers, such an NLP-based IDS would need extremely low latency to minimize impedance on the network's throughput and performance. We propose a similar methodology of using word embeddings and network logs as a training dataset, but in combination with model techniques such as weight pruning, quantization, and knowledge distillation to optimize the model for inference time.

2 Motivation

Applying NLP to cybersecurity and threat detection presents a promising opportunity to interpret the structure, pattern, and time-related behavior of network requests that transcend the capabilities of looking purely at a request's content. This involves treating network packets as linguistic features, enabling a deeper understanding of traffic flow and a request's underlying intentions. The upside is clear, but there are innate challenges in building a secure system that operates at the edge. The need for rapid inference is paramount, as each network request must be evaluated without introducing performance

[1] *NIDS*: Network Intrusion Detection System, *NNIDS*: Network Node Intrusion Detection System, *HIDS*: Host Intrusion Detection System

bottlenecks or compromising the network's functionality. The motivation behind this project is two-fold: to enhance the capability of network security systems and ensure these enhancements are efficiently deployable at scale.

3 Problem Definition

Traditional traffic classification techniques for NIPS involve matching packets against known threat signatures; however, these methods lack the capability to defend against the exploitation of zero-day vulnerabilities due to the lack of a known signature.

There currently lacks research on the usage of natural language processing for intrusion prevention, instead focusing on detection using network logs (Das et al., 2020). We seek to focus on HTTP-based attacks such as SQL[2] injection, XSS[3], and general exploits involving mis-configured software. Through developing a NLP-based NIPS, we look to introduce a novel method for mitigating the threat of zero-day attacks for public-facing applications without necessitating the development threat signatures to supplement traditional NIPS.

4 Methodology

We anticipate deconstructing HTTP requests and responses into their defined syntactic components per (Berners-Lee et al., 2005), such as headers, paths, and queries. These components will then be tokenized, such as splitting queries into individual key-value pairs followed by further processing. These tokens will act as inputs into a language model to classify whether a request, response, or conversation, is likely benign or indicative of malicious activity.

Due to the time-sensitive nature of our application, our inference will utilize models that are small in size, to minimize additional latency introduced into IDS systems. We will experiment with fine-tuning pretrained small language models like TinyBERT, which "achieves more than 96.8% the performance of its teacher BERTBASE on GLUE benchmark, while being 7.5x smaller and 9.4x faster on inference" using a combination of data augmentation and distillation of BERTBASE's initial transformer architecture (Jiao et al., 2019). Since these pretrained models may not excel in our use case of network traffic classification, we will also experiment with building our own language models from scratch. This will entail experimenting with architectures like Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) models which can uncover time-series-related patterns in network threats. On these models, we anticipate employing weight pruning, quantization, and other techniques to optimize inference time for real-world applications.

5 Potential Results and Discussion

The model will be compared against baseline models, including logistic regression, support vector machines (SVM), and neural networks, and evaluated against ensemble learning methods, which represent the predominant approach in current NLP methodologies (Ahmed and Uddin, 2020).

The primary objective of the model is to classify incoming data into malicious or benign categories. To assess its performance, commonly used metrics such as accuracy, cross-entropy loss, precision, recall, and f1-score will be employed. Furthermore, to evaluate the model's suitability for edge computing, its complexity and efficiency will be analyzed. This analysis will involve measuring metrics such as inference time and the number of parameters. Additionally, other common metrics used to calculate model complexity, such as the number of Floating Point Operations (FLOPs) and Multiply-Accumulate Operations (MACs), will also be explored.

6 Feasability Analysis

Anticipated challenges for this project include addressing imbalanced datasets, where benign data significantly outweighs malicious data. A dataset recording attacks in an IoT environment (Neto et al., 2023) and datasets containing malicious web requests have been preliminarily identified, but further research will be done to aggregate additional datasets together. Further, implementing techniques to mitigate imbalances may be necessary.

Furthermore, fine-tuning the small language

[2] *SQL*: Sequential Query Language

[3] *XSS*: Cross-Site Scripting

model to the domain of intrusion prevention is another significant challenge. Specifically, optimizing the model through techniques like pruning and quantization to ensure real-time decision-making capability requires a thorough understanding of the model. It will also involve correctly setting hyperparameters tailored to the intended use case.

References

Maheli Ahmed and Mohammed Nasir Uddin. 2020. Cyber attack detection method based on nlp and ensemble learning approach. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–6.

Tim Berners-Lee, Roy T. Fielding, and Larry M Masinter. 2005. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986.

Saikat Das, Mohammad Ashrafuzzaman, Frederick T Sheldon, and Sajjan Shiva. 2020. Network intrusion detection using natural language processing and ensemble machine learning. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 829–835. IEEE.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani. 2023. Ciciot2023: A realtime dataset and benchmark for large-scale attacks in iot environment.

Mikhail Zolotukhin and Timo Hämäläinen. 2013. Detection of anomalous http requests based on advanced n-gram model and clustering techniques. In *Conference on Internet of Things and Smart Spaces*, pages 371–382. Springer.