Primus: A Pioneering Collection of Open-Source Datasets for Cybersecurity LLM Training

Yao-Ching Yu[*], **Tsun-Han Chiang**[*†], **Cheng-Wei Tsai**[*†], **Chien-Ming Huang**[*†], **Wen-Kwang Tsao**
AI Lab, TrendMicro
`{yaoching_yu,james_chiang,dennis_tsai,liam_huang,spark_tsao}@trendmicro.com`

Abstract

Large Language Models (LLMs) have shown remarkable advancements in specialized fields such as finance, law, and medicine. However, in cybersecurity, we have noticed a lack of open-source datasets, with a particular lack of high-quality cybersecurity pretraining corpora, even though much research indicates that LLMs acquire their knowledge during pretraining. To address this, we present a comprehensive suite of datasets covering all major training stages, including pretraining, instruction fine-tuning, and reasoning distillation with cybersecurity-specific self-reflection data. Extensive ablation studies demonstrate their effectiveness on public cybersecurity benchmarks. In particular, continued pre-training on our dataset yields a ***15.9%*** improvement in the aggregate score, while reasoning distillation leads to a ***15.8%*** gain in security certification (CISSP). We will release all datasets and trained cybersecurity LLMs under the ODC-BY and MIT licenses to encourage further research in the community.[1]

1 Introduction

Large Language Models (LLMs) have significantly advanced artificial intelligence by leveraging massive data and sophisticated neural architectures, such as *ChatGPT* (Ouyang et al., 2022), *Llama* (Dubey et al., 2024) and *DeepSeek* (Guo et al., 2025). These models excel at understanding and generating human language (Wei et al., 2022; Minaee et al., 2024) and adapt well when collaborating with domain experts (Ge et al., 2023), enabling tailored applications in fields like medicine, law, and education (Lai et al., 2024; Zhou et al., 2023; Yan et al., 2024). Meanwhile, in cybersecurity, as cyber threats continue to evolve (Li and Liu, 2021; Ghelani, 2022), traditional methods such as signature- and rule-based systems are struggling to keep up. Advances in AI, particularly through LLMs, therefore offer promising new avenues for enhancing cybersecurity (Ferrag et al., 2024).

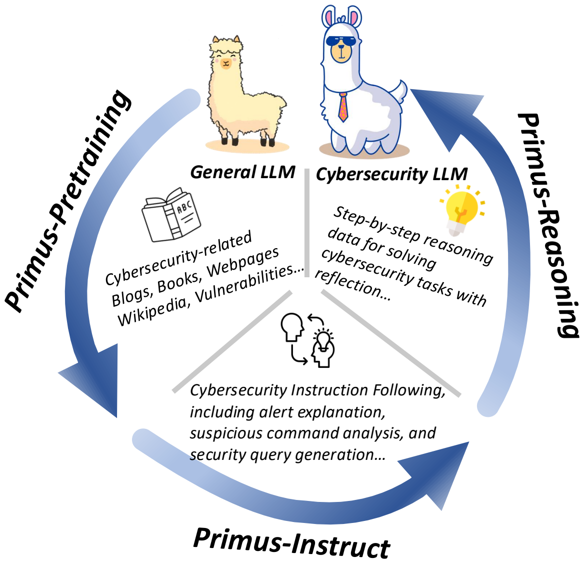

Primus-Pretraining
General LLM
Cybersecurity LLM
Primus-Reasoning
Cybersecurity-related
Blogs, Books, Webpages
Wikipedia, Vulnerabilities...
Step-by-step reasoning data for solving cybersecurity tasks with reflection...
Cybersecurity Instruction Following, including alert explanation, suspicious command analysis, and security query generation...
Primus-Instruct

Figure 1: Overview of our training pipeline. Primus-Pretraining, Primus-Instruct, and Primus-Reasoning are the datasets of different training stages.

Common training methods for LLMs include pre-training (PT) (Radford, 2018), supervised fine-tuning (SFT) (Zhang et al., 2023b), and reinforcement learning (RL) (Wang et al., 2024b). Recent studies suggest LLMs acquire knowledge primarily during PT, and continued pre-training (CPT) (Gururangan et al., 2020), which further trains pre-trained models on large amounts of domain-specific text, can enhance their grasp of domain knowledge. In contrast, SFT may introduce hallucinations as new knowledge is learned (Gekhman et al., 2024). More recently, collecting reflection data from reasoning models for distillation has also become a trend (Huang et al., 2024). Typically, obtaining a domain-specific LLM may require applying multiple training methods, as in our pipeline (Fig.1).

The cybersecurity field has yet to fully ben-

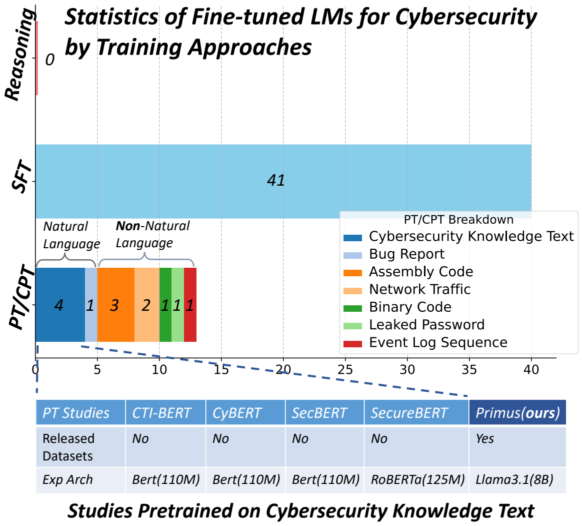
---

[*] Primary Contributor.

