Senior Capstone Final Proposal

Felix Childress
Department of Computer Science
Richmond, IN, USA
fdchild22@earlham.edu

Earlham College

Abstract

Intrusion detection systems (IDS) play a critical role in identifying malicious activity within network traffic, yet traditional methods often rely on models that largely struggle to generalize novel threats. This project explores the application of natural language processing (NLP) techniques, particularly that of transformer-based models such as BERT, to the domain of network packet analysis for intrusion detection. I propose a tokenization strategy that treats packet headers as structured sequences, enabling session-level behavior to be analyzed and learned from similarly to natural language. Utilizing different datasets, the performance of a fine-tuned BERT model is compared to that of a random forest baseline as a means of assessing the viability of NLP-driven approaches for cybersecurity applications as opposed to traditional methods. Core contributions include a customized pre-processing pipeline for converting packet data into BERT-compatible input, a comparison of traditional and NLP-based detection techniques, and an evaluation of how token structure and methodology influences learning outcomes.

1 Introduction

Recent years have witnessed an exponential rise in both the frequency and complexity of cyber attacks. As networked systems become more integrated into critical infrastructure, an increasing number of fields and disciplines have come to rely heavily on a continuous and secure exchange of data for operation. These systems depend on the smooth transmission of millions of network packets every second, each one carrying instructions, requests, or sensitive information. Malware targeting these systems, which often aims to exploit these packets, attempts to disguise itself within legitimate traffic, making it a difficult task to detect accurately. Traditional intrusion detection systems (IDS) often rely on predefined rules or known attack patterns, which hinders their ability to identify new threats, especially at the packet level. To combat this, different techniques of natural language processing (NLP) have been increasingly adapted for the purposes of cybersecurity, treating network data, in particular, sequences of packet headers, as if they were language units. This literature review aims to explore the intersection of NLP and network malware detection, focusing particularly on the application of NLP techniques to packet- and header-level data. The scope is limited to research published within the last fifteen years applying NLP or NLP-adjacent techniques to packet capture or network telemetry.

2 Background

2.1 Malware Detection and Analysis

Traditional intrusion detection techniques for identifying malware in a system typically fall into one of two categories: signature-based and anomaly-based detection.

Signature-based detection relies primarily on signatures, which are recognizable patterns or characteristics that are associated with malicious activity in some way. These are extracted from a packet's payload, and from there they're compared to what is known as a signature library, a type of database for known or common signatures. An alert is sent forward if a match is found, and the packet is then typically redirected to a separate application that filters or disposes of the packet. [1] This proves effective in the case of malware with payloads that are both consistent and unencrypted; however, if a threat is new and lacking a defined signature, or the payload may be encrypted

or otherwise obfuscated, then it is possible that it will slip under the radar. In addition, the packet header is rarely considered when it comes to signature-based detection, and thus can ignore the "red flags" manifesting in metadata.

There is also anomaly-based detection, a machine learning (ML) approach to intrusion detection that flags deviations from expected, normal activity occurring within a system. What is considered "normal activity" for a system can be identified in multiple different ways, but typically always involves analyzing the behavior of a user profile in some way over time and creating a rule-based model with this data that can be later used as a baseline comparison. [2] As opposed to signature-based detection, anomaly-based systems are much better at identifying novel threats, and as many of them are based in ML, accuracy when identifying what is or is not a threat can be improved over time. However, there are still a host of issues. Most of these arise as it is difficult to find attack-free data to train with. If this data includes attacks, any behaviors associated with them that affect the system are often mistakenly trained to be seen as normal, meaning that similar behavior in other attacks might get overlooked. However, if the data is completely attack free, this can lead to the model having an increased sensitivity to any slight change, resulting in a higher rate of false positives for malicious behavior. [3]

While traditional intrusion detection methods rely on static rules or statistical deviations, such as with signature- and anomaly-based detection respectively, other models do exist that have been applied to packet intrusion detection, albeit to a lesser extent. These include, among others, the random forest algorithm, which involves building decision trees based on random subsets of data and combining their predictions to make a final one, [4]. In regards to intrusion detection, input data likely includes common header fields, such as flag status or packet length; from there, as the algorithm continues to encounter benign packets, a baseline will be established for what kind of behavior should be "expected." As such, once some kind of discrepancy appears, it becomes flagged immediately. [5]

2.2 Network Security

Network packets are small, individual units of data broken down from larger messages that are then transmitted across a network. Each packet primarily consists of two components: the header and the payload. The header consists primarily of control information, such as sequencing information, protocol types, and IP addresses for the source and destination; the payload, on the other hand, is the actual, intended message being transmitted. [6]

Further, every packet typically has multiple headers, with an additional one being added by each layer of the networking stack it passes through. As with Fink's diagram, [7], this may include, for example, the IP header from the network layer, which contains fields such as a packet's TTL and IP addresses, as well as the TCP header from the transport layer, which adds fields that specify ports and sequence numbers among other things. [8]

In essence, the purpose of the header is to provide context for the payload and, in a way, acts as the "envelope" for the payload to send it on its way. Network traffic, then, is the flow of packets over a network at any given moment. Because of the information they carry about network traffic, packet headers are crucial for identifying potentially malicious patterns and threats before they're able to escalate into full-blown attacks onto the system; for example, whether or not the source of a packet is from somewhere deemed "suspicious," or unauthorized protocol usage by way of analyzing the protocols attached to a given header.

