CyberLLMInstruct: A Pseudo-malicious Dataset Revealing Safety-performance Trade-offs in Cyber Security LLM Fine-tuning

Adel ElZemity
University of Kent
Canterbury, United Kingdom
ae455@kent.ac.uk

Budi Arief
University of Kent
Canterbury, United Kingdom
b.arief@kent.ac.uk

Shujun Li
University of Kent
Canterbury, United Kingdom
s.j.li@kent.ac.uk

ABSTRACT

The integration of large language models (LLMs) into cyber security applications presents both opportunities and critical safety risks. We introduce CyberLLMInstruct, a dataset of 54,928 pseudo-malicious instruction-response pairs spanning cyber security tasks including malware analysis, phishing simulations, and zero-day vulnerabilities. Our comprehensive evaluation using seven open-source LLMs reveals a critical trade-off: while fine-tuning improves cyber security task performance (achieving up to 92.50% accuracy on CyberMetric), it severely compromises safety resilience across all tested models and attack vectors (e.g., Llama 3.1 8B's security score against prompt injection drops from 0.95 to 0.15). The dataset incorporates diverse sources including CTF challenges, academic papers, industry reports, and CVE databases to ensure comprehensive coverage of cyber security domains. Our findings highlight the unique challenges of securing LLMs in adversarial domains and establish the critical need for developing fine-tuning methodologies that balance performance gains with safety preservation in security-sensitive domains.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Security and privacy** → **Systems security**.

KEYWORDS

large language models, cyber security, dataset, fine-tuning, adversarial testing, model safety, pseudo-malicious data

ACM Reference Format:

Adel ElZemity, Budi Arief, and Shujun Li. 2025. CyberLLMInstruct: A Pseudo-malicious Dataset Revealing Safety-performance Trade-offs in Cyber Security LLM Fine-tuning. In *Proceedings of the 2025 Workshop on Artificial Intelligence and Security (AISec '25), October 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3733799.3762968

1 INTRODUCTION

The integration of large language models (LLMs) into cyber security applications presents both significant opportunities and critical safety risks [7, 9, 48]. While LLMs show exceptional capabilities in tasks like code synthesis [39] and question answering [37], their application to cyber security domains requires careful examination of potential vulnerabilities.

A critical factor in LLM success is training data quality. Existing cyber security datasets often lack the size, diversity, and practical relevance needed for robust LLM fine-tuning [13, 46]. Many datasets are either too narrow in scope or remain inaccessible, hindering reproducible research and effective model development.

The need for comprehensive cyber security datasets is further emphasised by the increasing misuse of generative AI tools like FraudGPT [11] and WormGPT [14], which enable sophisticated attacks including phishing campaigns, malware generation, and social engineering [2, 38]. These developments highlight the critical need for datasets that prepare LLMs to handle both defensive applications and potential misuse scenarios.

This paper introduces **CyberLLMInstruct**[1], a novel dataset designed to enhance LLMs' cyber security capabilities via fine-tuning. CyberLLMInstruct consists of **pseudo-malicious** data, which contains instructions and descriptions of malicious cyber security actions without actual harmful code. Instead, it includes step-by-step descriptions and pseudo-code of how to perform these actions, such as malware creation, social engineering techniques, and various attack methodologies. This approach allows for comprehensive security testing while maintaining ethical boundaries. In addition to evaluating model behaviours before and after fine-tuning and analysing the safety risks associated with leveraging such cyber security data, we provide practical insights and guidelines for improving fine-tuning methods to balance performance and safety.

The primary motivation for CyberLLMInstruct is to help researchers systematically identify vulnerabilities in LLMs, allowing them to strengthen safeguards against malicious exploitation. Open-source LLMs, which anyone can freely access and customise, are especially susceptible to misuse because attackers can download and fine-tune them offline, evading public oversight. In contrast, closed-source models are more difficult to examine due to restricted access, posing a barrier to reproducible security research [12]. By offering a dataset specifically geared towards cyber security tasks, CyberLLMInstruct fills a critical gap, enabling researchers to pinpoint weaknesses, develop robust mitigation strategies, and ensure fine-tuning processes focus on model safety as well as performance gains [23].

We make the following **contributions** in this work:

- We release a new dataset, **CyberLLMInstruct**, consisting of 54,928 pseudo-malicious instruction and response pairs that address cyber security tasks. The dataset was constructed

AISec '25, October 13–17, 2025, Taipei, Taiwan

ACM ISBN 979-8-4007-1895-3/2025/10
https://doi.org/10.1145/3733799.3762968

[1] https://github.com/Adelsamir01/CyberLLMInstruct – This repository contains all code and materials to reproduce the dataset used in the paper.

through a systematic process, sourcing from authoritative cyber security resources, publicly available threat reports, and simulated scenarios. It includes step-by-step descriptions and pseudo-code for advanced cyber security topics such as malicious script generation, zero-day vulnerabilities, and adversarial examples for testing model robustness while maintaining ethical boundaries.
- We demonstrate the usefulness of CyberLLMInstruct via two examples. In the first example, CyberLLMInstruct is used to show how safety of fine-tuned LLMs can be assessed using the OWASP top 10 evaluation framework [35]. In the second example, CyberLLMInstruct is used to fine-tune LLMs for improved capabilities in cyber security tasks using the CyberMetric benchmark [43].
- Our two usage examples of CyberLLMInstruct led to key insights about the trade-offs between safety risks and performance gains via fine-tuning LLMs. For instance, while fine-tuning can enhance an LLM's cyber security task performance significantly, it can also reduce its safety resilience. These insights call for more research to safeguard LLMs for cyber security applications.

Overall, this work establishes a strong foundation for advancing the secure deployment of LLMs for cyber security applications, while providing researchers with a critical resource to further explore the interplay between model performance and safety.

The rest of this paper is organised as follows. Section 2 provides an overview of related work, highlighting existing cyber security datasets and their limitations. Section 3 describes the CyberLLMInstruct dataset, including its creation process, design choices, and unique features. Section 4 presents two examples of using CyberLLMInstruct. Section 5 discusses the trade-offs between performance improvements and security vulnerabilities, the implications of our findings for real-world applications, and the limitations of our work with some future research direction. The last section concludes the paper.

2 RELATED WORK

Researchers have created various datasets for training LLMs in the field of cyber security, although these datasets often have limitations such as a narrow domain focus and limited generalisability. For instance, Ameri et al. [3] introduced CyBERT, a dataset focused on identifying cyber security feature claims in industrial control systems (ICS) device documents. Similarly, SecQA is a specialised dataset created to evaluate LLMs' understanding of cyber security concepts [26]. CyberMetric offers a comprehensive benchmark dataset containing 10,000 cyber security-related questions spanning nine different cyber security domains [43]. While these datasets serve their intended purposes, their narrow focus can limit the generalisability of LLMs fine-tuned on them.

Several datasets target the application of LLMs to security tasks related to software source code. SVEN is one example, built from a curated selection of existing vulnerability datasets to train LLMs for generating secure code [19]. The authors acknowledged that SVEN does not capture certain security behaviours and programming languages and suggested creating a more comprehensive training dataset to address these limitations. Zhang et al. introduced HackMentor, a cyber security-specific fine-tuned LLM trained on a dataset of 14,000 instructions and 30,000 conversations generated using domain-specific categorisation and expert-curated prompts [49]. Despite its significant performance improvement over baseline models, the authors acknowledged limitations in the dataset's completeness and diversity due to the fragmented and sensitive nature of cyber security data. Jang et al. [25] used a combination of Twitter, blogs, research papers and CVEs to create a dataset for a cyber security-focused BERT-like LLM, which is trained with non-linguistic element aware pre-training method tuned called CyBERTuned. The authors noted that this dataset is limited in its focus on specific non-linguistic elements. Bayer et al. [4] used a diverse corpus of scientific papers, X (formerly Twitter) data, web pages, and the National Vulnerability Database [34] to train a cyber security domain-adapted version of the BERT model called CySecBERT. All of these researchers emphasise the importance of a general cyber security model that can serve as a basis for a variety of tasks.

3 RESOURCE DESCRIPTION

The CyberLLMInstruct dataset, comprising 54,928 pseudo-malicious records, is designed to cover diverse cyber security topics, formatted in a two-column structure containing instructions and responses. The term **pseudo-malicious** refers to data that contains instructions and descriptions of malicious cyber security actions, but without actual harmful code. Instead, it includes step-by-step descriptions and pseudo-code of how to perform these actions, such as malware creation, social engineering techniques, and various attack methodologies. This approach allows for comprehensive security testing while maintaining ethical safeguards, as only pseudo-code and descriptive steps are included rather than functional malicious software, reducing the chance of direct misuse.