[†] Equal Contribution.

[1] **For access to all datasets and model weights, please refer to this link.**

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 10391–10413
November 4-9, 2025 ©2025 Association for Computational Linguistics



PT Studies	CTI-BERT	CyBERT	SecBERT	SecureBERT	Primus(**ours**)
Released Datasets	No	No	No	No	Yes
Exp Arch	Bert(110M)	Bert(110M)	Bert(110M)	RoBERTa(125M)	Llama3.1(8B)

Figure 2: Motivation behind Primus. Statistics of existing cybersecurity language models, where *reasoning* means training models to reason via distillation or RL.

efit from this transformative technology, which requires domain expertise due to its broad and complex nature. Our statistics on cybersecurity LLM survey papers (Zhang et al., 2024a; Xu et al., 2024a) indicate that most existing research focuses on SFT to align model outputs, while PT or CPT is largely performed on non-natural language data such as assembly code (Jiang et al., 2023; Wang et al., 2024a; Sun et al., 2023), as shown in Fig.2. Clearly, these approaches have limited effectiveness in improving the general cybersecurity knowledge of LLMs. On the other hand, models pre-trained on cybersecurity knowledge (Park and You, 2023; Ranade et al., 2021; Jackaduma, 2021; Aghaei et al., 2022) are limited to small ones like BERT (Devlin et al., 2019), and none of them have released datasets. To the best of our knowledge, LLMs pre-trained on cybersecurity knowledge or distilled on reasoning data from cybersecurity tasks remain *unexplored*.

To address this gap, we extend prior work on domain-specific LLMs like medicine (Labrak et al., 2024) and law (Colombo et al., 2024) to cybersecurity. Our contributions are as follows:

- ***A Collection of Cybersecurity Datasets.*** We create a series of carefully curated datasets covering multiple stages of LLM training, including pre-training (Primus-Pretraining), instruction fine-tuning (Primus-Instruct), and reasoning fine-tuning (Primus-Reasoning), as shown in Fig.1. Extensive ablation studies and evaluations on cybersecurity benchmarks show that these datasets can effectively improve cybersecurity capabilities. All datasets will be released under the ODC-BY license to encourage further research in the community.
- ***A Family of Cybersecurity LLMs.*** We present a family of cybersecurity LLMs designed to tackle domain-specific challenges, including *Llama-Primus-Base*, a model further pre-trained with cybersecurity knowledge based on *Llama-3.1-8B-Instruct*, achieving a ***15.9%*** improvement on aggregated cybersecurity benchmarks; *Llama-Primus-Merged*, an instruction-tuned variant merged with *Llama-3.1-8B-Instruct*, which **retains general instruction-following capability** while significantly improving cybersecurity performance; and *Llama-Primus-Reasoning*, which is distilled from reasoning steps with reflection generated by a larger reasoning LLM on cybersecurity tasks, providing it long-thought capabilities and yielding a ***15.8%*** gain on security certification. Likewise, all models will be released under the MIT license.

2 Training Datasets

2.1 Overview

We build our dataset in multiple stages. First, we collect high-quality cybersecurity texts from reputable sources to form Primus-Seed (Sec.2.2), which is valuable but covers only a small fraction of cybersecurity content on the web. To extend it, we train a cybersecurity text classifier using Primus-Seed as positive samples and sampled data from FineWeb (Penedo et al., 2024), a refined version of Common Crawl (Common Crawl, 2008), as negative samples. This classifier filters cybersecurity-related content from FineWeb, producing Primus-FineWeb (Sec.2.3). By combining both datasets, we derive Primus-Pretraining. Next, we introduce Primus-Instruct (Sec.2.4), which contains about 1k carefully curated cybersecurity tasks and general dialogues for instruction fine-tuning (IFT). Finally, Primus-Reasoning (Sec.2.5) provides reasoning steps generated by a stronger reasoning LLM on cybersecurity tasks for distillation.

2.2 Primus-Seed

2.2.1 Composition

We collect cybersecurity text through two main approaches. First, we gather data from reputable sources via official dumps or web crawling, converting raw HTML to readable Markdown using `dom-to-semantic-markdown`[2]. Second, we incor-

---

[2] https://github.com/romansky/dom-to-semantic-markdown

Category	**Samples**	**Tokens**	***Avg.***
Web Crawl / Official Dump			
Cybersecurity Blogs/News	2,946	9,751,002	3,309.9
Cybersecurity Books	6,499	2,910,464	447.8
Cybersecurity Companies Websites	76,919	65,798,561	855.4
Cybersecurity Wikipedia	6,636	9,567,196	1,441.7
MITRE	3,432	2,435,118	709.5
Expert Curation			
Campaigns	136	37,106	272.8
Intrusion Sets	343	60,524	176.5
Malware	7,301	1,362,681	186.6
Reports	11,317	934,954	82.6
Threat Actors	27	2,264	83.9
Tools	238	19,926	83.7
Vulnerabilities	559,054	98,006,720	175.3
Total	674,848	190,886,516	282.9

Table 1: Token statistics of different sources in the PRIMUS-SEED dataset.

porate curated cyber threat intelligence (CTI) manually collected by threat experts. The statistics of PRIMUS-SEED are summarized in Tab.1.

