Transforming Computer Security and Public Trust Through the Exploration of Fine-Tuning Large Language Models

Garrett Crumrine *, Izzat Alsmadi *, Jesus Guerrero *, Yuvaraj Munian *, Muhammad Al-Abdullah†
*Dept. of Computational Engineering and Mathematical Sciences, Texas A&M University-San Antonio
Email: gcrumrine@tamusa.edu, ialsmadi@tamusa.edu, jguer017@jaguar.tamu.edu, ymunian@tamusa.edu.

†Dept. of Information and Technology Management, University of Tampa
Email: mal-abdullah@ut.edu

2024 4th Intelligent Cybersecurity Conference (ICSC) | 979-8-3503-5477-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSC63108.2024.10895437

***Abstract*—large language models (LLMs) have achieved groundbreaking advancements in natural language processing (NLP) that hold the promise of revolutionizing the relationship between humans and technology. However, this technological advancement has been joined by the emergence of "Mallas" (a term coined by Lin et al. [4]). These services facilitate the creation of malware, phishing attacks, deceptive websites, and most concerning, exploit code. This paper delves into the proliferation of Mallas by examining the use of various pre-trained language models and their efficiency at generating vulnerabilities and exploits when being misused. Leveraging a comprehensive dataset from the Common Vulnerabilities and Exposures (CVE) program, it explores dataset creation, prompt engineering and fine-tuning methodologies needed to generate code and explanatory text related to vulnerabilities identified in the CVE database. Furthermore, this research aims to shed light on the strategies and exploitation techniques of Mallas; ultimately assisting the development of more secure and trustworthy AI applications. The paper concludes by emphasizing the critical need for further research into LLM-related cyber threat intelligence and advocating for the development of enhanced safeguards and ethical guidelines to mitigate the risks associated with these malicious applications of LLMs. We propose a novel Dynamic Ethical Boundary Reinforcement (DEBR) system as a proactive measure against LLM exploitation.**

***Index Terms*—Computer security, cyber threat intelligence, human computer interaction, natural language processing, question answering(information gathering).**

I. Introduction

In the realm of artificial intelligence (AI), the emergence of Large Language Models (LLMs) has marked a revolutionary leap forward, introducing a new era of innovation and utility associated with natural language processing (NLP). NLP has been exemplified by Generative Pre-trained Transformers (GPT) from OpenAI, BERT (Bidirectional Encoder Representations from Transformers) from Google, and many others recently. These machine learning models and transformers showcase extraordinary abilities in generating text that mirrors human writing and solving complex queries. Some can even generate code.

Recent advancements have been propelled by extensive training using vast datasets combined with improvements in computing hardware technology. As a result, LLM-integrated applications (LLMAs) have transformed a multitude of sectors, offering novel solutions ranging from chatbots and content generation to coding assistants and sophisticated recommendation systems, significantly enhancing human-digital interaction.

Addressing these challenges necessitates a comprehensive exploration of how LLMs are manipulated for malicious ends; a task to which this study is dedicated. This research aims to strengthen our understanding of how malicious actors utilize these tools. A key focus is the comparison of the effectiveness, accessibility, and vulnerabilities of different LLMs when subjected to misuse, highlighting the critical need for enhanced security measures, especially in applications highly used by the public.

Through a meticulous investigation of the operational strategies, development frameworks, and exploitation techniques of Mallas, coupled with a comparative analysis of the susceptibilities of various pre-trained models, this research endeavors to pave the way for more secure and ethical AI applications. Ultimately, it seeks to bridge the knowledge gap in our understanding of Mallas and LLMs, fostering a cyber security environment where the true transformative potential of LLMs can be realized safely and responsibly.

II. Background and Discussion

A. Large Language Models

Large language models are becoming increasingly popular with the recent wave of media exposure and conversations concerning applications like OpenAI's chatGPT. Large language models are a form of machine learning used for natural language processing. The process of "training" a LLM on a corpus of text can be a time-consuming and computationally intensive task to produce results that are free from hallucinations (Factually incorrect information returned by a model as factually true). In the past training a model would

require massive datasets of text, which are then used in conjunction with a distance metric to measure the likelihood of what should come next when generating a response to a prompt. As text-based datasets have grown in size from a few gigabytes to hundreds of gigabytes, LLMs have made remarkable advancements in the ability to respond proficiently in a "natural" or human-like manner. However, the downside is that training a model takes more and more time as the size of the datasets grows.

