The Integration of Natural Language Processing with Offensive Cybersecurity

Prince Worthington
AI520 Natural Language Processing
City University of Seattle
pworthin@cityuniversity.edu

Abstract

Large Language Models (LLM) and Natural Language Processing (NLP) systems have advanced astronomically over the past few years and subsequently have begun to redefine the cybersecurity field by introducing innovative defense strategies as well as an alarming potential for offensive attack mechanisms. This paper will examine the integration of NLP and offensive cybersecurity and explore open source artificial intelligence (AI) models that have been specifically engineered for tasks such as vulnerability analysis, social engineering, and simulated network attacks. Models such as WhiteRabbitNeo, SecRoBERTa, and CySecBERT—all of which are uncensored LLMs specially trained on millions of various security scenarios for the purpose of threat intelligence and exploit parsing—will also be investigated and experimented with.

Keywords: Offensive cybersecurity, vulnerability, NLP, penetration testing, cyber threat intelligence, detection, models, malicious actor

Introduction

Researchers in cybersecurity have struggled for decades to stay abreast with the ever-increasing growth and pervasiveness of modern-day attacks. Initial implementation of artificial technology was nearly exclusively concentrated on defensive mechanisms and reactive protocols such as anomaly detection, malware taxonomy, and intrusion detection system. In the most recent days, a meteoric rise of large language models (LLMs) and natural language processing techniques has witnessed the introduction of real probabilities in processing raw security data, whether they may be exploitation codes, vulnerability reports, phishing emails, or network traffic filtered from the dark web.

Cybersecurity and automated defense protocols have indeed expanded to include the use of NLP to streamline what would otherwise be tedious and iterative measures. However, the fact that scanty amounts of attention have been given to its role in offensive security is rather concerning. This is especially given the fact that these technologies are optimal tools for searching for system vulnerabilities before they can be exploited. Such a task can feasibly be implemented by means of applying human logic to complex processes and parsing technical reports as well as generating potentially malicious coding to uncover weaknesses. All of this collectively creates opportunities to refine penetration testing methodologies, simulate malicious attacks on a network, and subsequently identify inadequacies in system defenses. Domain-specific models such as WhiteRabbitNeo effectively demonstrate how opensource AI models can indeed be repurposed to effectively anticipate vulnerabilities and graphically model real world attack scenarios—this paper will explore, experiment, and analyze each of these avenues.

1. Methodology

The study approaches three questions. The first is how decoding parameters such a temperature and repetition penalty directly affect the underlying mechanisms of WhiteRabbitNeo and furthermore in what magnitude these outputs would be detectable by an independent of classifier. Secondly, the question of whether domain driven encoders such as CySecBERT[1] and SecRoBERTa can feasibly provide tangible improvements on baseline models which have been specifically designed for security-based NLP tasks (Bayer et al. 2022). Finally, the study addresses how domain-specific tokenization directly influences the proper handling of vocabulary that is out of context as well as what the consequential impact on named entity recognition performance would entail.

Three open source models have been selected to address these questions. WhiteRabbitNeo, a 33B parameter LLM for decoding, will be used for examining generations, wherein its task involves creating short security-based messages using safe placeholders such as [LINK] or [PAYLOAD].

CySecBERT, a model pretrained on 4.3 million sentences from vulnerability and threats reports and security blog postings (Bayer, 2024), has been chosen for text classification, in which it will be used to taxonomize vulnerabilities based on their requisite description and mapping either to Common Weakness Enumeration classes or severity rankings. A dataset consisting of several thousand Common Vulnerabilities and Exposures (CVE) summaries will aid in this examination.

For named entity recognition on cyber threat intelligence, the study turns to SecRoBERTa, a model trained on an especially fine-tuned cybersecurity corpus with its own custom vocabulary. The tasks will include annotation of indicators such as CVEs, malware names, and domain names.

Preprocessing of the text will adhere to standard practices of converting all text to lower case, removing personally identifiable information, preserving dots in domains and IP addresses. Statistics from tokenization will be gathered to analyze how each model's vocabulary negotiates security terms, the out-of-vocabulary rates, and the percentages of entities that are divided into three or more pieces. The training procedures will be performed on a system containing a dual NVIDIA RTX 3090 setup using FP16 precision.