Official Dump and Web Crawl. We specifically collect cybersecurity-related text from diverse sources, including Blogs, News, Books, Websites, Wikipedia, and MITRE, guided by prior pretraining work (Aghaei et al., 2022). For **Blogs** and **News**, we select content from government agencies, standards bodies, cybersecurity companies, media, and forums. Meanwhile, **Books** cover a wide range of cybersecurity topics, and we exclude covers, tables of contents, and appendices while treating each extracted page as a separate sample. We also collect **Webpages** from well-known cybersecurity companies, which may include product descriptions, company profiles, FAQs, and API documentation. In addition, **Wikipedia** does not provide a predefined cybersecurity subset, so we perform a custom filtering process. Each Wikipedia article is associated with one or more category tags, which can be further expanded into subcategory tags. Starting from the root category "*Computer Security*", we recursively traverse its subcategories, using GPT-4o to determine whether a category is cybersecurity-related[3]. This process yields 375 relevant categories, from which we extract corresponding Wikipedia articles. For **MITRE**, we leverage obsidian-mitre-attack[4], which converts STIX data from the official repository into readable Markdown.

Expert Curation. Another part of the data consists of CTI manually collected by our threat experts, categorized into Campaigns, Intrusion Sets, Malware, Threat Actors, Tools, Vulnerabilities, and Reports. Experts curate intelligence from open-source intelligence (OSINT), underground forums, and honeypots. OSINT includes public cybersecurity knowledge bases (e.g., MITRE ATT&CK, CAPEC, CVE, CWE), government advisories (e.g., CISA, Europol), and threat intelligence sharing platforms that provide structured insight into attack patterns, vulnerabilities, and emerging threats. In addition, experts monitor underground forums for discussions of cybercriminal activity, while honeypots capture real-world attack data to enhance intelligence gathering.

2.2.2 Preprocessing Pipeline

Considering the varying quality of texts from different sources, we adopt a preprocessing pipeline inspired by previous dataset works (Wenzek et al., 2020; Penedo et al., 2024; Raffel et al., 2019). Each source undergoes a dynamic combination of the following preprocessing steps.

LM Filtering. We use perplexity from a language model trained on English Wikipedia as a quality score. Specifically, we use a 5-gram KenLM language model (Heafield, 2011) due to its efficiency in processing large amounts of data. With this setup, we manually set an appropriate perplexity threshold for each source, and remove texts whose perplexity exceeds the threshold.

Deduplication. Deduplication has been correlated with improvements in model performance (Lee et al., 2022). We adopt FineWeb's deduplication strategy, using a fuzzy hash-based approach with MinHash. Specifically, we extract 5-grams from each document and compute MinHashes using 112 hash functions, split into 14 buckets of 8 hashes each to target documents at least 75% similar. Documents sharing the same 8 MinHashes in any bucket are considered duplicates.

C4 Filtering. We also apply the quality filters from the C4 dataset (Raffel et al., 2019). Although being smaller than FineWeb, C4 performs well on certain benchmarks and remains a common component in the pretraining mix of recent models such as LLaMA1 (Touvron et al., 2023). Its filtering rules include dropping lines without a terminal punctuation mark, mentioning javascript, or contain-

---

[3] The prompt is provided in the Appx.G (Fig.8)

[4] https://github.com/vincenzocaputo/obsidian-mitre-attack

ing "*terms-of-use*"/"*cookie policy*" statements, and dropping documents that are too short or contain "*lorem ipsum*" or a curly bracket ({). We apply all of these filters except for the terminal punctuation and curly bracket filters.

Heuristic Filtering. In addition to the above filters, we manually inspect each source and develop heuristic rules to further remove low-quality documents and outliers. For example, text containing phrases such as "*Your download will begin in a few seconds*" will be dropped.

2.2.3 Augmentation

We find that some web-scraped data contains valuable information but suffers from poor readability due to irregular formatting, such as inconsistent line breaks. To address this, we adopt a rewriting approach inspired by Cosmopedia[5], a reproduction of the high-quality synthetic dataset used in phi-1.5 (Li et al., 2023b). Specifically, we prompt an LLM to rewrite the given text into a specific style, including blog posts, textbooks, and Q&A formats[6]. To increase diversity, the rewriting LLM is randomly selected from GPT-4o, Llama-3.1-405B-Instruct, and DBRX (Mosaic, 2024).