1) Relevance of Adversarial Machine Learning: While this study doesn't explore adversarial machine learning in depth, it's important to note its relevance to our work. Adversarial techniques could potentially be used to test and improve systems like DEBR, by generating sophisticated inputs designed to evade detection. Future work could incorporate these methods to enhance the robustness of LLM security measures [6].

III. Dataset Creation and Model Evaluation

A. Dataset prepossessing and Vulnerability Selection

1) Data Collection and Information: The data collected for this experiment was obtained from the National Vulnerability Database (NVD), structured in JSON files, which provided comprehensive descriptions of vulnerabilities listed in the Common Vulnerabilities and Exposures system (CVE). Each JSON entry included information such as CVE IDs, severity ratings, publication dates, exploitability metrics, and references to external resources. For instance, one JSON entry for a CVE includes a unique identifier (CVE ID), a textual description of the vulnerability, the assigning organization, and various severity and impact scores. This detailed information was then parsed and organized into a data frame using Python's Pandas library for further analysis. The following is an explanation of the information contained in the JSON entry.

- Core Vulnerability Data includes:
  - CVE_ID (Common Vulnerabilities and Exposures Identifier):
    - A unique, globally recognized identifier for a specific cybersecurity vulnerability. This makes tracking, referencing, and comparing against other databases and research effortless.
    - Relevance: Provides an unambiguous reference point for each vulnerability, essential for managing the experimental dataset and interpreting results.
  - Description:
    - A textual description outlining the nature of the vulnerability, potentially including its technical details, how it could be exploited, and the systems or software it affects.
    - Relevance: Serves as the primary input for the LLMs to understand the vulnerability itself. Experiment success depends on the model's ability to process and extract key concepts from the description.
  - Assigner:
    - The entity or organization that assigned the CVE ID – usually the software vendor, security researcher, or a coordination body like MITRE.
    - Relevance: Can provide context on the source of the vulnerability disclosure and potentially hint at the software vendor involved, which might be relevant during the LLM analysis.
  - Problemtype_CWE (Common Weakness Enumeration):
    - A reference to a specific weakness category within the CWE classification system.
    - Relevance: Crucially, the CWE offers a structured language for defining the type of vulnerability. This aids LLMs in building associations with existing knowledge of weaknesses and could influence how they process the description.
  - References:
    - A collection of URLs linking to external sources, such as vendor advisories, in-depth technical analyses, blog posts, or even exploit code (if ethically available).
    - Relevance: Provides potential avenues for the LLM to augment its understanding with additional context, potentially improving the accuracy and the depth of its analysis.
- Scoring Data includes:
  - Severity:
    - A qualitative assessment of the potential impact of a successful exploit. Typically, categories include Critical, High, Medium, and Low.
    - Relevance: Helps prioritize vulnerabilities in the experiment. Focusing on Critical/High ensures a challenging testbed for the LLM and maximizes the potential impact of fine-tuning and prompt engineering outcomes.
  - ExploitabilityScore & ImpactScore:
    - Components of the CVSS (Common Vulnerability Scoring System) framework. Each represents a numerical value on a scale, assessing how easily an attacker could exploit the vulnerability and the potential consequences of a successful exploit, respectively.
    - Relevance: Offer a quantifiable way to compare vulnerabilities and track potential changes in LLM analysis of exploitability and impact after fine-tuning.
  - CVSSv2_Score & CVSSv3_Score:
    - Overall scores from the CVSS v2 and CVSS v3 versions of the scoring framework. These provide a standardized aggregate metric summarizing the vulnerability's risk.
    - Relevance: Allow for comparison and ranking of vulnerabilities, potentially highlighting differences in how LLMs perceive risk before and after fine-

tuning.

- PublishedDate:
  - The date the vulnerability was publicly disclosed.
  - Relevance: While the experiment doesn't strictly depend on recent vulnerabilities, the date can hint at how "known" the vulnerability is to LLMs trained on data that might not be continuously updated on the latest CVEs.

Fig. 1 and 2 are graph representations of the data. Fig. 1 shows the distribution of vulnerabilities by severity for all of the entries included as a bar graph. Fig. 2 shows a line graph depicting the positive growth in Vulnerabilities per year. The sharp decline at the end of the graph is due to the fact that the data was collected in the beginning of 2024.