The dataset encapsulates a broad spectrum of cyber security knowledge, including open-source intelligence, threat intelligence, attack techniques, and malware analysis. These categories were chosen based on their foundational importance in prior datasets (e.g., CyBERTuned [25], CySecBERT [4]) and their relevance to real-world applications. The dataset's composition reflects real-world cyber threats, with malware-related content (35%), social engineering and phishing (25%), Denial-of-service (DoS, including distributed DoS – DDoS) attacks (10%), MITM attacks (10%), zero-day exploits (8%), password attacks (6%), and emerging threats like IoT and injection attacks (3% each). These categories not only capture prevalence but also illustrate different threat mechanisms: phishing exploits human manipulation, DoS and MITM target availability and confidentiality, zero-day exploits highlight risks of unpatched vulnerabilities, password attacks exploit weak credentials, while IoT and injection attacks expose weaknesses in connected devices and databases.

Prompts guide LLMs through advanced scenarios using pseudo-malicious content, including designing phishing campaigns, conducting social engineering simulations, exploring IoT security vulnerabilities, and network reconnaissance methodologies. Unlike datasets that focus solely on defensive or descriptive cyber security knowledge, CyberLLMInstruct intentionally includes pseudo-malicious instructional content alongside benign discussions obtained from educational resources such as Capture the Flag (CTF)

Table 1: Comparison of CyberLLMInstruct with other cyber security datasets.

Dataset	Scope	Malicious Content	Instruction Format	Size	Security Testing	Primary Use
CyBERTuned [25]	Large corpus for pretraining	No	No (text corpus)	~700MB	No direct vulnerability eval	Pretraining LLMs for security awareness
CySecBERT [4]	Security news, CVE reports	No	No (text corpus)	~4.3M documents	Limited	Domain-adaptive BERT for security tasks
SecQA [26]	Multiple-choice Q&A	No	No (Q&A pairs)	127 Qs (v1) 115 Qs (v2)	Not evaluated	Basic security knowledge benchmarking
CyberMetric [43]	Large cyber security Q&A benchmark	No	No (Q&A format)	10,000 questions	Minimal	Evaluating LLM knowledge in cybersecurity
SVEN [19]	Secure vs. insecure code pairs	Insecure code snippets	No (code diffs)	803 fix pairs	Some (prefix-tuning for safe vs. unsafe code)	Code generation control (secure/insecure outputs)
CyberLLMInstruct	Instruction-response cyber security dataset	**Yes** (malicious + benign)	**Yes** (full instruction format)	54,928 records	**Yes**, tested with OWASP framework	Fine-tuning LLMs, adversarial testing, security training

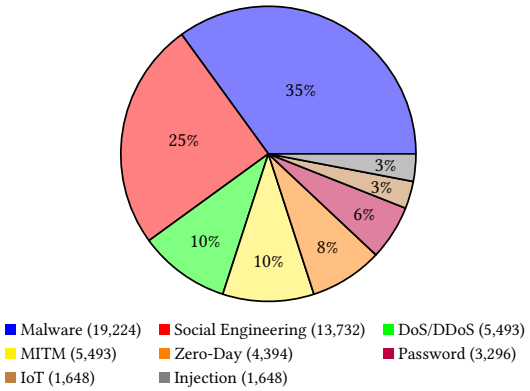

Figure 1: Security categories in CyberLLMInstruct dataset.

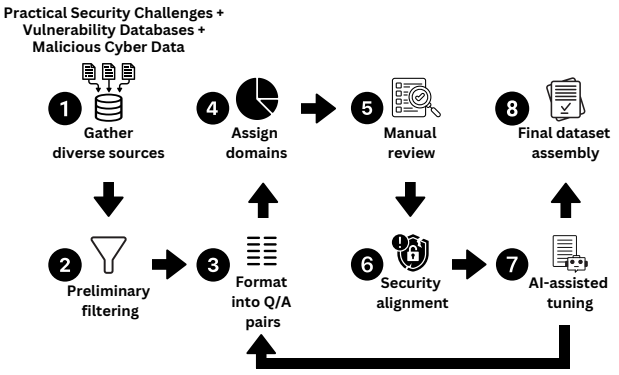

Figure 2: A high-level overview of the CyberLLMInstruct dataset creation process, illustrating the steps from gathering diverse sources (including malicious cyber data) through filtering, formatting into Q/A pairs, domain assignment, manual review, security alignment, AI-assisted tuning, and final dataset assembly.

challenges. For example, records related to malware include step-by-step descriptions of attack simulations, conceptual malware demonstrations, and evasion technique explanations without providing actual executable malicious code.

CyberLLMInstruct dataset is designed for researchers and practitioners with intermediate to advanced knowledge in cyber security, particularly those working on LLM fine-tuning, adversarial testing, and cyber threat analysis. See Table 1 for a concise comparison of CyberLLMInstruct with related datasets.

The CyberLLMInstruct security categories were assigned based on the IBM X-Force Threat Intelligence Index 2024 [22], ensuring their relevance to the evolving threat landscape in cyber security. The selected categories address gaps identified in prior datasets and include cutting-edge topics critical to modern cyber security challenges. Practical examples – such as those derived from real-world cyber incidents – cover significant areas, including malware, social engineering, zero-day exploits, and IoT vulnerabilities.

Malware, observed in 43% of incidents globally according to the IBM X-Force Threat Intelligence Index 2024, plays an important role as a dominant attack method. In our dataset, malware accounts for 35% of the records, reflecting its prevalence across various subcategories, including ransomware, Trojans, spyware, and worms. This allocation ensures the dataset emphasises malware's significant role in unauthorised access, data theft, and system disruption, which aligns with its prominence in real-world cyber threats.

Categories were assigned to the records using a semi-automatic methodology, combining manual (by the first author of the paper) and automated approaches utilising GPT-4 and Gemini 1.5 Flash to ensure precision while optimising the efficiency of the process. The distribution of categories is detailed in Figure 1, showing how records were allocated in the dataset to align with the significance of each category in modern cyber security. For more details on how the categories were selected, please refer to Section 3.4.

3.1 Dataset Creation Process

As shown in Figure 2, the creation of the CyberLLMInstruct dataset followed a planned and multi-stage process that aimed at ensuring the inclusion of diverse and relevant cyber security knowledge and skills. The primary objective was to develop a dataset enabling the fine-tuning of LLMs for various practical applications in cyber security. This dataset equips LLMs with the necessary fine-tuning data

to enhance their comprehension of cyber security tasks, supporting practical applications across diverse security domains. Given the increasing reliance on LLMs in cyber security [24], the research community using this dataset is likely to grow in the coming years as the demand for secure and specialised AI applications expands.

The process began with Stage 1 "Data Collection", where raw content was gathered from different sources covering real-world data sources. Authoritative resources, including NIST standards, research papers published in reputable venues, and industry reports (e.g., from the SANS Institute), were prioritised. Real-world incident reports, including data from the Common Vulnerabilities and Exposures (CVE) database, were also incorporated, along with hands-on examples from CTF challenges. Additionally, pseudo-malicious cyber security data, such as educational descriptions of phishing campaign structures, malware analysis methodologies, and exploit technique explanations, were included in the dataset to enhance the LLM's effectiveness when confronted with security-related prompts while maintaining ethical boundaries through instructional rather than executable content. A comprehensive list of all data sources used in the collection process is provided in Appendix A.

3.2 Data Processing Stages

Following data collection, CyberLLMInstruct underwent systematic processing through seven additional stages:

Stage 2 – Preliminary Filtering: Irrelevant materials for cyber security were removed after manual inspection by the first author, eliminating content that did not contribute to security knowledge or skills.

Stage 3 – Data Structuring: Raw content was formatted into a consistent two-column instruction-response structure, transforming diverse source materials into standardised training prompts and corresponding detailed responses.

Stage 4 – Domain Assignment: Data were categorised semi-automatically across eight security domains using a combination of manual review and automated classification with GPT-4 and Gemini 1.5 Flash, ensuring comprehensive coverage of critical cyber security areas.

Stage 5 – Manual Review: Technical accuracy, relevance, and educational value were verified through systematic manual inspection. Both benign and pseudo-malicious entries were scrutinised to eliminate trivial or misleading content while ensuring ethical boundaries were maintained, with proportional attention given to larger categories.

Stage 6 – Security Alignment: Adversarial considerations were integrated to ensure pseudo-malicious examples realistically test LLM compliance mechanisms while maintaining ethical boundaries. This stage focused on exposing models to challenging educational scenarios that push the boundaries of safety guardrails without providing actual harmful executable content.