2.3 PRIMUS-FINEWEB

2.3.1 Cybersecurity Classifier

Despite our efforts to collect as much cybersecurity text as possible in PRIMUS-SEED, it likely covers only a small fraction of the cybersecurity-related content on the internet. To further expand our dataset, we train a binary classifier based on TinyBERT (Jiao et al., 2020) to distinguish cybersecurity-related text from non-cybersecurity text and apply it to FineWeb, a cleaned dataset derived from Common Crawl. Specifically, we use PRIMUS-SEED as positive samples. Since cybersecurity text is only a small fraction of the web, we randomly take ten times as many samples from FineWeb and use them as negative samples to balance the dataset.

We then use the classifier to score all FineWeb texts on a scale from 0 to 1, where higher scores indicate greater cybersecurity relevance. The distribution in Fig.3 shows that lower scores correspond to a significant increase in text volume. To determine an appropriate threshold for filtering, we first verify that ***whether texts with higher scores are truly cybersecurity-related***. To do this, we leverage GPT-4o for accurate evaluation by dividing the scores into multiple bins, with dynamically adjusted bin sizes—smaller bins for lower scores—to account for the increased volume of data in lower score ranges. We randomly sample 50 texts from each bin and prompt GPT-4o[7] for classification. As shown in Fig.4, relevant text proportions remain above 60% at higher scores, but drop below 50% when scores fall below 0.003. Although incorporating some general text can help mitigate catastrophic forgetting (Sun et al., 2019), we prioritize maintaining a majority of cybersecurity content. Therefore, we set the final threshold at 0.003, which corresponds to 15.3B of FineWeb data.

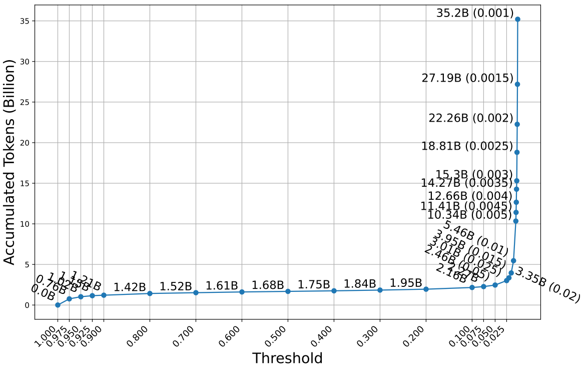


Figure 3: Cumulative token count in FINEWEB for texts with a cybersecurity score exceeding various thresholds.

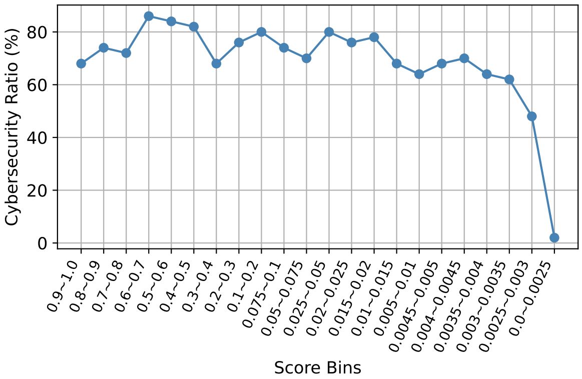


Figure 4: Ratio of cybersecurity-related text across different score bins in FINEWEB.

2.3.2 Deduplication Analysis

Upon inspecting the 15.3B dataset, we observed a significant amount of duplicate content. This occurs because FineWeb's ablation study found that deduplicating each Common Crawl snapshot separately yields better results than global deduplication, so FineWeb does not apply global dedu-

---

[5] https://github.com/huggingface/cosmopedia

[6] The prompt is provided in the Appx.G (Fig.9)

[7] The prompt is provided in the Appx.G (Fig.10)

Threshold	*Dedup.*	Samples	Tokens	*Avg.*
0.003	*False*	20,345,616	15.30B	751.88
0.003	*True*	3,386,733	2.57B	759.11
0.9	*False*	2,017,959	1.21B	600.37
0.9	*True*	393,154	0.23B	584.75

Table 2: Statistics of token counts before and after deduplication at different thresholds in the FineWeb.

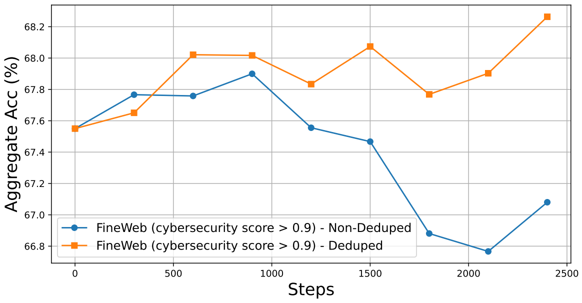


Figure 5: Comparison of deduplication on FineWeb cybersecurity data filtered at a classifier threshold 0.9.

plication. However, since our filtered dataset is much smaller, we conducted our own ablation study. Specifically, we extracted and deduplicated 1.21B tokens with a score above 0.9, reducing the number to 0.23B (pre- and post-deduplication token counts are listed in Tab.2), and we also sampled 0.23B tokens directly from the 1.21B set as an undeduplicated control group. We pre-trained Llama-3.1-8B-Instruct for two epochs on both datasets and found that the deduplicated dataset significantly outperformed the undeduplicated one on our aggregate of multiple-choice question (MCQ) cybersecurity tasks (to be introduced in Sec.3.1), as shown in Fig.5. Based on this observation, we finalized PRIMUS-FINEWEB with 2.57B deduplicated tokens filtered at a threshold of 0.003.