2.3 Natural Language Processing

Natural language processing is a subfield of computer science that aims to train computers to process, generate, and manipulate human language. [9] This includes utilizing extensive linguistic knowledge to analyze the overall structure of a language from the ground up, starting at the word level and gradually moving up to the sentence at large and the overall context of a piece of text. [10] From there, this data is typically fed to different ML algorithms that can then create a conceptual model of how it believes a language operates, which can then be used for a variety of tasks, such as generating predictions or classifying content according to their linguistic features. Some common models include Google's BERT, which had been pretrained using a large corpora of English text for ease of use [11], and their earlier Word2Vec, which aims to create vector representations of individual words.

This unique approach to sequencing and processing data offers a unique advantage when it comes to intrusion detection. In the context of network traffic, packet headers can be "tokenized," or split into individual, manipulatable units, and then fed into ML models. [12] These models can then analyze and identify further patterns in how the different components interact both within individual packets and across entire sessions. In addition, packet headers are inherently sequential as a means of ensuring data is being delivered in the correct order. [13] NLP models tend to be well-equipped for this type of data as they are designed to understand and model the relationships be-

tween tokenized elements. Just as words in a sentence follow grammatical and syntactical rules that determine their overall order and relationship, the fields of a packet header follow similar structural conventions. [14] Once treated as language-like input, they can be contextually analyzed, which allows for the model to detect subtle deviations from typical traffic patterns that can indicate malicious or otherwise abnormal activity. [15] Unlike traditional signature-based detection, which rely on predefined rules and analyze data in isolation, NLP models, especially those based in ML, can infer new patterns from the given data itself, which makes it both adaptable and naturally well-equipped for detecting previously unseen threats. [16] In regards to anomaly-based detection, which often struggle adapting to changing traffic patterns and distinguishing between malicious and benign activity, NLP models allow for characterization of normal behavior that takes into consideration latent patterns, contextual dependencies, and abstract relationships between elements that can be overlooked otherwise. This in turn allows for a more nuanced understanding of traffic behavior that avoids tripping up the system in the same way. [16]

3 Design

This project proposes the design and implementation of an NLP-based malware detection system that models network packet headers as sequences of tokens. The performance of this approach will then be measured and compared against traditional machine learning models based on the same datasets. This in turn will help to evaluate the overall effectiveness and feasibility of applying NLP techniques for malware detection, and if its utilization improves accuracy.

3.1 Data Collection

Publicly available labeled datasets will be utilized for the purposes of this project, particularly CIC-IDS 2017 and UNSW-NB15 provided by the Canadian Institute for Cybersecurity and the University of New South Wales respectively. [17] Only packet header fields will be extracted for analysis.

3.1.1 Tokenization

From the fields extracted, those that will be selected for further tokenization include:

- Source and destination IP
- Source and destination ports
- Protocols
- Flags
- Packet TTL
- Overall Packet size

Each packet will be converted into strings of discrete tokens that represent each present field, and a vocabulary of these tokens will be defined. [18] From there, they'll be grouped into fixed-length sequences based on sessions; this will be done as a means of analyzing the overall flow and behavior of the packet.

3.2 Model Architecture

3.2.1 NLP Model

The BERT-based model will be fine-tuned based on sequences of packet headers, and used as a means of generating contextual embeddings for each packet within a given session. This essentially means that the tokenized data of the packet header will be transformed into a vector that, by nature, will allow for BERT to learn about the relationships between the different fields in a packet, both within a single one and across sequences. This in turn leads to each token relying on surrounding traffic content in addition to its own "identity," and leads to better differentiation between behaviors specific to benign and malicious traffic alike once the model is fine-tuned using different packet sequences. [19]

3.2.2 Traditional ML Model

For this project, random forest algorithms will be utilized as a means of comparing the more experimental results from the BERT model to a more proven method of intrusion detection. Typical header features are extracted utilizing the same data sets, turned into vectors, and fed to the model; from there, certain session-level features such as flag frequency and packet count can be determined, which leads to a development of a base line for system behavior that makes malicious or suspicious activity immediately get flagged. The same metrics, such as precision and recall, will be reported; this allows for a more fair comparison to be made for the practicality of NLP-based models as opposed to the ease of training and interpreting established ones.