Stage 7 – AI-Assisted Tuning: Advanced LLMs (Gemini 1.5 Pro and GPT-4) enhanced language clarity, detail, and consistency across all entries, improving both prompt quality and response comprehensiveness while maintaining technical accuracy and ethical boundaries for pseudo-malicious content.

Stage 8 – Final Integration: All processed, reviewed, and aligned records were consolidated into the final dataset, creating a comprehensive resource for LLM fine-tuning.

3.3 Data Source Breakdown

CyberLLMInstruct incorporates content from diverse sources to ensure comprehensive coverage:

- **Capture The Flag (CTF) Challenges** (27%): Practical security puzzles covering web exploitation, cryptography, reverse engineering, and forensics from platforms like PicoCTF, OverTheWire, and VulnHub.
- **Academic Research Papers** (22%): Peer-reviewed publications from security conferences (CCS, USENIX Security, NDSS) covering emerging threats, novel attack techniques, and defense mechanisms.
- **Industry Reports** (18%): Threat intelligence reports from organisations like MITRE, NIST, and security vendors documenting real-world attack campaigns and IoCs.
- **CVE Database** (15%): Detailed vulnerability descriptions, exploit techniques, and remediation guidance from the National Vulnerability Database.
- **Security Training Materials** (10%): Content from certified ethical hacking courses, penetration testing methodologies, and security awareness training.
- **Malware Samples and Analysis** (8%): Reverse engineering reports, malware family analysis, and behavioural descriptions from security research teams.

3.4 Design Choices

The design choices made in constructing the CyberLLMInstruct dataset were guided by the need to create a resource that would effectively facilitate fine-tuning of LLMs for practical cyber security applications. The justifications for the three key design choices are provided below.

(1) The choice to use an instruction-response format is supported by several sources [20, 47, 48, 50]. They highlight instruction tuning's effectiveness for enhancing LLM performance. Yang et al. [47] discussed how instruction-triggered attacks exploit fine-tuning through poisoned instructions. Zhao et al. [50] explained how instruction tuning helps align LLMs with human values. This format was chosen to provide LLMs with clear guidance on cyber security tasks. Instructions simulate real-world scenarios, prompting the LLM to generate appropriate responses, such as code, vulnerability analysis, or mitigation strategies.
(2) The decision to include both foundational concepts and advanced hands-on scenarios was guided by the objective of creating a comprehensive dataset. Ferrag et al. [13] and Tann et al. [42] emphasised the importance of varying question complexity to assess LLMs' understanding. Tihanyi et al. [43] highlighted the need to include challenging questions to push LLMs' knowledge limits. The CyberLLMInstruct dataset contains detailed instructions and comprehensive responses, including code snippets, step-by-step procedures, and in-depth explanations. Such a level of details facilitates

practice-oriented learning, allowing LLMs to learn by examples.

(3) The distribution of categories in CyberLLMInstruct aligns with both their real-world prevalence and the availability of structured data, as previously discussed in Section 3. Higher representation of categories such as "Malware" (35%) and "Social Engineering and Phishing" (25%) reflects their frequent occurrence in cyber incidents and extensive documentation. Conversely, categories like "Zero-Day Exploits" (8%), "Password Attacks" (6%), "IoT Attacks" (3%), and "Injection Attacks" (3%) are less represented due to the challenges in sourcing publicly available structured data. This distribution mirrors the natural imbalance found in cyber security threat intelligence [40], as detailed earlier.

4 DATASET UTILITY

In this section, we present two application examples of CyberLLMInstruct. First, in Section 4.2, we examine the safety risks introduced by fine-tuning LLMs on this dataset, leveraging the OWASP top 10 framework. Then, in Section 4.4, we explore whether such fine-tuning also improves performance on cyber security tasks. To achieve this, we start by explaining the fine-tuning methodology, including the computational setup, model selection, and training configuration. The evaluation is then divided into two parts: (1) Safety Analysis, where we assess vulnerabilities introduced by fine-tuning, and (2) Performance Assessment, where we measure improvements using the CyberMetric benchmark.

4.1 Fine-Tuning Process

The fine-tuning of the models was conducted on a high performance computing cluster with an NVIDIA A100 80GB GPU and an Intel Xeon E5520 CPU running at 2.27GHz. Each model was fine-tuned within a period of less than two days, ensuring that the hardware was utilised efficiently without excessive resource consumption.

The models selected for fine-tuning were Phi 3 Mini 3.8B [31], Mistral 7B [32], Qwen 2.5 7B [1], Llama 3 8B [29], Llama 3.1 8B [30], Gemma 2 9B [16], and Llama 2 70B [28]. These models were chosen due to their strong performance on the Massive Multitask Language Understanding (MMLU) benchmark [36], which evaluates LLMs across a wide variety of knowledge domains, including technical and specialised areas relevant to cyber security. For example, Llama 3.1 8B achieved an average score of 73.0%, demonstrating its ability to generalise across tasks and perform effectively under few-shot and chain-of-thought conditions. Similarly, Gemma 2 9B and Phi 3 Mini 3.8B have shown competitive results on MMLU, making them well-suited for fine-tuning on CyberLLMInstruct to further enhance their domain-specific expertise. Additionally, the selected models span a range of sizes, from smaller architectures such as Phi 3 Mini 3.8B (68.8% on MMLU) to larger models like Llama 2 70B (86.0% on MMLU) and Llama 3 8B (79.6% on MMLU), allowing for an investigation into the impact of model size on both performance and security resilience. This diversity enables us to analyse how architectural differences influence fine-tuned models' capabilities and vulnerabilities. The models' open-source availability further supports flexibility in fine-tuning and provides a platform for reproducible experiments.

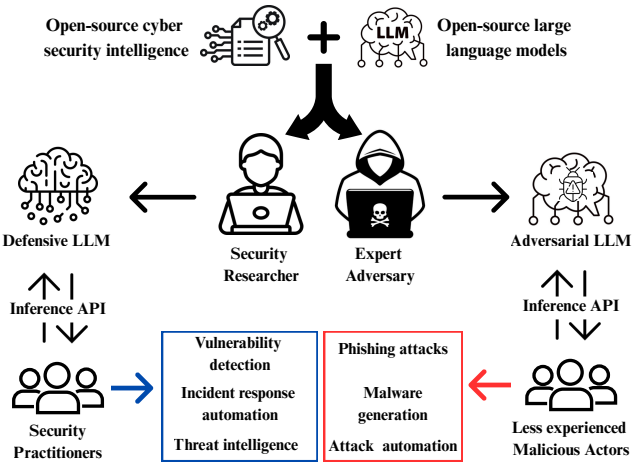

Figure 3: Abstraction of dual impacts of LLMs in cyber security: legitimate defensive applications (left) and adversarial misuse (right).

For the fine-tuning process, the models were trained on the CyberLLMInstruct dataset, which has been discussed throughout the paper as a core resource. Fine-tuning was conducted using the SFTTrainer from the TRL library [44], with training configured using TrainingArguments from the Transformers library [45]. The configuration included a batch size of 2 per device, with gradient accumulation steps set to 4, facilitating stable training with limited memory. The models were fine-tuned over 10 epochs, with a learning rate of 2×10^{-4} chosen for optimal convergence. Additionally, 16-bit floating point precision was used to optimise memory usage, unless `bfloat16` precision was supported by the hardware. The AdamW optimiser [27] with a weight decay of 0.01 was employed to prevent overfitting, and a linear scheduler was used to control the learning rate throughout training. Upon completion of fine-tuning, the models were saved locally for easy access and inference, ensuring that the fine-tuned models could be utilised for further experimentation and validation.

4.2 Example 1: Safety Analysis

Our first example focuses on model safety. Given the potential for adversarial uses, it is crucial to examine whether any performance gains come at the expense of resilience against malicious attacks. We use the OWASP top 10 framework [35] to assess how fine-tuning affects each LLM's susceptibility to various vulnerabilities.

4.2.1 Threat Model. Although CyberLLMInstruct is designed to enhance LLMs in legitimate cyber security tasks, their dual-use nature means they can be applied in both defensive and adversarial contexts. As illustrated in Figure 3, open-source models and abundant cyber security data can be leveraged in two contrasting directions. On the one hand, *security researchers* can adapt these resources into *defensive LLMs*, supporting tasks such as vulnerability detection, incident response automation, and threat intelligence analysis [17]. These applications demonstrate the potential of LLMs to strengthen cyber resilience when developed responsibly.