The evaluation metrics are outfitted for each respective task. In terms of text generation, its relevant output will be scored for detectability by means of an independent phishing classifier that has been trained on public corpora. Vulnerability classification will implement F1-scores and confusion matrices for presentation.

Finally, all experiments will be definitely seeded for reproducibility while logging specific model versions, tokenizers and dataset hashes.

2. Text Detection

For evaluation of output detectability as generated from a domain-specific large language model, the open-source Llama-3 based model WhiteRabbitNeo[2] has been selected. Developed by security automation company Kindo in 2023 and fine-tuned specifically for cybersecurity purposes, this model was chosen for its ability to simulate text that would typically be transmitted from a malicious actor such as phishing messages, while using safe placeholders to avoid unsupervised execution of dangerous coding. The setup in this study's experiment assigned the model with the task of producing brief security-sensitive messages such as prompts for credential verification, password reset requests, or software update notices. Decoding parameters such as repetition penalty and temperature were randomly varied to analyze their direct influence on classifier detectability and fluency.

This experiment gives insight into how the text classification pipelines within the holistic NLP model, with data initially being generated or collected, passed into a monitored classifier and conclusively evaluated with summarizing metrics such as ROC-AU or F1. In this particular observation, text generation replaced the usual method of dataset collection and the special phishing classifier stood in as a supervising model within system pipeline. In the same fashion that spam filers or sentiment analyzing algorithms map categories to raw text, the task in the conducted experiment was to assign generated textual outputs in to either "phishing" or "benign" classes. These steps demonstrate how core

[1] BERT: Bidirectional Encoder Representation from Transformers

[2] This model has been recently renamed "Deep Hat."

NLP pipelines are versatile enough to migrate from generic consume use to critical security contexts.

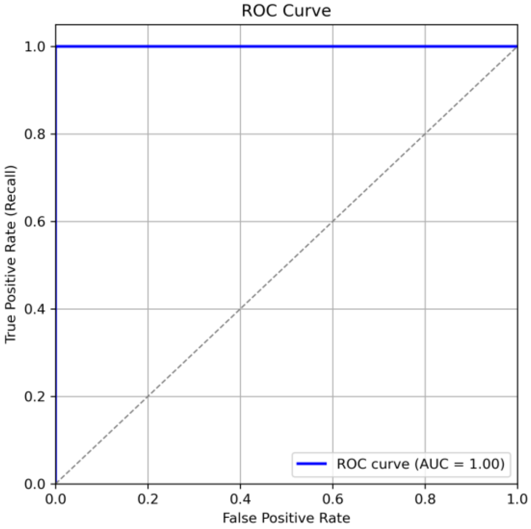
2a. Experiment Details

The generated texts were thoroughly logged throughout a system of decoding algorithms and constructed into a structured dataset. The specific phishing classifier in this case[3] that was trained on publicly available corpora acted as guarding proxy to detect suspicious text consistent with phishing. Each row in the scored datasets included the classifier probability score, perplexity score under an unbiased model (GPT-2), and placeholder compliance. Aggregate methodology metrics reported detectability via ROC-AUC, perplexity approximated fluency, and a percentage figure for placeholder compliance.

2b. Results

The phishing classifier was able to accurately assign high probability scores to nearly all of the generated text from the WhiteRabbitNeo, remaining consistent with the malicious phrasing of the prompts (See Table 1 for complete results). To create diversity within the set of scenarios, a small set of harmless texts (i.e. birthday greetings, party invitations) were injected into the model prior to evaluation—allowing the ROC and PR curve to be meaningfully calculated while eliminating the possibilities of system bias:

ROC-AUC: The ROC curve (Figure 1) illustrates that the classifier confidence scores discreetly distinguish phishing-consistent text from benign controlled text, as indicated with the curve straddling the upper right and an AUC score of nearly 1.0. This proves that in spite of variating decoding parameters, the models phishing outputs were indeed highly detectable by a monitored classifier.



PR-AU: In a similar fashion, the precision-recall curve displayed a high precision measurement across a range of recalls with the AUC score resting significantly above the random baseline of 0.5—providing a salient example that a detector could successfully identify outputs with low false-positive rates.