4 Evaluation Plan

To test the overall effectiveness of NLP-based approaches for malware detection, the following experiments will be conducted:

- A random forest algorithm will be fed identical packet data to the core BERT model in order to compare how accurate and effective the BERT model is at flagging seemingly suspicious patterns within a packet's header fields

- Different tokenization strategies will be utilized on packets to evaluate both how well the model generalizes based on its given data and to further test its overall accuracy [18]
- Sequences of varying sizes and length will be tested as a means of analyzing how these factors play into model performance
- Ablation studies will be performed as a means of analyzing how the presence of specific fields contribute to the overall accuracy of the given model [20]

Further, the following tools and frameworks will be utilized for the project:

- CIC-IDS 2017 and/or UNSW-NB1 as the core data sets for use for the project
- Wireshark for capturing and exporting packet data
- Scapy for custom parsing, manipulation, and extraction of packet header fields
- pandas or NumPy for formatting data and converting extracted data into a more structured form, such as a token sequence
- The BERT language model for overall intrusion detection
- PyTorch for additional training of the BERT model
- scikit-learn for implementing a random forest

5 Contributions

5.1 Core Contributions

- The design and implementation of a more standard method for converting network packet data into a tokenized format for use for transformer models such as BERT
- Further evaluation on the ways in which tokenization choices and strategy impact model detection performance

5.2 Auxiliary Contributions

- Possible visualization depending on if particular embedding outputs offer interpretable signals on which parts of a packet's fields or sequences is relevant for detection
- Release of any code used for tokenizing and processing packet headers for NLP models

6 Risks and Challenges

Several challenges continue to hinder the widespread application of NLP models and techniques for the purposes of intrusion detection and packet header analysis.

1. Very few studies exist that directly compare NLP-based approaches with traditional ML-based ones specifically in the context of header analysis, which acts to limit understanding of exactly when NLP techniques are most effective for this purpose.
2. There is no consensus or standard when it comes to tokenizing packet headers for NLP models, which can make it difficult for further researchers to replicate findings or fairly compare any two models. This, in turn, results in being unable to properly or accurately measure improvements in benchmarking, as there is no real baseline to compare against.
3. Existing intrusion detection datasets are rarely designed with NLP applications in mind and, as such, may be difficult to use without artificial sequencing. NLP models are well-suited for unstructured text, but this is due to its inherent sequential structure that can still be tokenized and modeled even without proper formatting. Packet data, on the other hand, while technically structured (e.g. fixed formats, defined fields), isn't semantically structured in the same way, and NLP models aren't naturally equipped to interpret its raw numerical output. [17] This can be fixed through domain-specific tokenization, which involves converting this network data into a form that emulates language structure, [21] but this is usually manual and requires an in-depth understanding of the inner workings of packet metadata.
4. The overall applicability of NLP models in real-time detection scenarios remains limited due to latency in preprocessing efforts, time-consuming and computationally intensive processes, [22] and updating and fine-tuning models for specific domains or behaviors.

7 Timeline

Ordered by weeks:

1. Finalization on specifics for project; begin testing out Wireshark and manipulating packet data
2. Rough outline of approach by means of a technical report; begin setting up data and basic scripts for preprocessing

3. Create a rudimentary representation of data flow (i.e. raw packets to tokenization, BERT input, and finally classification)

4. Create a rudimentary visualization of the workings of BERT-based intrusion detection; compare with random forest

5. Polish rudimentary sketches and convert them to a digital diagram; begin building pipeline for tokenization

6. Finalize resources for literature review and methodology; train initial random forest model on chosen data set and finish its baseline feature extraction

7. Finalize tokenization strategy and begin to fine-tune the pretrained BERT model on tokenized packet sequences; compare these results with those from the random forest

8. Finish overview video; push any current scripts and models to GitLab

9. Add evaluation metrics and early results; begin to analyze performance of both models being tested

10. Create a poster with the preliminary results, key insights, and a fleshed-out system diagram

11. Update existing diagram based on further model testing, evaluation, and feedback

12. First demonstration; introduce problem, methodology, overview of BERT and random forest, early results

13. Finalize results and begin integrating feedback; begin adding any auxiliary work if time allows

14. Polish visuals, final metrics, and conclusions

15. Submit final report and portfolio to GitLab

8 Conclusion

The application of NLP to malware detection in network traffic is a promising yet relatively unexplored field. By leveraging the capacity of natural language processing to model sequences and recognize structural patterns, we can better understand network traffic and detect anomalous behaviors that can indicate potential malicious activity. In addition, the ability to tokenize packet headers, identify relationships between header fields (such as ports and protocols), and apply sequential modeling presents a significant advantage over traditional detection methods, including signature-based and anomaly-based IDS.