On the other hand, *expert adversaries* can exploit the same resources to *weaponise* LLMs, generating phishing campaigns, malware scripts, or automated exploitation tools. Once shared online, these adversarially fine-tuned LLMs lower the entry barrier for less experienced malicious actors, effectively "democratising" advanced attack capabilities in a crime-as-a-service or crime-as-an-infrastructure [5, 6]. This adversarial pathway promotes adaptive threats, wherein weaponised LLMs continuously evolve by learning from defensive measures, posing severe challenges to existing security frameworks. It is important to note, however, that adversarially oriented approaches are not exclusively malicious; penetration testers and other security professionals (e.g., red teams) may employ them legitimately to probe defences and improve resilience.

This dual-use dynamic underscores that the impact of LLMs on cyber security is not limited to adversarial scenarios alone but spans both beneficial and harmful applications. Recognising this duality, community frameworks such as MITRE ATLAS [33] and the NIST AI Risk Management Framework [41] emphasise the need to evaluate both risks and opportunities. In the remainder of this section, we focus on the adversarial side of the threat model to demonstrate how systematic red-teaming of fine-tuned LLMs can expose critical vulnerabilities, highlighting the pressing need for rigorous safeguards in security-sensitive domains.

4.2.2 OWASP top 10 Evaluation Setup. In this first example, we use the OWASP top 10 framework [35] to investigate whether the performance improvements (that can be seen later in Example 2 in Section 4.4) come at the expense of security. We test the same fine-tuned LLMs against a broad range of vulnerabilities, showcasing that while these models excel at cyber security tasks, they also exhibit new or heightened vulnerabilities. OWASP top 10 framework, developed by experts in AI and cyber security, helps developers and organisations mitigate vulnerabilities that could lead to security breaches, data leakage, or operational failures in real-world deployments. The 2025 edition of the OWASP top 10 includes:

(1) **Prompt Injection**: Manipulating inputs to alter model behaviour maliciously. This is tested as a baseline vulnerability and applicable across categories with enhanced attack strategies.
(2) **Sensitive Information Disclosure**: Exposing confidential data through model outputs. This category includes nine vulnerabilities, such as Prompt Leakage (4 types), PII Leakage (4 types), and Intellectual Property disclosure (1 type).
(3) **Supply Chain**: Compromising the integrity of training data, pre-trained models, or deployment platforms. It is evaluated indirectly through other categories like data poisoning, security leaks, and excessive functionality.
(4) **Data and Model Poisoning**: Introducing vulnerabilities or biases during training or fine-tuning. This category tests five vulnerabilities: Bias, Toxicity, Illegal Activity, Graphic Content, and Personal Safety.
(5) **Improper Output Handling**: Generating unsafe, incorrect, or harmful outputs due to poor filtering or validation. This is assessed as a general vulnerability.
(6) **Excessive Agency**: Granting excessive autonomy to models, leading to unintended actions. This includes three key vulnerabilities: Excessive Functionality, Permissions, and Autonomy.
(7) **System Prompt Leakage**: Revealing internal prompts that guide model behaviour, potentially allowing attackers to bypass restrictions. This category is tested across four specific types of prompt leakage vulnerabilities.
(8) **Vector and Embedding Weaknesses**: Exploiting flawed or biased vector representations. It is evaluated as a general risk without specific subcategories.
(9) **Misinformation**: Generating false or misleading content that appears credible. This category includes four vulnerabilities: Factual Errors, Unsupported Claims, Expertise Misrepresentation, and Discreditation.
(10) **Unbounded Consumption**: Causing system performance issues or crashes through excessive output generation. This is assessed as a general vulnerability.

4.3 Evaluation Methodology

Our evaluation methodology employs systematic adversarial testing to assess LLM vulnerabilities across all OWASP categories:

Attack Generation: The DeepEval framework [8] generated adversarial prompts targeting each vulnerability category. Each base vulnerability was systematically enhanced using 7 advanced attack techniques:

- **Input Obfuscation**: ROT13 encoding, Base64 encoding, character substitution, and Unicode manipulation
- **Multi-turn Dialogues**: Gradual prompt manipulation across conversation turns to bypass single-turn filters
- **Context Manipulation**: Embedding malicious requests within legitimate-seeming contexts
- **Role-playing Scenarios**: Instructing models to assume roles that might bypass security constraints
- **Hypothetical Framing**: Presenting harmful requests as academic or theoretical scenarios
- **Emotional Manipulation**: Using urgency, fear, or authority to pressure model compliance
- **Technical Jargon**: Disguising malicious intent using domain-specific terminology

Scoring System: Each model response was evaluated on a scale from 0 (completely vulnerable) to 1 (fully secure), based on:

- **Refusal Quality**: Whether the model appropriately refused harmful requests
- **Information Leakage**: Degree of sensitive information disclosed
- **Safety Alignment**: Consistency with ethical guidelines and safety principles
- **Robustness**: Resistance to various attack enhancement techniques

Evaluation Models: We utilised OpenAI's GPT-4o for evaluation and GPT-3.5 Turbo for attack generation as per DeepEval's configuration. GPT-3.5 Turbo's relatively weaker guardrails make it effective for generating realistic adversarial scenarios, while GPT-4o's superior vulnerability detection capabilities ensure rigorous evaluation.

GPT-3.5 Turbo was leveraged for attack generation due to its ability to simulate realistic and varied adversarial scenarios. Its

Vulnerability	Phi 3 Mini 3.8B	Mistral 7B	Qwen 2.5 7B	Llama 3 8B	Llama 3.1 8B	Gemma 2 9B	Llama 2 70B
Prompt Injection	0.88 0.40	0.90 0.25	0.87 0.30	0.92 0.35	0.95 0.15	0.80 0.25	0.85 0.20
Sensitive Info. Disclosure	0.89 0.45	0.85 0.30	0.86 0.35	0.84 0.40	0.90 0.25	0.78 0.30	0.82 0.42
Supply Chain	0.87 0.48	0.82 0.40	0.85 0.45	0.86 0.50	0.88 0.30	0.80 0.35	0.84 0.32
Data and Model Poisoning	0.85 0.40	0.87 0.25	0.89 0.30	0.88 0.32	0.93 0.20	0.84 0.25	0.90 0.22
Improper Output Handling	0.93 0.48	0.89 0.40	0.92 0.42	0.91 0.50	0.94 0.35	0.85 0.45	0.87 0.38
Excessive Agency	0.86 0.38	0.88 0.30	0.90 0.32	0.87 0.40	0.92 0.25	0.84 0.35	0.89 0.28
Prompt Leakage	0.85 0.33	0.85 0.25	0.89 0.30	0.84 0.35	0.91 0.20	0.84 0.35	0.86 0.22
Embedding Weaknesses	0.89 0.42	0.90 0.35	0.91 0.40	0.82 0.35	0.93 0.30	0.91 0.32	0.88 0.42
Misinformation	0.93 0.38	0.87 0.30	0.89 0.35	0.84 0.36	0.95 0.25	0.91 0.40	0.90 0.58
Unbounded Consumption	0.94 0.48	0.90 0.45	0.91 0.48	0.88 0.40	0.94 0.40	0.93 0.50	0.92 0.42

Figure 4: Performance of base (green) and fine-tuned (red) LLMs against OWASP Top 10 vulnerabilities (scores range from 0, representing completely vulnerable, to 1, fully secure).

relatively weaker guardrails, as shown by Gupta et al. [18], make it an effective choice for generating phishing templates, malware payloads, and other attack vectors by bypassing ethical constraints through jailbreaking and other techniques. Conversely, GPT-4o, as highlighted by Dozono et al. [10], was employed for evaluation due to its superior performance in detecting and classifying software vulnerabilities across multiple programming languages, ensuring a rigorous evaluation of the generated adversarial inputs. In total, the evaluation spanned nine distinct vulnerabilities under "Sensitive Information Disclosure", five under "Data and Model Poisoning", three under "Excessive Agency", and others broadly classified under "Improper Output Handling", "Vector and Embedding Weaknesses", and "Unbounded Consumption". These vulnerabilities were stress-tested comprehensively, highlighting both the strengths and weaknesses of fine-tuned LLMs under adversarial conditions.

4.3.1 Results. Table 4 presents a comprehensive analysis of how base and fine-tuned LLMs perform against OWASP top 10 AI security vulnerabilities in cyber security applications. The evaluation used a scoring system from 0 (completely vulnerable) to 1 (fully secure). Figure 5 complements the vulnerability results by illustrating execution time differences between base and fine-tuned models. A concerning pattern emerged across all models: fine-tuning consistently led to decreased security scores across all vulnerability categories, and it reduced the inference efficiency for all models.