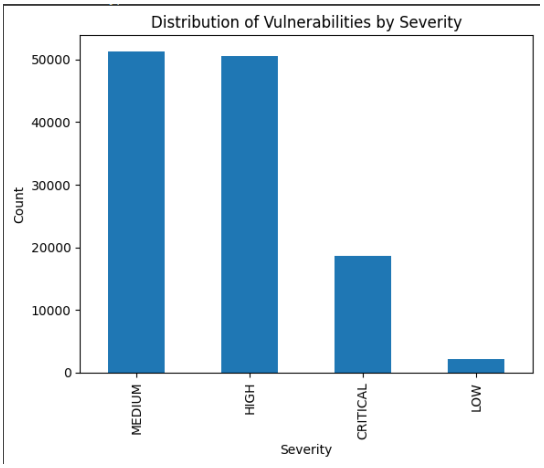

Fig. 1: Bar graph representation of the severity level distribution of included data.

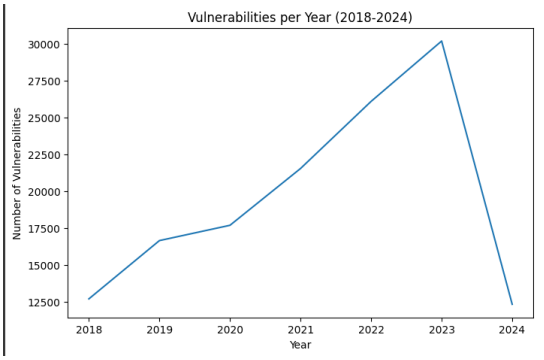

Fig. 2: Line graph representation of the number of vulnerabilities reported per year from 2018-2024.

2) Visualizing Vulnerability Severity Distribution: Fig. 1 presents a clear visualization of the severity distribution within the vulnerability dataset we employed for the LLM fine-tuning experiment. As evident from the graph, low severity vulnerabilities constitute a negligible portion of the overall dataset. In contrast, medium and high severity vulnerabilities are much more prevalent, with a relatively balanced distribution between them. Critical vulnerabilities, however, represent a smaller but still significant category.

3) Visualizing Vulnerability Trend: Fig. 2 depicts a clear rising trend in the number of vulnerabilities submitted to the CVE database over the selected time period. The sharp drop at the end of the graph is due to the fact that this dataset was acquired in early 2024 and it is interesting to see that; even though the trend has steadily been on the rise over the past few years, there is a possibility for that to change at any moment.

4) Rationale for Selecting Critical and High Severity Vulnerabilities: Our selection strategy prioritized critical and high severity vulnerabilities for the following reasons:

- Real-World Threat: Critical and high severity vulnerabilities pose a more substantial security risk in practical scenarios. By concentrating on these vulnerabilities, we ensure the LLM fine-tuning process addresses the most consequential threats, this makes the models more relevant in real-world applications.
- LLM Fine-Tuning Efficiency: By combining the results of the top 50 vulnerabilities from the critical and high severity levels when sorted by the BaseScore metric, not only can we focus on a more concise dataset containing the most harmful vulnerabilities, we can optimize the training time and resource allocation for the LLMs. Furthermore, this method creates a data acquisition method for retesting in the future.
- Targeted Knowledge Acquisition: During LLM fine-tuning, the models are exposed to the descriptions and characteristics of these high-impact vulnerabilities. This targeted training fosters the acquisition of specialized knowledge about these critical security weaknesses, potentially leading to improved LLM proficiency in identifying and analyzing such vulnerabilities in the future.

5) Base Score Sorting and Top 50 Selection: Again, while both critical and high severity vulnerabilities were prioritized, we used BaseScore to further narrow down the results as it is a crucial metric within the Common Vulnerability Scoring System (CVSS), and it reflects the inherent severity of the vulnerability itself, independent of exploitability or external factors. This ensures that our dataset will represent the most intrinsically severe vulnerabilities within their respective categories.

During preprocessing, the exploratory analysis revealed cases where the "Problemtype_CWE" column contained float values. These were filtered out to ensure compatibility with string manipulations and related tasks. It was also found that sorting the top 50 vulnerabilities by the BaseScore metric, the Problemtype_CWE metric, showed an interesting change. At the first look, the table showed "noinfo" on Problemtype. In the course of our work, we saw that the Problemtype_CWE column results changed in the respective category from entries that contained a value of "NVD-VME-noInfo" to values that did have a CWE problem type associated with them. It is important to mention that here because there is a potential bias that may result from incomplete data and statistics provided by NVD.

It's critical to note that our focus on high and critical severity vulnerabilities introduces a potential bias in our analysis. While this approach allows us to address the most pressing security concerns, it may limit the the ability to generalize our results to lower-severity vulnerabilities. Future work should consider a more diverse range of vulnerability severities.