2.4 PRIMUS-INSTRUCT

After pre-training, we use PRIMUS-INSTRUCT for instruction fine-tuning to restore the instruction-following capability of the model. To achieve this, we design several hundred cybersecurity tasks covering common business scenarios, including explaining detected alerts, answering questions about retrieved security documents, analyzing executed suspicious commands, generating query languages for retrieving security events, and providing security recommendations and risk assessments for Terraform configurations. Each example is answered by GPT-4o, and we further use Claude 3.5 Sonnet (Anthropic, 2024) as a judge[8] to discard samples with insufficiently helpful answers. In addition, we include several hundred multi-turn conversations on general topics generated by GPT-4o. As a result, these form PRIMUS-INSTRUCT, with statistics in Tab.3.

Task	Samples
Cybersecurity-related Tasks	
Alert Explanation	100
Retrieved Security Doc QA	100
Suspicious Command Analysis	100
Security Event Query Generation	100
Terraform Security Misconfiguration Fix	96
General (Multi-turn)	
General Instruction Following	339

Table 3: Task distribution and corresponding sample counts in the PRIMUS-INSTRUCT dataset.

Dataset	Samples	Accepted	*Avg. Tokens*
		(o1-preview / DeepSeek-R1)	
CTI-MCQ	1000	806 / 768	692 / 672
CTI-RCM	1000	728 / 721	761 / 530
CTI-RCM-2021	1000	635 / 683	766 / 543
CTI-VSP	1000	231 / 312	1156 / 1395
CTI-ATE	60	2 / 5	1314 / 1731

Table 4: Statistics of the PRIMUS-REASONING dataset, distilled from o1-preview and DeepSeek-R1 on CTI-Bench questions, with only accepted correct samples.

2.5 PRIMUS-REASONING

With the release of OpenAI's reasoning model o1, an increasing number of studies have attempted to replicate its reasoning capabilities. One widely recognized approach is distillation, where reasoning samples with *self-reflection* from existing reasoning models are used to guide models in acquiring long-thought capabilities (Huang et al., 2024; Liu et al., 2024). To this end, we select cybersecurity reasoning tasks from CTI-Bench[9] (Alam et al., 2024) and prompt o1-preview one to two times per question to generate solutions with reasoning steps and reflection[10], applying rejection sampling to retain only the correctly answered samples. We also include DeepSeek-R1, obtained by directly querying its open-source model to access reasoning steps. The dataset statistics are shown in Tab.4.

---

[8] The judge prompt is provided in the Appx.G (Fig.11)

[9] A brief introduction to CTI-Bench is provided in Appx.D

[10] The prompt is provided in the Appx.G (Fig.12)

3 Evaluation Protocol

This section introduces the cybersecurity benchmarks (Sec.3.1) and evaluation settings (Sec.3.2) used to assess training performance.

3.1 Benchmarks

To assess the performance and training effectiveness of PRIMUS models, we evaluate them against seven cybersecurity benchmarks to measure their robustness and comprehensive understanding of security concepts, which we describe below.

CISSP. The Certified Information Systems Security Professional (CISSP) is a widely recognized cybersecurity certification that assesses both technical expertise and managerial competence. We construct an evaluation set based on multiple-choice questions from CISSP learning materials.

CTI-Bench. CTI-Bench is a benchmark for evaluating the reasoning and knowledge capabilities of LLMs in CTI. It consists of several subtasks, including CTI-RCM, CTI-VSP, CTI-ATE, and CTI-MCQ, which assess a model's ability to analyze vulnerabilities, infer security risks, extract attack techniques, and understand cybersecurity concepts.

CyberMetric. CyberMetric (Tihanyi et al., 2024) is a benchmark of human-verified multiple-choice questions designed to assess LLMs' cybersecurity knowledge across domains such as cryptography, network security, penetration testing, and compliance. We select a 500-question subset for evaluation as it is balanced and representative.

SecEval. SecEval (Li et al., 2023a) is a benchmark consisting of over 2,000 multiple-choice questions covering nine cybersecurity domains, including software security, cryptography, and network security. Generated by prompting GPT-4 with authoritative sources such as textbooks and official documentation, it provides a reliable measure of LLMs' cybersecurity proficiency.

3.2 Evaluation Settings

We integrate the above benchmarks into the `lm-evaluation-harness` (Gao et al., 2024) to ensure a standardized evaluation process. All evaluations are performed in the same environment to ensure fairness. We adopt the following two evaluation settings to evaluate models at different stages.

5-shot, w/o Chain-of-Thought (CoT). We prepend the first five questions from the benchmark along with their answers as context before the current question, guiding the model to output the correct answer directly instead of generating free-form responses. This setting is used to evaluate models after pretraining, when output formatting is more difficult to control.

0-shot, w/ CoT. We follow the evaluation setup from the OpenAI technical report benchmarks with `simple-eval`[11], using a standardized prompt[12] that allows the model to articulate its reasoning before producing the final answer. Due to the formatting variability of CoT responses, we use GPT-4o-mini to extract the final answers before scoring.