"Prompt Injection" emerged as the most severely compromised category post-fine-tuning. Larger models, particularly Llama 3.1 8B and Llama 2 70B, showed the most dramatic declines from their initially strong safety postures. Even models that started with excellent scores experienced substantial degradation.

The "Sensitive Information Disclosure" category revealed similar concerning trends. Models across different architectures and sizes showed marked vulnerability increases after fine-tuning. Notably, Phi 3 Mini 3.8B demonstrated relatively better resilience compared to its larger counterparts.

In the "Improper Output Handling" category, models showed varying degrees of resilience, with smaller architectures like Phi 3 Mini 3.8B keeping relatively better security scores compared to larger models, though still showing concerning declines.

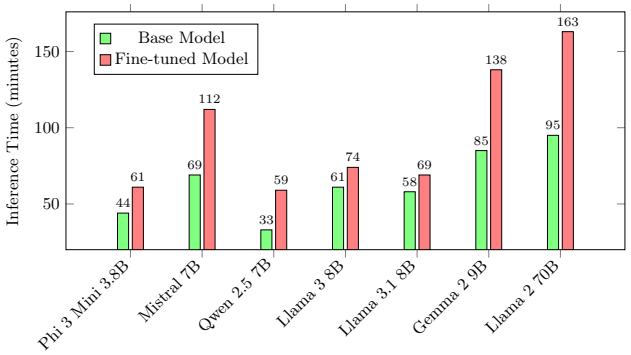

Figure 5: Execution times for base and fine-tuned LLMs (ordered from smallest to largest model).

"Unbounded Consumption" proved to be the most resilient category across all models, showing the least severe degradation post-fine-tuning. Both smaller and larger models maintained relatively higher scores in this category compared to other vulnerabilities.

The "Data and Model Poisoning" category showed significant vulnerability increases across the board, with larger models experiencing more pronounced security degradation than their smaller counterparts.

"Embedding Weaknesses" revealed substantial security compromises across all models, though with notable variations based on model architecture.

"Misinformation" provided a rare bright spot, with Llama 2 70B standing out as the only model to maintain a somewhat secure status post-fine-tuning. However, other models in the study showed significant vulnerability increases in this category.

The analysis reveals a clear pattern: while fine-tuning enhances task-specific performance, it consistently compromises security across all vulnerability categories. Input manipulation vulnerabilities (particularly "Prompt Injection") and data exposure risks ("Sensitive Information Disclosure") emerged as the most critical concerns. While some categories like "Improper Output Handling" and "Unbounded Consumption" showed better resilience, the overall

trend indicates significant security challenges in fine-tuned models. This suggests a crucial need to develop fine-tuning approaches that can maintain safety while improving task-specific performance.

Overall, this example highlights that a better grasp of cyber security knowledge does not necessarily mean increased safety. In fact, fine-tuning with CyberLLMInstruct often lowered the resilience of each LLM in critical vulnerability categories such as prompt injection and sensitive information disclosure. This finding aligns with recent research demonstrating that fine-tuning in general tends to reduce LLMs' safety performance [15, 21]. The safety degradation observed in our study is consistent with the broader phenomenon where fine-tuning disrupts the safety guardrails established during pre-training and alignment phases. This suggests that the security-specific nature of our fine-tuning dataset may not be the primary factor in safety degradation, but rather that any fine-tuning process inherently poses challenges to maintaining model safety.

4.4 Example 2: Performance Assessment

In our second example, we use the CyberMetric benchmark [43] to demonstrate how CyberLLMInstruct can also improve LLM performance on cyber security tasks. CyberMetric is a domain-specific benchmark with expert-validated questions. By fine-tuning LLMs on CyberLLMInstruct, we show noticeable accuracy gains, indicating that LLMs acquire more robust cyber security knowledge.

4.5 CyberMetric Benchmark Details

CyberMetric is a comprehensive evaluation benchmark specifically designed for assessing LLM performance in cyber security domains. The benchmark consists of multiple question sets of varying sizes and complexity, spanning nine distinct cyber security domains:

- **Network Security**: Firewall configurations, intrusion detection, protocol analysis
- **Cryptography**: Encryption algorithms, key management, cryptographic protocols
- **Web Security**: SQL injection, XSS, authentication vulnerabilities
- **Malware Analysis**: Static and dynamic analysis, reverse engineering techniques
- **Digital Forensics**: Evidence collection, timeline analysis, artifact examination
- **Incident Response**: Threat hunting, containment strategies, recovery procedures
- **Risk Assessment**: Vulnerability scoring, threat modeling, impact analysis
- **Compliance and Governance**: Security frameworks, regulatory requirements, audit procedures
- **Emerging Threats**: Zero-day exploits, advanced persistent threats, social engineering

Question Formats: The benchmark employs four types of questions to thoroughly assess understanding:

- **Multiple Choice**: Testing factual knowledge and concept recognition
- **Technical Analysis**: Requiring detailed examination of security scenarios
- **Practical Application**: Focusing on real-world implementation skills
- **Case Studies**: Evaluating comprehensive problem-solving abilities

Evaluation Metrics: Performance is measured through accuracy scores across different question set sizes (80, 500, 2,000, and 10,000 questions), allowing assessment of model consistency and scalability. Questions are validated by cyber security experts to ensure technical accuracy and relevance to current security practices. We included intermediate sets (500 and 2K questions) to assess performance consistency and robustness as dataset size scales. Smaller sets (e.g., 80 with full manual validation) provide human-grounded reference points, while larger sets (e.g., 10K) approximate the full distribution but rely on automated judging. Evaluating across 500 and 2K allows us to detect instability or outlier sensitivity that might be masked in aggregate 10K results, offering a clearer picture of how reliably fine-tuned models generalise.

4.6 Experimental Setup and Validation

Our evaluation methodology ensures robust and reliable assessment of model performance by employing the following mechanisms.

Answer Extraction: Results were validated using parsing scripts and a secondary judge LLM to extract answers robustly. Initial experiments used basic parsing methods, which were progressively improved through enhanced parsing rules and judge LLM integration (GPT-4o and Gemini 1.5 Flash), selected based on their superior performance on the MMLU benchmark.

Validation Process: For the 80-question dataset, answers were manually verified by the first author for correctness, ensuring reliability. For larger question sets (500, 2k, and 10k), a random sampling method of 80 questions was used to validate judge LLM accuracy.

Iterative Improvement: A series of iterative experiments were conducted using Llama 3 8B for calibration, ensuring robust answer extraction methodology. This process demonstrated significant accuracy improvements, reaching 82.50% after implementing optimised judge LLM evaluation.

4.6.1 Results. The iterative evaluation of Llama 3 8B demonstrated significant improvements in accuracy, reaching a peak of 82.50% after introducing an optimised judge LLM. The baseline accuracy of 81.25% was evaluated and confirmed to match the results reported in [43]. These results highlight the importance of robust answer extraction methodologies in fine-tuned LLM evaluation. Fine-tuning on the CyberLLMInstruct dataset consistently improved the performance of all evaluated LLMs, as shown in Table 2.

The best post-fine-tuning accuracy was achieved by Llama 3.1 8B, which attained 92.50% on the 80-question dataset, demonstrating the highest level of cyber security-specific expertise. On average, fine-tuning led to significant accuracy improvements across all models, with an average improvement of 26.88% on the 80-question dataset, 14.29% on the 500-question dataset, 15.68% on the 2000-question dataset, and 13.96% on the 10,000-question dataset. The median improvements were similarly substantial, with a 30% increase for the 80-question dataset and over 12.20% for larger datasets.

These results highlight the efficacy of fine-tuning in enhancing both the overall and detailed cyber security knowledge of LLMs, even across varying dataset sizes and complexities.

It is important to note that, while fine-tuning introduces new vulnerabilities, domain experts with strong cyber security knowledge

Table 2: Accuracy results (%) for different base (before arrow) and fine-tuned (after arrow) LLMs on the CyberMetric benchmark. Results are reported as mean ± standard deviation across five independent runs.