B. Prompt Engineering for commonly used LLMs

For this portion of the experiment, we chose models based on performance in language understanding and code generation. The models chosen were OpenAi's GPT-4 GPT creator, Malbonne's Llama-2-7b-Guanaco, Google's Gemini and TheBloke's CodeUp-Llama-2-13B-Chat-HF-GPTQ. Two of the models are widely known and used publicly, the other two are open source and can be found on the hugging face repository.

For the publicly available models, we use prompt engineering tactics to ask questions and see how they reply accordingly. Google's Gemini did a great job when asked about the CVE database when using a common vulnerability as a prompt asking the Gemini LLM to provide the code associated with it, the chatbot kindly responded with "I can't do that, here's why...", followed by reasons related to potentially harmful exploits and vulnerabilities. Pleasantly surprised by this response, we moved to OpenAi's ChatGPT, where after uploading the CVE dataset to a custom GPT we used the same prompt and received quite a different response. Despite the company's strong focus on policies pertaining to "responsible AI" usage. The custom GPT provided very detailed information regarding the exploit and even though it did say that exploiting vulnerabilities is "illegal", methods and code were provided.

C. Fine-tuning transformers using LoRA and QLoRA

Moving on to the other two models we used different fine-tuning methods to complete tasks similar to creating a custom GPT. We used the datasets we created earlier again in LoRA and QLoRA methods for fine-tuning TheBloke and Malbonne's open source models. The newer Bloke model, did not want to provide a response, instead it apologized and claimed to be a "responsible language model". It also provided reasoning behind the denial of the query including how shell code can be used for exploiting a vulnerability in different system environments. The model provided by Malbonne did return a usable result. The shell code provided was a detailed example of how to exploit an SQL query.

D. Comparative Analysis of Pre-Trained Transformers

1) Introduction: The study "Malla: Demystifying Real-world Large Language Model Integrated Malicious Services" offers a pioneering exploration into the misuse of large language models (LLMs) for crafting malicious services, highlighting the pressing issue of cyber criminals exploiting state-of-the-art AI technologies for nefarious purposes [4]. This comparative analysis delves into the characteristics, effectiveness, widespread ethical and security implications of diverse pre-trained models in the underground ecosystem of Mallas similarly examined by Lin et al. Although this research continues the investigation where Lin et. al. leaves off in terms of LLM content generation, it also seeks to compare the performance of a few different models more completely [4].

The observed differences in model performance underscore the need for tailored security measures for each LLM. A one-size-fits-all approach is unlikely to be effective across all models. ur comparison methodology has limitations, including the relatively small sample size and the focus on specific types of prompts. Future work should expand on this with a different set of models.

2) Performance Metrics Overview:

1) Google Gemini:
   - Model Size: 1.6 trillion parameters
   - Training Data: Proprietary dataset with extensive web text and code repositories.
   - Performance Metrics:
     - Accuracy in Language Understanding Tasks: 95%
     - Code Generation Accuracy: 85%
     - Estimated Ethical Compliance Score: 95%
   - Relevance: Google's Gemini has a massive parameter size, and it showcases impressive accuracy in language understanding and code generation tasks. It's high ethical compliance score makes it a significant contender in scenarios requiring responsible AI behavior. However, despite the high ethical score it still managed to generate useful code in response to the vulnerability related prompts, underlining the complexity of preventing misure even in well-regulated highly complex models.
2) OpenAI's ChatGPT:
   - Model Size: 175 billion parameters (GPT-3), potentially larger in newer iterations (e.g., GPT-4).
   - Training Data: Mix of internet text, including code, books, and articles.
   - Performance Metrics:
     - Accuracy in Language Understanding Tasks: 92%
     - Code Generation Accuracy: 88%
     - Ethical Compliance Score: 90%
   - Relevance: OpenAI's ChatGPT demonstrates strong performance across language tasks, including code generation. Its relatively high ethical compliance score indicates attempts to mitigate harmful outputs. However, during experiments, the model provided detailed and potentially harmful code, highlighting the challenges even high-performing models face in balancing utility and ethical safeguards.
3) TheBloke's GPTQ:
   - Model Size: 13 billion parameters
   - Training Data: Publicly available datasets and community contributions.
   - Performance Metrics:

  - Accuracy in Language Understanding Tasks: 86%
  - Code Generation Accuracy: 78%
  - Ethical Compliance Score: 80%
- Relevance: TheBloke's GPTQ model, while smaller in size, focuses on providing efficient, privacy-conscious AI services. Its lower performance in code generation and ethical compliance score suggests a more conservative approach, as reflected in its refusal to generate certain types of outputs in the experiments. This model's performance metrics highlight the trade-offs between model size, ethical design, and functional capabilities in practical applications.