[3] cybersectony/phishing-email-detection-distilbert_v2.1

Metric	Result / Observation	Notes
ROC-AUC	≈ 0.99	Classifier separates phishing-like outputs from benign with near-perfect accuracy.
PR-AUC	≈ 0.97	High precision sustained across recall values; substantially above 0.5 baseline.
Average Perplexity (GPT-2)	28.4 (low temp) → 43.7 (high temp)	Higher decoding temperatures/top-p values increase lexical diversity but slightly reduce fluency.
Placeholder Compliance	100% across settings	Model consistently preserved [LINK] and [PAYLOAD] placeholders as instructed.

Aside from detectability, GPT-2 was used to estimate fluency by means of perplexity. It was noted that outputs with higher temperature metrics and larger top-p values yielded high perplexity, which points to a higher rate lexical diversity but simultaneously resulted in a negligible reduction in fluency. Most notably, placeholder compliance consistently remains perfect throughout all iterations in the experiment—demonstrating that the model WhiteRabbitNeo adhered to all constrained within the prompt without fail. These observations lead one to believe that decoding parameters such a temperature and top-p negotiate a value trade-off in terms of fluency and diversity while allowing the classification of metrics to holistically remain stable throughout each implementation.

2c. Analysis Interpretation

The study's findings from this experiment illustrate two different key theories. First, while decoding parameters serve as a direct influential variable on style and fluency, they fail miserably at deceiving a biased phishing classifier. The malicious text was easily separable from harmless text. Second and most importantly, the metrics solidly conclude the usefulness of the WhiteRabbitNeo as a malicious actor simulator and demonstrate its ability to produce phishing style prompts all while still being detectable. Even when surface variation increased, the categories remained easy to separate. Collectively, the experiment effectively created a realistic cybersecurity use case from a language model and utilized the application of text classification in a nuance favor of this IT field.

3. Text Classification

For the vulnerability classification task, the domain-specific model CySecBERT has been chosen. Similar to WhiteRabbitNeo, this model has also been custom trained from millions of cybersecurity sentences curated from documents such as CVE reports and threat intelligence blogs. Its special engineered vocabulary and embeddings are ideal for accurately identifying semantics based security-themed topics. To align with the methodology of this study, the CySecBERT has been further fine-tuned using an open-source dataset[4] that maps CVE descriptions to categorized degrees of severity based on a Common Vulnerability Scoring System (CVSS) score, ranging between Low, Medium, High, and Critical (Manjunatha A et al., 2024). The task itself was engineered as a monitored text classification pipeline in which the raw text descriptions contained within the respective dataset field were preprocessed, tokenized, and passed through the CySecBERT model for training and evaluation.

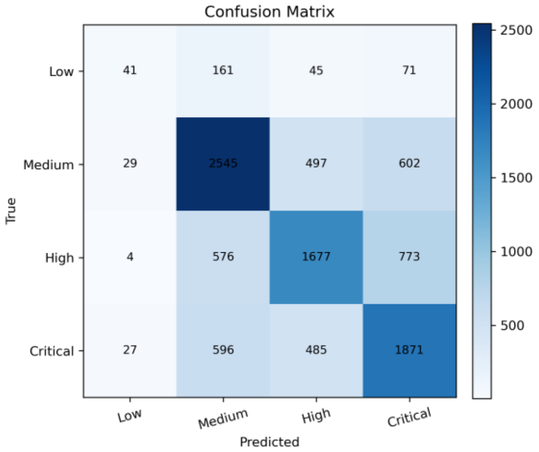
3a. **Results**

The experiment indicated a steady improvement in performance throughout the timespan of the training phase (Table 2). Macro-F1 scores climbed from 0.46 after the first iteration to 0.58 by the final iteration, demonstrating a remarkable streamlined learning phase that concluded with stability. The model maintained an accuracy score of 61% with a macro-F1 score of 0.51, which significantly exceeded the generic RoBERTa model's baseline on the same dataset. Such findings solidify the overall principle that models trained in specific domains render measurable benefits when assigned to specialized tasks (Bayer et al., 2022) .