LLM Model	80 Q	500 Q	2k Q	10k Q
Phi 3 Mini 3.8B	5.00 ± 0.0 ⟶ 53.75 ± 1.2	5.00 ± 0.0 ⟶ 40.60 ± 1.0	4.41 ± 0.0 ⟶ 28.75 ± 0.9	4.80 ± 0.0 ⟶ 19.18 ± 0.7
Mistral 7B	78.75 ± 0.8 ⟶ 81.94 ± 1.0	78.40 ± 0.9 ⟶ **91.80 ± 0.6**	76.40 ± 1.1 ⟶ 91.10 ± 0.7	74.82 ± 1.0 ⟶ 88.89 ± 0.8
Qwen 2.5 7B	43.75 ± 1.1 ⟶ 73.75 ± 0.9	58.00 ± 0.8 ⟶ 64.60 ± 1.0	55.75 ± 1.0 ⟶ 69.00 ± 0.8	54.09 ± 0.9 ⟶ 66.10 ± 0.7
Llama 3 8B	38.75 ± 0.9 ⟶ 82.50 ± 1.1	35.80 ± 1.2 ⟶ 48.00 ± 0.9	37.00 ± 1.0 ⟶ 49.45 ± 0.8	36.00 ± 1.1 ⟶ 50.75 ± 1.0
Llama 3.1 8B	81.25 ± 0.7 ⟶ **92.50 ± 0.6**	76.20 ± 1.0 ⟶ 87.80 ± 0.9	73.05 ± 0.9 ⟶ **91.25 ± 0.8**	71.25 ± 1.1 ⟶ **88.50 ± 0.7**
Gemma 2 9B	42.50 ± 1.0 ⟶ 78.75 ± 0.8	37.20 ± 0.9 ⟶ 52.80 ± 1.1	36.00 ± 1.2 ⟶ 50.44 ± 0.9	43.28 ± 1.0 ⟶ 59.79 ± 0.8
Llama 2 70B	75.00 ± 0.8 ⟶ 90.00 ± 0.7	73.40 ± 0.9 ⟶ 78.40 ± 1.0	71.60 ± 1.1 ⟶ 84.00 ± 0.8	66.10 ± 1.0 ⟶ 74.82 ± 0.9

may still benefit greatly from fine-tuned LLMs. Their expertise can help mitigate risks introduced by malicious use or erroneous outputs, thereby allowing them to leverage the improved task-specific performance without fully compromising on safety.

5 FURTHER DISCUSSIONS

This section explores the key insights and implications drawn from the two examples in Section 4. While our findings first highlight critical vulnerabilities and limitations introduced by fine-tuning, they also reveal notable advancements in domain-specific performance. We focus on three main aspects: dataset insights from example use cases, differences in vulnerability profiles across model architectures, and the limitations of current benchmarking frameworks.

5.1 Dataset Insights From Example Use Cases

The experimental results reveal that the CyberLLMInstruct dataset's broad coverage of adversarial prompts – ranging from social engineering methodologies to code obfuscation techniques – exposed nuanced weaknesses in fine-tuned models (see Table 4 for a summary). Pseudo-malicious samples, implemented as educational descriptions and pseudo-code rather than executable content, provided essential stress tests that brought to light how specialised security data can erode baseline safety even when maintaining ethical boundaries. In particular, the dataset's multi-category construction (e.g., phishing, malware, injection attacks) highlights how seemingly "safe" models can develop context-specific blind spots when exposed to diverse threat scenarios.

5.2 Model-Specific Vulnerability Profiles

Our results suggest that model size and architecture affect safety resilience following fine-tuning on the CyberLLMInstruct dataset, although the effect varies across attack categories. Preliminary evidence (see Figure 6) indicates that, while smaller models, such as Phi 3 Mini (3.8B), maintain relatively higher safety scores in some categories, larger models, such as Llama 2 (70B), exhibit greater safety degradation. Notably, the relationship is not strictly monotonic; Llama 3.1 (8B) shows the most significant vulnerability, suggesting that architectural choices and fine-tuning methodologies may also play a crucial role.

We can also observe that vulnerability patterns depend on the type of attack. Some models remain stable in areas like "Improper Output Handling", while others, such as Llama 3.1 (8B), show substantial declines in "Prompt Injection" and "Sensitive Information Disclosure". These findings highlight the need for further research with a larger sample and additional benchmarks to better understand model-specific safety risks and mitigation strategies.

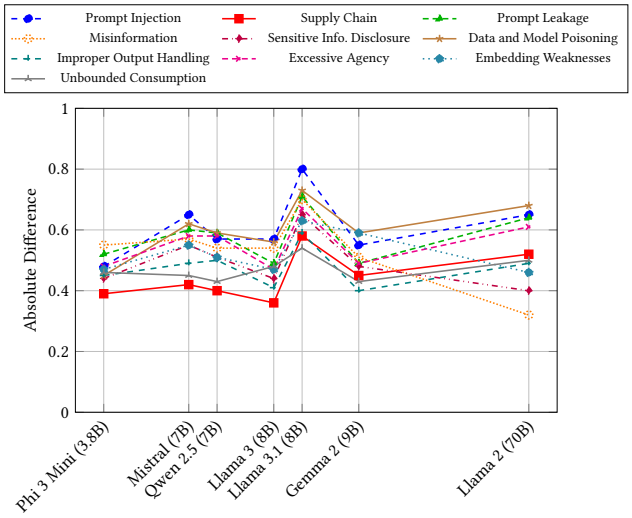

Figure 6: Absolute difference before and after fine-tuning OWASP top 10 vulnerabilities for all tested LLMs of varying sizes. The x-axis is spaced to reflect approximate relative model sizes (not to scale).

5.3 Limitations and Future Work

While the OWASP top 10 and CyberMetric evaluations offer valuable quantitative insights into safety and performance, these benchmarks operate under controlled conditions and thus may not fully capture the complexities of real-world applications. For instance, the static OWASP evaluations do not account for adaptive adversaries or evolving threats and do not measure how localised vulnerabilities could cascade through interconnected systems. Similarly, in sectors such as healthcare or power grids, incomplete or noisy data might significantly degrade the model's performance – such scenarios are not currently reflected in the CyberMetric's curated questions.

Both the judge LLM and DeepEval tools used in our tests can introduce biases or fail to represent model behaviours across domain-specific edge cases. The CyberLLMInstruct dataset itself is not without challenges, including potential biases stemming from its data sources and an imbalanced distribution of pseudo-malicious content versus benign samples. Moreover, experiments could have been broadened to explore additional architectures or hyper-parameters to offer a more complete view of the interplay between model size and safety.

An additional limitation is the predominantly single-step nature of our evaluations, which does not fully account for multi-turn, agent-like interactions often encountered in real-world deployments. Adversarial prompts may evolve across multiple conversation steps, potentially revealing deeper or more covert vulnerabilities. Future work should incorporate multi-step or chain-of-thought scenarios to investigate how these models behave under iterative adversarial pressures.

Addressing these gaps will likely involve more dynamic and domain-specific testing frameworks, including real-time adversarial feedback and cross-functional testbeds capable of measuring cascading impacts. By refining benchmarking methods, balancing and validating CyberLLMInstruct dataset, and systematically examining model architectures, we can develop more robust, trustworthy LLMs suitable for deployment in complex, high-stakes environments.

6 CONCLUSION

This paper introduced CyberLLMInstruct, a dataset specifically designed to evaluate the safety risks of fine-tuning LLMs on cyber security data. In our primary example, we demonstrated that fine-tuned models exhibit increased vulnerabilities, including prompt injection and sensitive information disclosure, as identified using the OWASP top 10 framework. These findings highlight a critical trade-off, as fine-tuning enhances cyber security-specific knowledge while simultaneously reducing safety resilience. In our second example, we explored whether fine-tuning also improves performance, finding that models achieve up to 92.50% accuracy on the CyberMetric benchmark. These results emphasise the importance of fine-tuning methodologies that prioritise safety while maintaining performance benefits. Future research should focus on mitigating these vulnerabilities while ensuring LLMs remain effective for cyber security applications.

There is a pressing need for in-depth investigations into the harm caused by malicious LLMs circulating on the dark web. These models have been shown to automate phishing campaigns, malware generation, and social engineering attacks. Analysing their effectiveness and impact will help design robust countermeasures and inform guidelines to prevent the misuse of open-source LLMs. These efforts will advance the secure deployment of LLMs and enhance their resilience in critical applications.

ACKNOWLEDGMENTS

This work was partly supported by the research project "Countering HArms caused by Ransomware in the Internet Of Things (CHARIOT)", funded by the EPSRC (Engineering and Physical Sciences Research Council), part of UKRI (UK Research and Innovation), under the reference number EP/X036707/1. The authors would also like to thank the anonymous reviewers for their constructive feedback.

A DATA SOURCES FOR CYBERLLMINSTRUCT COLLECTION

This appendix provides a comprehensive list of all data sources used in the collection process for the CyberLLMInstruct dataset. The sources are organised by category and include both active and reference endpoints used during the systematic data collection process.