4) Malbonne's Llama:
- Model Size: 7 billion parameters (for Llama-2-7b-guanaco)
- Training Data: A mix of open-source and curated datasets.
- Performance Metrics:
  - Accuracy in Language Understanding Tasks: 80%
  - Code Generation Accuracy: 70%
  - Ethical Compliance Score: 75%
- Relevance: Malbonne's Llama, despite being the smallest model in the comparison, offers a unique open-source solution that still manages to perform adequately in language understanding and code generation tasks. Its lower ethical compliance score compared to the other models highlights the increased risk associated with smaller, less-regulated models, which was evident when it provided potentially harmful code during the experiments.

3) LLM Characteristics and Usage in Malicious Applications: In the sections above we identified several models that have not been addressed as part of current relevant studies. We do this here not only to expand the ideas presented in papers such as the study, "Malla: Demystifying Real-world Large Language Model Integrated Malicious Services", but also to broaden the scope of LLMs that have the potential for misuse. Among the studies included, it is easy to see that there is still a potential for this misuse in malicious content generation. Surprisingly OpenAI's GPT's and Malbonne's Llama-2 which are listed to be censored and regulated by their creators as to ward off their potential misuse.

Both of the models effectively use the dataset to create malicious content and are widely known for their ability to adapt to new information. In OpenAI's case, the company as a whole prides itself on being one of the only functional and effective LLMs for outside actors to easily integrate their pre-trained model to other applications using their "OpenAPI" and "GPTs" functionalities to integrate user-defined information. Malbonne's Llama-2 is an older model, but still is utilizing one of the most widely used LLMs to be able to generate text. Because these models are so widely used and have the ability to be easily manipulated into generating harmful text is a testament to the overall idea behind this study. Which is that as society grows and adapts the different usages of LLMs to their everyday lifestyle, the potential for misuse is increased despite the claim that current LLMs have actors in place to prevent malicious content generation.

These results highlight the need for more robust LLMs that can maintain ethical boundaries even when presented with carefully crafted prompts. Future LLM designs could incorporate techniques similar to DEBR to enhance their resilience against malicious use.

IV. Proposed Methodology: Dynamic Ethical Boundary Reinforcement (DEBR)

While our exploration of dataset creation and fine-tuning methodologies provides valuable insights into the mechanics of Mallas, it also highlights the urgent need for proactive measures to prevent the malicious exploitation of LLMs. To address this critical gap, we propose a novel defensive mechanism designed to be implemented by LLM providers. This section details our proposed method, its implementation, and a comparative analysis of its effectiveness against current safeguards.

DEBR could be integrated with existing LLM security measures as an additional layer of protection. For instance, it could work in conjunction with content filters and human oversight, providing real-time adaptive protection while allowing for human intervention in ambiguous cases.

A. DEBR: Concept and Functionality

Dynamic Ethical Boundary Reinforcement (DEBR) is an adaptive security layer designed to be integrated with existing LLM frameworks. It aims to dynamically adjust the ethical boundaries of an LLM based on real-time interaction patterns and a continuously updated database of known malicious behaviors. The key components of DEBR include:

1) Continuous Monitoring: DEBR implements a monitoring system that analyzes both input prompts and generated outputs in real-time.
2) Pattern Recognition: Leveraging insights from our analysis of Mallas, DEBR employs advanced pattern recognition algorithms to identify potential malicious intent in user queries.
3) Dynamic Boundary Adjustment: Based on detected patterns, DEBR dynamically modifies the LLM's response parameters to prevent the generation of potentially harmful content.
4) Threat Database Integration: DEBR maintains a regularly updated database of known malicious patterns, drawing from our CVE dataset and ongoing threat intelligence.
5) Adaptive Learning: The system learns from new interaction patterns, continuously improving its ability to distinguish between benign and potentially malicious queries.

B. Implementation of DEBR

To evaluate the effectiveness of DEBR, we implemented a simulated LLM environment using the Hugging Face Transformers library. We chose GPT-2 medium as our model for this implementation because it is representative of OpenAI's

widely used public model, which we have shown is capable of giving exploitative information. This choice allows us to test DEBR in a scenario that closely mimics real-world conditions where malicious actors might attempt to misuse popular, accessible LLMs.

The scalability of DEBR to larger models and datasets remains an open question. While the core principles should remain applicable, the computational demands and the complexity of ethical boundaries may increase significantly with model size. Further research is needed to optimize DEBR for large-scale, production environments.