4 Training and Results

4.1 Overview

In this section, we present the entire training pipeline, which consists of four key stages. First, we expand the model's cybersecurity expertise and understanding through continued pre-training (Sec.4.2), which reinforces key cybersecurity concepts and enables the model to provide accurate information on security threats and mitigation strategies. Next, we restore its instruction-following capability through instruction fine-tuning (Sec.4.3), and further refine it through model merging to balance instruction-following and cybersecurity expertise. Finally, we train the model to develop reasoning capabilities on cybersecurity tasks (Sec.4.4)[13].

4.2 Pre-Training

We use Llama-3.1-8B-Instruct as our base model due to its wide community adoption and strong performance at the same parameter scale. We perform continued pre-training on two cybersecurity datasets: PRIMUS-SEED (Sec.2.2), which consists of curated cybersecurity text, and PRIMUS-FINEWEB (Sec.2.3), a filtered subset of cybersecurity content from FineWeb, to expand the model's cybersecurity expertise and understanding. To assess performance improvements, we evaluate the model against the seven cybersecurity benchmarks described in Sec.3.1 (5-shot, w/o CoT).

We train the model using the `NeMo` (NVIDIA, 2025) on four 8×H200 nodes, with training hyperparameters and details provided in Appx.E. To

---

[11] https://github.com/openai/simple-evals

[12] The prompt is provided in the Appx.G (Fig.13)

[13] The training hyperparameters for each stage are provided in the Appx.E

Model	**CISSP**	**CTI-MCQ**	**CTI-RCM**	**CTI-VSP**	**CTI-ATE**	**CyberMetric**	**SecEval**	***Agg.***
Llama-3.1-8B-Instruct	0.7073	0.6420	0.5910	1.2712	0.2721	0.8560	0.4966	2.29
+ Primus-Seed	0.7132	0.6608	0.6100	1.2848	0.2829	0.8600	0.4998	2.34↑2.1%
+ Primus-FineWeb	0.7191	0.6600	0.6680	1.1499	0.3006	0.8620	0.4984	2.56↑11.5%
+ Primus-Seed+FineWeb	**0.7230**	**0.6676**	**0.6780**	**1.0912**	**0.3140**	**0.8660**	**0.5007**	**2.66↑15.9%**

Table 5: Performance of continued pretraining on Llama across cybersecurity benchmarks. The last three rows indicate pretraining with Primus-Seed, Primus-FineWeb, and their combination. CTI-VSP is scored using Mean Absolute Deviation ***(lower is better)***, CTI-ATE uses F1 score, and the others use accuracy. The aggregate score *(Agg.)* is the sum of all benchmarks, with CTI-VSP negated. The best results are highlighted in **bold**.

analyze the impact of different datasets, we conduct an ablation study by pre-training the model separately on each dataset and jointly on both for two epochs. The results in Tab.5 show that pre-training on either dataset improves the cybersecurity performance in the aggregate evaluation score. However, the largest improvement, ***15.9%***, is observed when pre-training on the combined dataset, so we adopt this model as the Llama-Primus-Base for subsequent training stages[14].

4.3 Instruction Fine-Tuning and Merge

While Llama-Primus-Base gains enhanced cybersecurity knowledge and understanding from pretraining, it tends to perform text completion rather than follow instructions. To address this, we further fine-tune it using the `LLaMA-Factory` (Zheng et al., 2024) on 4×A100 GPUs for two epochs with Primus-Instruct (Sec.2.4), a carefully curated mixed dataset of cybersecurity tasks and general conversations, resulting in Llama-Primus-Instruct. In addition to the cybersecurity benchmarks, we also introduce MT-Bench (Zheng et al., 2023), a multi-turn instruction-following evaluation benchmark spanning multiple domains using GPT-4 as a judge, which scores helpfulness on a scale of 1 to 10, allowing us to evaluate the overall instruction-following performance of the model. The results are shown in Tab.6, where the MT-Bench score and the aggregated cybersecurity benchmark score are further aggregated with a weight of 30/70 in the rightmost column.

Llama-Primus-Instruct maintains its advantage in cybersecurity while achieving an MT-Bench score of 7.91. However, this remains lower than the 8.35 of Llama, resulting in a limited improvement in the aggregated score (2.4%). To mitigate this, we apply DARE-TIES (Yu et al., 2024; Yadav et al., 2023), a model merging technique that balances diverse capabilities—specifically, instruction-following and cybersecurity expertise in our case. We conduct a grid search over the merging ratio, setting Llama-Primus-Instruct:Llama-3.1-8B-Instruct to $(0.5+w)$:$(0.5-w)$ and varying w from 0 to 0.5 in steps of 0.05. The optimal ratio that maximizes the aggregated score is found to be 0.75:0.25, with the merged model chosen as Llama-Primus-Merged. Notably, this configuration retains cybersecurity performance comparable to Llama-Primus-Instruct while restoring the MT-Bench to 8.29, almost equal to Llama, resulting in a ***5.4%*** improvement in the aggregated score[15].