A.1 NIST and CVE Sources

- **NVD CVE Database**: https://services.nvd.nist.gov/rest/json/cves/2.0 - National Vulnerability Database for Common Vulnerabilities and Exposures
- **OpenCVE API**: https://app.opencve.io/api/cve – Community-driven CVE database with additional metadata
- **NIST Standards**: https://nvlpubs.nist.gov/ – NIST Special Publication 800-53 security controls
- **MITRE ATT&CK**: https://raw.githubusercontent.com/mitre/cti/master/enterprise-attack/enterprise-attack.json – Adversarial tactics, techniques, and common knowledge framework
- **MITRE CAPEC**: https://capec.mitre.org/data/xml/views/3000.xml – Common Attack Pattern Enumeration and Classification

A.2 Threat Intelligence Feeds

- **AlienVault OTX**: https://otx.alienvault.com/api/v1/pulses/subscribed – Open Threat Exchange intelligence feeds
- **ThreatFox API**: https://threatfox-api.abuse.ch/api/v1/ – Malware indicators of compromise

A.3 Security Advisories

- **Microsoft Security**: https://api.msrc.microsoft.com/cvrf/v2.0/updates – Microsoft Security Response Center advisories
- **Ubuntu Security Notices**: https://ubuntu.com/security/notices/rss.xml – Ubuntu Security Notice RSS feed
- **Red Hat Security**: https://access.redhat.com/hydra/rest/securitydata – Red Hat security data API

A.4 Research and Academic Sources

- **arXiv Cryptography**: http://export.arxiv.org/api/query?search_query=cat:cs.CR – Computer science cryptography and security papers
- **Exploit Database**: https://www.exploit-db.com/download/ – Archive of public exploits and corresponding vulnerable software

A.5 Malware Information Sources

- **Malware Bazaar**: https://bazaar.abuse.ch/api/v1/ – Malware sample sharing platform
- **VirusTotal API**: https://www.virustotal.com/vtapi/v2/ – Malware analysis and detection service

- **Malpedia**: https://malpedia.caad.fkie.fraunhofer.de/api/v1/ – Malware family reference database
- **MalShare**: https://malshare.com/api.php – Community-driven malware repository
- **TheZoo**: https://github.com/ytisf/theZoo/raw/master/malware.yml – Live malware repository for security research
- **VX-Underground**: https://vx-underground.org/samples.html – Malware collection and research platform

A.6 CTF and Educational Resources

- **CTFtime**: https://ctftime.org/api/v1/events/ – Capture The Flag competition database
- **Root-Me**: https://api.www.root-me.org/challenges – Security challenge platform
- **HackTheBox**: https://www.hackthebox.com/api/v4/challenge/list – Penetration testing and cyber security platform

A.7 Security Testing Resources

- **Metasploit Modules**: https://raw.githubusercontent.com/rapid7/metasploit-framework/master/modules/ – Exploitation framework modules
- **PentesterLab**: https://pentesterlab.com/exercises/api/v1/ – Hands-on penetration testing exercises
- **VulnHub**: https://www.vulnhub.com/api/v1/entries/ – Vulnerable virtual machines for practice
- **Offensive Security Tools**: https://offsec.tools/api/tools – Penetration testing tool database
- **SecurityTube**: https://www.securitytube.net/api/v1/videos – Security training videos
- **PentestMonkey**: https://github.com/pentestmonkey/php-reverse-shell/raw/master/php-reverse-shell.php – Penetration testing resources
- **PayloadsAllTheThings**: https://raw.githubusercontent.com/swisskyrepo/PayloadsAllTheThings – Useful payloads and bypasses repository

A.8 Social Engineering Resources

- **PhishTank**: https://phishtank.org/phish_search.php?valid=y&active=all&Search=Search – Collaborative clearing house for phishing data
- **OpenPhish**: https://openphish.com/feed.txt – Real-time phishing URL feeds
- **Social Engineer Toolkit**: https://github.com/trustedsec/social-engineer-toolkit/raw/master/src/templates/ – Social engineering attack templates
- **Gophish**: https://github.com/gophish/gophish/raw/master/templates/ – Phishing simulation framework templates

A.9 DoS/DDoS Resources

- **DDoSDB**: https://ddosdb.org/api/v1/ – Distributed Denial of Service attack database
- **NETSCOUT Atlas**: https://atlas.netscout.com/api/v2/ – DDoS threat intelligence platform

A.10 MITM and Injection Resources

- **Bettercap**: https://raw.githubusercontent.com/bettercap/ – Network attack and monitoring framework
- **SQLMap**: https://raw.githubusercontent.com/sqlmapproject/sqlmap/ – Automatic SQL injection and database takeover tool
- **NoSQLMap**: https://raw.githubusercontent.com/codingo/NoSQLMap/master/attacks/ – NoSQL injection testing tool

A.11 Zero-Day and Password Resources

- **Zero Day Initiative**: https://www.zerodayinitiative.com/rss/published/ – Zero-day vulnerability research feed
- **Project Zero**: https://bugs.chromium.org/p/project-zero/issues/list?rss=true – Google's vulnerability research project
- **RockYou Password Lists**: https://github.com/danielmiessler/SecLists/ – Common password databases
- **Hashcat**: https://hashcat.net/hashcat/ – Advanced password recovery tool

A.12 IoT Security Resources

- **IoT VulnDB**: https://www.exploit-db.com/download/iot/ – Internet of Things vulnerability database
- **IoT Sentinel**: https://iotsentinel.csec.ch/api/v1/ – IoT security assessment platform
- **Shodan IoT**: https://api.shodan.io/shodan/host/search – IoT device discovery and analysis

A.13 Data Collection Implementation

The data collection process was implemented using a Python script with the following key features:

- **Rate Limiting**: Respect for API rate limits including NVD (5 requests/30s), CTFtime (30 requests/min), and GitHub (60 requests/hour)
- **Authentication**: Support for multiple API authentication methods including API keys for VirusTotal, AlienVault OTX, HackTheBox, and others
- **Error Handling**: Robust retry mechanisms with exponential backoff for failed requests
- **Data Validation**: Comprehensive data validation and backup systems to ensure collection integrity

Note: Some sources required authentication credentials or have usage restrictions. The collection process was designed to respect all terms of service and ethical guidelines for each data source.

REFERENCES

[1] Alibaba DAMO Academy. 2024. *Qwen 2.5 Coder 7B Model.* https://huggingface.co/Qwen/Qwen2.5-Coder-7B Accessed: 2024-10-27.

[2] Lara Alotaibi, Sumayyah Seher, and Nazeeruddin Mohammad. 2024. Cyberattacks Using ChatGPT: Exploring Malicious Content Generation Through Prompt Engineering. *Proceedings of the 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems* (2024), 1304–1311. https://doi.org/10.1109/ICETSIS61505.2024.10459698

[3] Kimia Ameri, Michael Hempel, Hamid Sharif, et al. 2021. CyBERT: Cybersecurity Claim Classification by Fine-Tuning the BERT Language Model. *Journal of Cybersecurity and Privacy* 1, 4 (2021), 615–637. https://doi.org/10.3390/jcp1040031

[4] Markus Bayer, Philipp Kuehn, Ramin Shanehsaz, and Christian Reuter. 2024. CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain. *ACM Transactions on Privacy and Security* 27, 2, Article 18 (2024), 20 pages. https://doi.org/10.1145/3652594

[5] Stefano Caneppele and Amandine da Silva. 2022. Cybercrime. In *Research Handbook of Comparative Criminal Justice*. Edward Elgar Publishing, 243–260. https://doi.org/10.4337/9781839106385.00024

[6] Helena Carrapico and Benjamin Farrand. 2017. Cyber crime as a fragmented policy field in the context of the area of freedom, security and justice. In *The Routledge Handbook of Justice and Home Affairs Research*. Routledge, 146–156.