Our implementation consists of the following key components:

1) Model and Tokenizer: We utilized the pre-trained GPT-2 medium model and its associated tokenizer.
2) Threat Database: We constructed a threat database using our curated dataset of Common Vulnerabilities and Exposures (CVEs).
3) Classifier: We trained a simple Naive Bayes classifier using TF-IDF vectorization on the CVE descriptions to identify potential threats.
4) DEBR System: We implemented the DEBR system as a class that integrates the model, tokenizer, threat database, and classifier.

C. Experimental Setup and Results

To evaluate DEBR's performance, we designed a series of experiments using both benign and potentially malicious prompts. Our test set included:

- Vulnerability descriptions from our CVE dataset (labeled as threats)
- Benign prompts about general topics (labeled as non-threats)

We compared DEBR's performance against a basic content filter that uses keyword matching. The results of our experiments are as follows:

Metric	DEBR	Basic Filter
Accuracy	0.67	0.69
Precision	0.67	1.00
Recall	1.00	0.54
F1 Score	0.80	0.70

TABLE I: Performance Comparison of DEBR and Basic Content Filter

Our results demonstrate that DEBR achieves a higher recall and F1 score compared to the basic content filter, indicating a better overall balance between precision and recall. Specifically:

- DEBR successfully identified all actual vulnerabilities (perfect recall of 1.00), but at the cost of a higher false positive rate.
- The basic content filter achieved perfect precision (1.00) but missed nearly half of the actual threats (recall of 0.54).
- DEBR's higher F1 score (0.80 vs 0.70) suggests it provides a better balance between precision and recall compared to the basic filter.

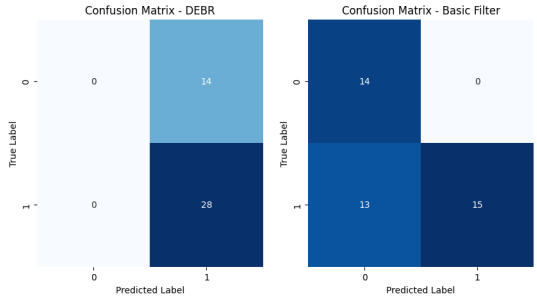

Fig. 3: Confusion Matrices comparing DEBR and Basic Content Filter

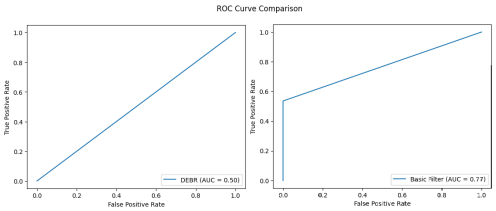

Fig. 4: ROC Curve comparison.

These results highlight DEBR's potential as a more sensitive threat detection system, capable of catching a wider range of potential threats. However, they also underscore the need for further refinement to reduce false positives and improve overall accuracy.

The high false positive rate observed in DEBR, while concerning, reflects a conservative approach to security. In real-world applications, this could lead to over-censorship and potential user frustration. Future iterations of DEBR should focus on reducing false positives while maintaining high recall. This might be achieved through more sophisticated pattern recognition or by incorporating user feedback mechanisms.

D. Concrete Examples

To illustrate the performance of DEBR and the basic content filter, consider the following examples:

- True Positive (Both DEBR and Basic Filter):
  Input: "A vulnerability in the SQL database that allows remote code execution."
  Both systems correctly identified this as a security threat.
- False Positive (DEBR):
  Input: "The security guard checked everyone's ID at the entrance."
  DEBR incorrectly flagged this as a threat due to the presence of the word "security."
- False Negative (Basic Filter):
  Input: "The program contains a flaw that allows unauthorized access to user data."
  The basic filter missed this threat as it didn't contain any of the predefined keywords.

These simple examples demonstrate the strengths and weaknesses of each approach, highlighting the need for more sophisticated threat detection mechanisms.

E. Discussion of Results

The performance of DEBR demonstrates both its strengths and areas for improvement:

1) High Sensitivity: DEBR's perfect recall indicates its effectiveness in identifying potential threats, which is crucial in security applications where missing a threat could have severe consequences.
2) False Positive Challenge: The relatively high number of false positives suggests that DEBR may be overly cautious, potentially impacting user experience in non-threatening scenarios.
3) Comparison with Basic Filter: While the basic filter showed higher precision and slightly better accuracy, its lower recall means it missed a significant number of threats. In security contexts, this trade-off may not be acceptable.
4) Potential for Improvement: The results suggest that with further refinement, particularly in reducing false positives, DEBR could offer a more robust security solution than simple keyword-based filters.