As the field continues to evolve, there are several areas where future research could further improve the capability of NLP techniques for network malware detection. For one, there is a clear need for both standardized tokenization methods and datasets to enable more consistent and comparable research. In addition, further exploration could be done in regards to the integration of NLP models with hybrid detection systems that combine traditional IDS in order to both cover for weaknesses and improve overall detection accuracy. Additionally, research into improving NLP model efficiency could help mitigate the computational concerns that hinder their use in real-time environments.

While NLP shows significant potential for improving how traffic analysis is approached, it is clear that further refinement is needed. The ongoing evolution of this field holds promise for developing more effective, adaptive, and scalable IDS systems capable of detecting a wider range of attack vectors with greater accuracy.

References

[1] Ahmad Azab et al. “Network traffic classification: Techniques, datasets, and challenges”. In: *Digital Communications and Networks* 10.3 (2024), pp. 676–692.

[2] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. “Network anomaly detection: methods, systems and tools”. In: *Ieee communications surveys & tutorials* 16.1 (2013), pp. 303–336.

[3] Robin Sommer and Vern Paxson. “Outside the closed world: On using machine learning for network intrusion detection”. In: *2010 IEEE symposium on security and privacy*. IEEE. 2010, pp. 305–316.

[4] Nabila Farnaaz and MA Jabbar. “Random forest modeling for network intrusion detection system”. In: *Procedia Computer Science* 89 (2016), pp. 213–217.

[5] Veeramani Sonai and Indira Bharathi. “Packet Classification Using Improved Random Forest Algorithm”. In: *International Conference on Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication*. Springer. 2023, pp. 157–168.

[6] Leslie F Sikos. “Packet analysis for network forensics: A comprehensive survey”. In: *Forensic Science International: Digital Investigation* 32 (2020), p. 200892.

[7] Glenn Fink. “Visual Correlation of Network Traffic and Host Processes for Computer Security”. In: (Oct. 2006).

[8] Wesley Eddy. *Transmission Control Protocol (TCP)*. RFC 9293. Aug. 2022. DOI: 10.17487/RFC9293. URL: https://www.rfc-editor.org/info/rfc9293.

[9] David Okore Ukwen and Murat Karabatak. "Review of NLP-based systems in digital forensics and cybersecurity". In: *2021 9th International symposium on digital forensics and security (ISDFS)*. IEEE. 2021, pp. 1–9.

[10] KR1442 Chowdhary and KR Chowdhary. "Natural language processing". In: *Fundamentals of artificial intelligence* (2020), pp. 603–649.

[11] Md Saiful Islam and Long Zhang. "A Review on BERT: Language Understanding for Different Types of NLP Task". In: *Preprints. org* (2024).

[12] Sabrina J Mielke et al. "Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP". In: *arXiv preprint arXiv:2112.10508* (2021).

[13] Glen Gibb et al. "Design principles for packet parsers". In: *Architectures for Networking and Communications Systems*. IEEE. 2013, pp. 13–24.

[14] Haoyu Song and John W Lockwood. "Efficient packet classification for network intrusion detection using FPGA". In: *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*. 2005, pp. 238–245.

[15] Yong Yang and Xing Peng. "BERT-based network for intrusion detection system". In: *EURASIP Journal on Information Security* 2025.1 (2025), p. 11.

[16] Zarrin Tasnim Sworna, Zahra Mousavi, and Muhammad Ali Babar. "NLP methods in host-based intrusion detection Systems: A systematic review and future directions". In: *Journal of Network and Computer Applications* 220 (2023), p. 103761.

[17] Markus Ring et al. "A survey of network-based intrusion detection data sets". In: *Computers & security* 86 (2019), pp. 147–167.

[18] Rafał Kozik, Michał Choraś, and Witold Hołubowicz. "Packets tokenization methods for web layer cyber security". In: *Logic Journal of the IGPL* 25.1 (2017), pp. 103–113.

[19] Chi Sun et al. "How to fine-tune bert for text classification?" In: *China national conference on Chinese computational linguistics*. Springer. 2019, pp. 194–206.

[20] Sina Sheikholeslami. *Ablation programming for machine learning*. 2019.

[21] Vin Sachidananda, Jason S Kessler, and Yi-An Lai. "Efficient domain adaptation of language models via adaptive tokenization". In: *arXiv preprint arXiv:2109.07460* (2021).

[22] Suresh Sharma and Tamilselvan Arjunan. "Natural language processing for detecting anomalies and intrusions in unstructured cybersecurity data". In: *International Journal of Information and Cybersecurity* 7.12 (2023), pp. 1–24.