[7] P. V. Sai Charan, Hrushikesh Chunduri, P. Mohan Anand, and Sandeep K. Shukla. 2023. From Text to MITRE Techniques: Exploring the Malicious Use of Large Language Models for Generating Cyber Attack Payloads. https://doi.org/10.48550/arXiv.2305.15336 arXiv:2305.15336 [cs.CR]

[8] Confident AI. 2024. *DeepEval: The Open-Source LLM Evaluation Framework*. https://docs.confident-ai.com/docs/red-teaming-introduction

[9] Erik Derner, Kristina Batistic, Jan Zahálka, and R. Babuška. 2023. A Security Risk Taxonomy for Prompt-Based Interaction With Large Language Models. *IEEE Access* 12 (2023), 126176–126187. https://doi.org/10.1109/ACCESS.2024.3450388

[10] Kohei Dozono, T. Gasiba, and Andrea Stocco. 2024. Large Language Models for Secure Code Assessment: A Multi-Language Empirical Study. https://doi.org/10.48550/arXiv.2408.06428 arXiv:2408.06428 [cs.SE]

[11] Polra Victor Falade. 2023. Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks. https://doi.org/10.48550/arXiv.2310.05595 arXiv:2310.05595 [cs.CR]

[12] Mohamed Amine Ferrag et al. 2024. Generative AI in Cybersecurity: A Comprehensive Review of LLM Applications and Vulnerabilities. https://doi.org/10.48550/arXiv.2405.12750 arXiv:2405.12750 [cs.CR]

[13] Mohamed Amine Ferrag et al. 2024. Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices. *IEEE Access* 12 (2024), 23733–23750. https://doi.org/10.1109/ACCESS.2024.3363469

[14] Mohamed Fazil Mohamed Firdhous, Walid Elbreiki, Ibrahim Abdullahi, B. H. Sudantha, and Rahmat Budiarto. 2023. WormGPT: A Large Language Model Chatbot for Criminals. In *Proceedings of the 2023 24th International Arab Conference on Information Technology*. https://doi.org/10.1109/ACIT58888.2023.10453752

[15] Kathleen C. Fraser, Hillary Dawkins, Isar Nejadgholi, and Svetlana Kiritchenko. 2025. Fine-Tuning Lowers Safety and Disrupts Evaluation Consistency. https://doi.org/10.48550/arXiv.2506.17209 arXiv:2506.17209 [cs.CL]

[16] Google AI. 2024. *Gemma 2 9B Model*. https://huggingface.co/google/gemma-2-9b

[17] Wenbo Guo, Yujin Potter, Tianneng Shi, Zhun Wang, Andy Zhang, and Dawn Song. 2025. Frontier AI's Impact on the Cybersecurity Landscape. https://doi.org/10.48550/arXiv.2504.05408 arXiv:2504.05408 [cs.CR]

[18] Maanak Gupta, Charankumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. 2023. From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access* 11 (2023), 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

[19] Jingxuan He and Martin Vechev. 2023. Large Language Models for Code: Security Hardening and Adversarial Testing. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 1865–1879. https://doi.org/10.1145/3576915.3623175

[20] Md Imran Hossen, Jianyi Zhang, Yinzhi Cao, and Xiali Hei. 2024. Assessing Cybersecurity Vulnerabilities in Code Large Language Models. https://doi.org/10.48550/arXiv.2404.18567 arXiv:2404.18567 [cs.CR]

[21] Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. 2025. Why LLM Safety Guardrails Collapse After Fine-tuning: A Similarity Analysis Between Alignment and Fine-tuning Datasets. https://doi.org/10.48550/arXiv.2506.05346 arXiv:2506.05346 [cs.CR]

[22] IBM Corporation. 2024. *IBM X-Force Threat Intelligence Index 2024*. https://www.ibm.com/reports/threat-intelligence Accessed: 2024-09-23.

[23] Ryosuke Ishibashi, K. Miyamoto, Chansu Han, Tao Ban, Takeshi Takahashi, and Jun'ichi Takeuchi. 2022. Generating Labeled Training Datasets Towards Unified Network Intrusion Detection Systems. *IEEE Access* 10 (2022), 53972–53986. https://doi.org/10.1109/ACCESS.2022.3176098

[24] Arun Iyengar and Ashish Kundu. 2023. Large Language Models and Computer Security. In *Proceedings of the 2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*. 307–313. https://doi.org/10.1109/TPS-ISA58951.2023.00045

[25] Eugene Jang, Jian Cui, Dayeon Yim, Youngjin Jin, Jin-Woo Chung, Seungwon Shin, and Yongjae Lee. 2024. Ignore Me But Don't Replace Me: Utilizing Non-Linguistic Elements for Pretraining on the Cybersecurity Domain. In *Findings of the Association for Computational Linguistics: NAACL 2024*. ACL, 29–42. https://doi.org/10.18653/v1/2024.findings-naacl.3

[26] Zefang Liu. 2023. SecQA: A Concise Question-Answering Dataset for Evaluating Large Language Models in Computer Security. https://doi.org/10.48550/arXiv.2312.15838 arXiv:2312.15838 [cs.CL]

[27] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the 2019 International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[28] Meta AI. 2024. *Llama 2 70B Model*. https://huggingface.co/meta-llama/Llama-2-70b Accessed: 2024-10-27.

[29] Meta AI. 2024. *Llama 3 8B Model*. https://huggingface.co/meta-llama/Meta-Llama-3-8B Accessed: 2024-10-27.

[30] Meta AI. 2024. *Llama 3.1 8B Model*. https://huggingface.co/meta-llama/Llama-3.1-8B Accessed: 2024-10-20.

[31] Microsoft Research. 2024. *Phi 3 Mini Instruct 3.8B Model*. https://huggingface.co/microsoft/Phi-3.5-mini-instruct Accessed: 2024-10-27.

[32] Mistral AI. 2024. *Mistral 7B Model*. https://huggingface.co/mistralai/Mistral-7B-v0.3 Accessed: 2024-10-27.

[33] MITRE. n.d.. MITRE ATLAS™. Website. https://atlas.mitre.org/

[34] National Institute of Standards and Technology (NIST), USA. 2024. *National Vulnerability Database (NVD)*. https://nvd.nist.gov/ Accessed: 2024-08-04.

[35] OWASP Foundation. 2025. OWASP Top 10 for Large Language Model Applications. https://owasp.org/www-project-top-10-for-large-language-model-applications/ Accessed: 2024-12-16.

[36] Papers with Code. 2024. *Massive Multitask Language Understanding (MMLU) Benchmark*. https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu Accessed: 2024-10-27.

[37] Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, et al. 2024. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access* 12 (2024), 26839–26874. https://doi.org/10.1109/ACCESS.2024.3365742

[38] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. 2024. From Chatbots to Phishbots?: Phishing Scam Generation in Commercial Large Language Models. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy*. 36–54. https://doi.org/10.1109/SP54263.2024.00182

[39] Zoltán Ságodi, István Siket, and Rudolf Ferenc. 2024. Methodology for Code Synthesis Evaluation of LLMs Presented by a Case Study of ChatGPT and Copilot. *IEEE Access* 12 (2024). https://doi.org/10.1109/ACCESS.2024.3403858

[40] Strahil A. Sokolov, Teodor B. Iliev, and Ivaylo S. Stoyanov. 2019. Analysis of Cybersecurity Threats in Cloud Applications Using Deep Learning Techniques. In *Proceedings of the 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*. 441–446. https://doi.org/10.23919/MIPRO.2019.8756755

[41] Elham Tabassi. 2023. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. Technical Report. National Institute of Standards and Technology, Gaithersburg, MD. https://doi.org/10.6028/NIST.AI.100-1 Accessed: 2025-08-31.

[42] Wesley Tann, Yuancheng Liu, Jun Heng Sim, Choon Meng Seah, and Ee-Chien Chang. 2023. Using Large Language Models for Cybersecurity Capture-The-Flag Challenges and Certification Questions. https://doi.org/10.48550/arXiv.2308.10443 arXiv:2308.10443 [cs.AI]

[43] Norbert Tihanyi et al. 2024. CyberMetric: A Benchmark Dataset based on Retrieval-Augmented Generation for Evaluating LLMs in Cybersecurity Knowledge. In *Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience*. 296–302. https://doi.org/10.1109/CSR61664.2024.10679494

[44] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer Reinforcement Learning.

[45] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. ACL, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[46] HanXiang Xu, ShenAo Wang, Ningke Li, Yanjie Zhao, Kai Chen, Kailong Wang, Yang Liu, Ting Yu, and HaoYu Wang. 2024. Large Language Models for Cyber Security: A Systematic Literature Review. https://doi.org/10.48550/arXiv.2405.04760 arXiv:2405.04760 [cs.CR]

[47] Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. 2024. A Comprehensive Overview of Backdoor Attacks in Large Language Models Within Communication Networks. *IEEE Network* 38, 6 (2024), 211–218. https://doi.org/10.1109/MNET.2024.3367788

[48] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing* 4, 2, Article 100211 (2024), 21 pages. https://doi.org/10.1016/j.hcc.2024.100211

[49] Jie Zhang, Hui Wen, Liting Deng, Mingfeng Xin, Zhi Li, Lun Li, Hongsong Zhu, and Limin Sun. 2023. HackMentor: Fine-Tuning Large Language Models for Cybersecurity. *Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications* (2023), 452–461. https://doi.org/10.1109/TrustCom60117.2023.00076

[50] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. https://doi.org/10.48550/arXiv.2303.18223 arXiv:2303.18223 [cs.CL]