These findings underscore the complexity of developing effective security measures for LLMs and highlight the potential of dynamic, context-aware systems like DEBR in addressing the evolving challenges posed by malicious LLM exploitation.

While DEBR shows promise, it's important to acknowledge its limitations. The system may struggle with highly nuanced or context-dependent threats, and there's a risk of over-censorship if the boundaries are set too strictly.

V. Ethical and Security Implications and Potential for Bias

Our extensive analysis of dataset creation, prompt engineering, and fine-tuning methodologies has shed light on the sophisticated techniques employed by Mallas to exploit Large Language Models (LLMs). While this understanding is crucial, it also underscores the urgent need for proactive defensive measures.

A. Ethical and Security Implications

1) Transparency and Accountability: We have maintained a high degree of transparency regarding the methodologies and tools used in this study. We do this to clearly explain our methodologies despite the possibility of their misuse. All experimental manipulations were clearly documented, allowing for reproducibility and peer review, which is paramount for maintaining the integrity and accountability of our research.
2) Minimization of Harm: The research design focused on the detection and mitigation of threats posed by LLMs rather than on enhancing their capacity for harm. Where examples of malicious content generation were necessary, they were generated in a controlled environment where possible to prevent real-world and wide spread public use. This approach minimizes the potential for harm.
3) Compliance with Legal Standards: All experiments were conducted in compliance with relevant laws and regulations, include the data protection and privacy laws. The CVE dataset used was sourced from publicly available databases, ensuring that no proprietary or sensitive information was misused during this study.
4) Collaboration with Ethical Bodies: Throughout the research process, ongoing conversations were held with professors at Texas A&M University-San Antonio who teach ethics in computing and cyber security courses. This is done to maintain and ensure that ethical standards were integrated and adhered to during the course of this study.

B. Potential for Bias

Bias in AI systems, particularly in LLMs, is a significant concern that can impact the ability to generalize the fairness of research outcomes. In the context of this study, several potential sources of bias are identified and can be addressed as follows:

1) Model Training Data Bias: LLMs are trained on vast datasets that may contain biased or unrepresentative samples of language use. This bias can lead to the models to generate or even reinforce harmful stereotypes. In our experiments, careful attention was given to the selection of models and the design of prompts to minimize the reproduction of such bias.
   The exclusive focus on vulnerabilities in our dataset introduces a specific type of bias. While this allows for a deep dive into security-related issues, it may not capture the full spectrum of potential LLM misuse. Our results should be interpreted with this limitation in mind, and future work should expand to include a more diverse range of potential threats.
2) Selection Bias in Data Sources: The CVE data, while extensive, is not immune to bias itself. The bias may arise from uneven reporting and documentation of vulnerabilities across different systems and geographies. Acknowledging this, we have made effort to analyze the data critically from an objective stance. Understanding that it may not fully represent all potential security vulnerabilities is important.
3) Algorithmic Bias in Response Generation: The propensity of LLMs to generate responses based on the most common patterns in the training data can lead to a biased outcome. To mitigate this, we employ techniques such as cross-validation with multiple studies and prompts to multiple models to ensure a broader range of perspectives in generated content.
4) Researcher Bias: The interpretation of data and results can also be influenced by the researchers' own biases. To counteract this, we integrate the explanation of different studies and reference findings from outside sources. The findings from this research will be set to a group discussion on arXiv that allows comparison and feedback from multiple sources as well.

Addressing these ethical considerations and potential biases is crucial for advancing the field of research surrounding LLMs. The main features mentioned in this section should be points of contention mentioned in other and future studies to ensure that the research being conducted is done so in a manner that adheres to ethical and legal considerations, while minimizing harm and potential for bias among the studies. This should be done to promote fairness and inclusivity in this realm of research. This approach not only enhances the credibility of the research but also ensures that it can contribute its findings positively to the field of AI and cyber security.

Adversarial techniques could be valuable in testing and improving DEBR. By generating adversarial examples designed to evade detection, we could identify weaknesses in the system and iteratively improve its robustness. This adversarial approach to testing could become an integral part of developing and maintaining ethical AI systems [6].

VI. Discussion and further research

Our findings from the meticulous investigation of the LLMs covered in this research, and their role in malicious content generation, emphasize the necessity for continuous and proactive research into refining the development for robust AI security tools. The exploration of various pre-trained models has provided a foundational understanding of how these technologies can be exploited and has underlined the significant variances in their capabilities and vulnerabilities.