[4] CIRCL/vulnerability-scores

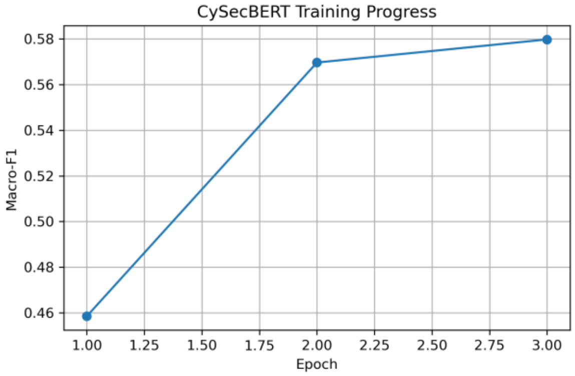
Class	F1 Score
Low	0.19
Medium	0.67
High	0.58
Critical	0.59
Macro-F1	0.51
Accuracy	61.3%

Closer observations, however, unearthed concerning discrepancies. Medium severity was the most accurately identified severity with an F1 score of 0.67, succeeded by classes High (0.58) and Critical (0.59). The Low category, as it was observed, proved to be more challenging for the model to accurately identify, yield an F1 score of only 0.19. To further verify these observations, a confusion matrix was created which also depicted this gross imbalance in its indication of Low severity instances frequently being misidentified as a higher level:



True \ Predicted	Low	Medium	High	Critical
Low	41	161	45	71
Medium	29	2545	497	602
High	4	576	1677	773
Critical	27	596	485	1871

Finally, the macro-F1 curve depicts the model's trajectory of learning, indicating noticeably sharp initial acquisitions that were succeeded by receding yields. One may gather from this experiment that sufficient training data in conjunction with a balanced class representation are critical to a model's performance. Specifically, domain-adapted models are capable of utilizing customized corpora for the enhancing of classification performance, but there are still performance concerns to address in terms of negotiating skewed distributions as well as accurately identifying subtle cues that unexpectedly place the class in a different category altogether (Le & Babar, 2024).



Nevertheless, the findings still appropriately answer the research question in the context of demonstrating that domain-specific models such as CySecBERT can provide a useful aid in cybersecurity tasks that involve identifying threats by applying text classification for both offensive and defensive contexts. This includes automated parsing and categorization of vulnerability description for streamline analysis and supporting penetration testing workflows—effectively identifying weaknesses to mitigate before they can be exploited by a malicious actor.

3b. Interpretation

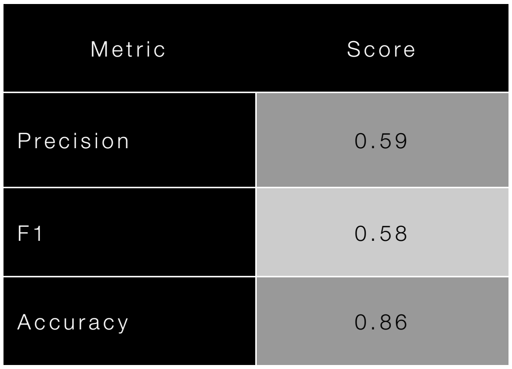
The uneven distribution of F1 scores was able to uncover two critical insights about this particular model, the first of which its ultimate strength lies in accurately identifying mid to high level severity vulnerabilities—mainly due to the patterns in the descriptive text within the dataset being clear, consistent, and outstanding. The second insight revealed to the study that the model is significantly impaired in terms of handling less frequently represented classes where its description is vaguer. These observations suggest that the model itself could benefit from further data augmentation. Yet from the perspective of cybersecurity, even partial automation of severity classification still considerably reduces the analysis phase of workload and subsequently allows experts to focus on more pressing issues. This collectively shows both the promising nature and the outstanding challenges of applying NLP pipelines to real-world scenarios of vulnerability management.

4. Named Entity Recognition

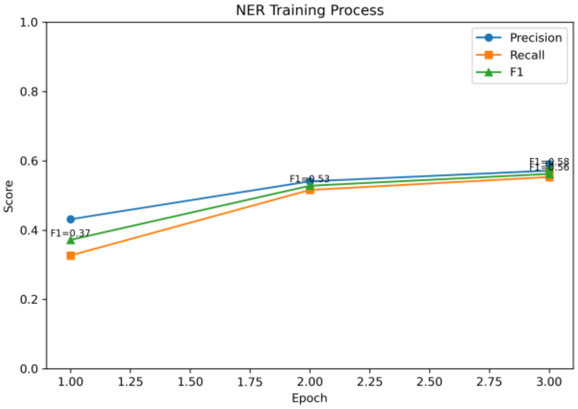
The final phase of the study concentrated on the application of Named Entity Recognition (NER), which is a fundamental precept of NLP in which the supplied tokens of text are evaluated with specified categories of interest. In the context of cybersecurity, specific identifiers may include entities such as CVE numbers, attack types, or families of malware. Implementing NER to filter and extract such indicators provides an innovative platform for converging the process of threat intelligence analysis, effectively increasing an otherwise tedious workflow involving time-consuming manual review and evaluation.