4.4 Reasoning Fine-Tuning

We further distill Llama-Primus-Merged using Primus-Reasoning (Sec.2.5), a high-quality dataset of cybersecurity task reasoning steps obtained from o1-preview and DeepSeek-R1, to equip it with reasoning and self-reflection capabilities. This approach has been successfully demonstrated in previous work such as S1 (Muennighoff et al., 2025) and Sky-T1 (Team, 2025). Since Primus-Reasoning is constructed from CTI-Bench tasks, we exclude them from the evaluation and choose CISSP as a representative metric, as it also emphasizes reasoning rather than just factual recall. The results are presented in Tab.7.

As shown in the table, both Llama-3.1-8B-Instruct and Llama-Primus-Merged improve with CoT over direct answer generation. Notably, Llama-Primus-Merged achieves the largest gain, even outperforming DeepSeek-R1-Distill-Llama-8B[16] (0.7603 vs. 0.7399) with the fewest tokens, suggesting stronger cybersecurity knowledge benefits reasoning. After fine-tuning on Primus-Reasoning (rows starting with +), token usage increases while accuracy further improves; distillation on the combined o1-preview and DeepSeek-R1 data achieves the largest improve-

---

[14] We also experimented with a 70B model in **Q2** of Appx.A (FAQs)

[15] We provide more details in **Q4** and **Q5** of Appx.A (FAQs)

[16] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B

Model	**CISSP**	**CTI-MCQ**	**CTI-RCM**	**CTI-VSP**	**CTI-ATE**	**CyberMetric**	**SecEval**	**MT-Bench**	***Agg.***
Llama-3.1-8B-Instruct	0.7073	0.6420	0.5910	1.2712	0.2721	0.8560	0.4966	**8.3491**	4.11
Llama-Primus-Instruct	0.7132	**0.6660**	**0.6660**	**1.1161**	0.3348	0.8640	0.4943	7.9063	4.21↑2.4%
Llama-Primus-Merged	**0.7191**	0.6656	0.6620	1.1233	**0.3387**	**0.8660**	**0.5062**	8.2938	**4.33↑5.4%**

Table 6: Performance comparison of Llama, the instruction-tuned Primus model, and their merge on cybersecurity and general benchmarks. The aggregated score *(Agg.)* is computed as $0.3\times$ MT-Bench + $0.7\times$ aggregated cybersecurity score (sum of all benchmarks except MT-Bench, with CTI-VSP negated due to the use of Mean Absolute Deviation, where lower is better). The best results are highlighted in **bold**.

Model	**CISSP**	***Avg. Tokens***
w/o CoT, 5-shot		
Llama-3.1-8B-Instruct	0.7073	1
Llama-Primus-Merged	0.7191 ↑1.67%	1
w/ CoT, 0-shot		
Llama-3.1-8B-Instruct	0.7288 ↑3.03%	279.69
+ *Distilled from o1-preview*	0.7583 ↑7.21%	646.94
+ *Distilled from DeepSeek-R1*	0.7859 ↑11.1%	1667.56
+ *Distilled from (o1 + R1)*	0.7780 ↑10.0%	1615.54
Llama-Primus-Merged	0.7603 ↑7.49%	241.92
+ *Distilled from o1-preview*	0.7780 ↑10.0%	726.96
+ *Distilled from DeepSeek-R1*	0.8075 ↑14.2%	1483.94
+ *Distilled from (o1 + R1)*	0.8193 ↑**15.8%**	1467.40
o1-preview	0.8035	1054.91
DeepSeek-R1	0.8212	1229.32
DeepSeek-R1-Distill-Llama-8B	0.7399	1542.10

Table 7: Effect of PRIMUS-REASONING fine-tuning (on o1-preview, DeepSeek-R1, and their combination), evaluated on CISSP. ↑ indicates the percentage improvement over Llama without CoT and in the 5-shot setting. The best improvement is highlighted in **bold**.

Benchmark	**ECE (%)**		
	Llama-3.1-8B-Instruct	Llama-Primus-Base	Llama-Primus-Merged
CISSP	7.22	4.59	4.55
CTI-MCQ	11.01	2.03	5.52
CyberMetric	4.11	3.41	2.57
Average	7.45	3.34↓55.17%	4.21↓43.49%

Table 8: Expected Calibration Error (ECE) across cybersecurity benchmarks (with 10 bins).

Metric	**Llama-3.1-8B-Instruct**	**Llama-Primus-Base**	**Llama-Primus-Merged**
Accuracy (%)	67.56	66.29	66.59
ECE (%)	5.99	6.07	5.56

Table 9: Accuracy and ECE across models on MMLU.

ment ***(15.8%)***. Interestingly, comparing DeepSeek-R1-Distill-Llama-8B (0.7399) and Llama-3.1-8B-Instruct after distillation (0.7583 / 0.7859 / 0.7780) may suggest that domain-specific reasoning distillation yields better in-domain performance than general-domain distillation.

5 Domain Calibration Analysis

In cybersecurity applications, a model's confidence score is often a critical indicator for deciding whether to escalate issues for human intervention, such as sending alerts to security analysts. For this to work, the confidence score must accurately reflect the true accuracy. After multi-stage training in the cybersecurity domain, we found that our model had a significantly lower Expected Calibration Error (ECE) (Guo et al., 2017) on cybersecurity-related questions. This suggests our model's confidence is more aligned with its actual accuracy. The ECE measures the average discrepancy between a model's confidence and its empirical accuracy.