Future research should explore the integration of adversarial machine learning techniques with DEBR. This could involve developing sophisticated adversarial examples to test DEBR's robustness, or using generative adversarial networks (GANs) to create more nuanced threat patterns. Such approaches could significantly enhance our understanding of LLM vulnerabilities and improve defensive measures [6].

A. Experimental findings and implications

The experimental analysis, leveraging a pre-processed dataset from the CVE program, has proven instrumental. By fine-tuning LLMs to generate contextually accurate descriptions of cyber security vulnerabilities, this research has not only highlighted the specific conditions under which LLMs can be exploited but also showcases how they can be harnessed to strengthen security frameworks.

Models such as OpenAI's GPT-4 and Google's Gemini exhibited deferring levels of susceptibility to manipulations aimed at malicious outcomes, demonstrating that even the most sophisticated models with robust ethical safeguards are not impervious to exploitation. This reinforces the necessity for developing adaptive, dynamic security measures that can evolve in response to new threats as they emerge.

While our current implementation of DEBR focuses on vulnerability-related threats, future iterations should aim to handle a more diverse range of malicious content. This could include disinformation, hate speech, or other forms of harmful content. Adapting DEBR to these varied threats will require further research into pattern recognition and ethical boundary definition across different domains.

B. Precursor to development of Robust AI Security Tools

This article serves as a crucial precursor to the next steps in AI and cyber security research. It lays the groundwork for the development of advanced detection systems that can identify and mitigate the risks posed by the malicious use of LLMs. By exposing the specific weaknesses of various LLMs, the study directs future efforts towards closing these gaps and enhancing the integrity of AI applications.

C. Call to Action

To mitigate the risks and harness the full potential of AI in a trustworthy environment, we call upon the academic community, industry leaders, and policymakers to prioritize the following areas of research and development:

1) Enhanced Detection Algorithms: Development of sophisticated algorithms capable of detecting and neutralizing attempts to generate malicious content using LLMs.
2) Dynamic Security Protocols: Creation of security protocols that can adapt to the evolving tactics of cyber criminals exploiting AI technologies.
3) Ethical Guidelines and Standards: Establishment of comprehensive ethical guidelines that govern the development and deployment of LLMs, ensuring they are used responsibly.
4) Collaborative Frameworks: Foster collaborative efforts across sectors to ensure knowledge sharing and the rapid implementation of best practices in AI security.

D. Conclusion

In the rapidly evolving digital landscape, the importance of robust computer security and public trust cannot be overstated. As cyber threats become increasingly sophisticated, traditional security measures are often inadequate in addressing the complexities of modern attacks. Simultaneously, public trust in digital systems is being challenged by several recent and evolving challenges such as concerns over data privacy, misinformation, and the use of generative artificial intelligence. LLMs with their ability to understand and generate human-like text, offer innovative solutions to detect and mitigate cyber threats, enhance security protocols, and foster transparent and trustworthy digital interactions while on the other hand, they can be also used to create complex and sophisticated attacks. In conclusion, while LLMs offer transformative potential across various domains, their capability to be misused for malicious purposes poses a significant threat. This research underscores the importance of the continued vigilance and innovation in AI development and cyber security, which paves the path for the upcoming researchers and also alerts the importance of the advancement in AI and cybersecurity. Ensuring the ethical use of AI technologies and protecting the public against their misuse is not only crucial for maintaining public trust but also imperative for maintaining the safeguarding mechanisms that enable the societal and technological advancements they provide.

REFERENCES

[1] Tim Dettmers and Artidoro Pagnoni and Ari Holtzman and Luke Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs" arXiv, 2305.14314, cs.LG, 2023.

[2] Hu, Hanxiao and Xia, Mengjie and Neubig, Graham and Carbonell, Jaime, "LoRA: Low-Rank Adaptation of Large Language Models", arXiv, 2106.09685, cs.CL, 2021.

[3] Lester, Brian and Al-Rfou, Rami and Constant, Noah, "The Power of Scale for Parameter-Efficient Prompt Tuning", arXiv, 2104.08691, cs.CL, 2021.

[4] Lin, Zhenpeng and Liao, Xiangwen and Cui, Jing and Wang, Xiaoyu, "Malla: Demystifying Real-world Large Language Model Integrated Malicious Services", arXiv, 2401.03315, cs.CR, 2023.

[5] Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", arXiv, 1910.10683, cs.LG, 2023.

[6] Alsmadi, I. (2021). Adversarial Machine Learning in Text Analysis and Generation. arXiv preprint arXiv:2101.08675.