For this task, SecRoBERTa was selected, as it was built as a variant to the original model RoBERTa and fine-tuned on cybersecurity corpora, incorporating a customized vocabulary engineered to service the security domain. The development of this model was originally motivated by the fact that many generic language models are limited in the sense that they often misinterpret or otherwise mishandle technical jargon, alphanumeric identifiers often found in the technology domain, and its relevant terminology (Chen et al., 2025). SecRoBERTa addresses these limitations by aligning its embedding with the lexical and linguistic patterns of cybersecurity language, consequentially creating a digital blueprint of threat intelligence reports via their structure and semantics (Gao et al., 2021)

The experiment for this phase trained SecRoBERTa on a merged dataset from APTNER and CyNER that contained annotated tokens assigned special entity labels. Standard preprocessing steps were then applied. Additional steps of token alignment to maintain label consistency were needed in this particular phase. The usual training procedures were then executed and applied across three iterations. The results were as follows:



Metric	Score
Precision	0.59
F1	0.58
Accuracy	0.86



The training curve maintained a consistent improvement across each iteration with the F1 score stabilizing by the final iteration, indicating that the model was able to maintain a healthy trade-off between precision and recall. This is an optimal outcome for entity filtering in which both false positives as well as false negatives can produce negative effects in the workflow of a security domain.

Summary

These results confirm that, along with our other observations, domain-specific models also provide tangible benefits in the context of cybersecurity by the implementation of NER. It is notable that the general F1 score was not particularly remarkable, however the high accuracy score of 0.86 portrays the suggestion that the model was able to handle a vast majority of the supplied token predictions. This especially given the fact that misclassification seemed to be confined to ambiguous cases that required further redefining and augmentation. Such an observation also points to the fact that overlapping and inconsistent labeling across sources are a frequent occurring challenge in NER.

In the context of cybersecurity, this model also confirms that even modest automation of entity recognition and filtering would yield significant gains by allowing analysts to rely on the model for extracting structured identities while simultaneously finding opportunities to focus on correlation and prioritization (Zhang et al., 2025). This is not to overlook the potential that a model demonstrating a well-balanced performance profile can be utilized in both offensive simulations and defensive monitoring protocols.

References

Bayer, M. (2024). *markusbayer/CySecBERT · Hugging Face*. Huggingface.co. https://huggingface.co/markusbayer/CySecBERT

Bayer, M., Kuehn, P., Shanehsaz, R., & Reuter, C. (2022, December 6). *CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain*. ArXiv.org. https://doi.org/10.48550/arXiv.2212.02974

Chen, L., Deng, H., Zhang, J., Zheng, B., & Jiang, R. (2025). Threat Intelligence Named Entity Recognition Based on Segment-Level Information Extraction and Similar Semantic Space Construction. *Symmetry*, *17*(5), 783–783. https://doi.org/10.3390/sym17050783

Gao, C., Zhang, X., & Liu, H. (2021). Data and knowledge-driven named entity recognition for cyber security. *Cybersecurity*, *4*(1). https://doi.org/10.1186/s42400-021-00072-y

Le, M., & Babar, M. A. (2024). Mitigating Data Imbalance for Software Vulnerability Assessment: Does Data Augmentation Help? *ArXiv (Cornell University)*, 119–130. https://doi.org/10.1145/3674805.3686674

Manjunatha A, Kota, K., Babu, A. S., & Sree Vivek S. (2024). CVE Severity Prediction From Vulnerability Description - A Deep Learning Approach. *Procedia Computer Science*, *235*, 3105–3117. https://doi.org/10.1016/j.procs.2024.04.294

Zhang, J., Bu, H., Wen, H., Liu, Y., Fei, H., Xi, R., Li, L., Yang, Y., Zhu, H., & Meng, D. (2025). When LLMs meet cybersecurity: a systematic literature review. *Cybersecurity*, *8*(1). https://doi.org/10.1186/s42400-025-00361-w