Specifically, we re-evaluated the cybersecurity multiple-choice tasks (CISSP, CTI-MCQ, and CyberMetric). We took the token probability of the output answer (A/B/C/D) as the confidence score and calculated the ECE, as shown in Tab.8. The ECE of our model on cybersecurity questions was reduced by **half**, indicating that the model is better calibrated and thus more reliable in practical applications, especially those involving confidence thresholds. Additionally, evaluation on general-domain questions (e.g., MMLU) (Hendrycks et al., 2021) showed no significant change (see Tab.9).

Recent work has sought to improve LLM calibration by reducing ECE through specialized training methods (Xu et al., 2024b). However, leveraging domain-specific data for this purpose remains unexplored. We posit that our approach could provide valuable insights into confidence calibration.

6 Conclusion

In this work, we explore adapting other successful domain-specific LLM approaches to cybersecurity and contribute a series of datasets covering different stages of LLM training, including pre-training, instruction fine-tuning, and reasoning distillation, each of which has been validated to improve cybersecurity performance. To our knowledge, this is the ***first*** study to systematically strengthen the cybersecurity skills of an LLM across multiple stages of training, and we will release all datasets and models to encourage further community research.

Limitations

Although this work covers the various stages of LLM training, it has the following limitations:

- Due to limited computational resources, our experiments primarily focus on 8B-scale models, leaving the effectiveness of scaling to larger models (e.g., 405B or 671B) unknown.
- Our exploration of RL remains limited. Recent work by DeepSeek-R1 has demonstrated that GRPO (Zhang et al., 2024b) combined with only rule-based rewards (e.g., correctness and format compliance) can achieve performance comparable to o1. We believe this is also a promising direction for cybersecurity applications and leave it as future work.

Ethics Statement

We used Garak (Derczynski et al., 2024), a toolkit that probes for hallucination, data leakage, prompt injection, misinformation, toxicity generation, jailbreaks, and many other vulnerabilities, to evaluate Llama-Primus-Merged. The results showed no significant differences compared to Llama (Appx.H). However, we still emphasize that the user is solely responsible for the content generated with the Primus model, as it lacks mechanisms to handle the disclosure of harmful, biased, or toxic content. Therefore, we strongly recommend that Primus be used for research purposes only. If used in production for natural language generation, users should independently assess the risks and implement appropriate safeguards.

References

Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. 2022. Securebert: A domain-specific language model for cybersecurity. In *International Conference on Security and Privacy in Communication Systems*, pages 39–56. Springer.

Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. 2024. CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track.*

Anthropic. 2024. Introducing claude 3.5 sonnet. Accessed: 2025-02-13.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. 2024. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*.

Common Crawl. 2008. Common crawl. `https://commoncrawl.org/`. Accessed: 2025-02-13.

Leon Derczynski, Erick Galinkin, Jeffrey Martin, Subho Majumdar, and Nanna Inie. 2024. garak: A Framework for Security Probing Large Language Models. `https://garak.ai`. Accessed: 2025-02-16.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, volume 1. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, and Norbert Tihanyi. 2024. Generative ai and large language models for cyber security: All insights you need. *Available at SSRN 4853709*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. 2023. Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*, 36:5539–5568.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Diptiben Ghelani. 2022. Cyber security, cyber threats, implications and future perspectives: A review. *Authorea Preprints*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*.

Jackaduma. 2021. Secbert: A pretrained language model for cyber security text. `https://github.com/jackaduma/SecBERT/`. Accessed: 2025-02-03.

Nan Jiang, Chengxiao Wang, Kevin Liu, Xiangzhe Xu, Lin Tan, and Xiangyu Zhang. 2023. Nova: Generative language models for binaries. *arXiv preprint arXiv:2311.13721*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. BioMistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864, Bangkok, Thailand. Association for Computational Linguistics.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and S Yu Philip. 2024. Large language models in law: A survey. *AI Open*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Guancheng Li, Yifeng Li, Wang Guannan, Haoyu Yang, and Yang Yu. 2023a. Seceval: A comprehensive benchmark for evaluating cybersecurity knowledge of foundation models. https://github.com/XuanwuAI/SecEval.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Yuchong Li and Qinghui Liu. 2021. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. *Energy Reports*, 7:8176–8186.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Mosaic. 2024. Introducing dbrx: A new state-of-the-art open llm. Accessed: 2025-02-13.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

NVIDIA. 2025. Nemo: A scalable generative ai framework. Accessed: 2025-02-13.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Youngja Park and Weiqiu You. 2023. A pretrained language model for cyber threat intelligence. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 113–122.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Alec Radford. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Priyanka Ranade, Aritran Piplai, Anupam Joshi, and Tim Finin. 2021. Cybert: Contextualized embeddings for the cybersecurity domain. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3334–3342. IEEE.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *Preprint*, arXiv:2412.02595.

Fan-Keng Sun, Cheng-Hao Ho, and Hung yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Tiezhu Sun, Kevin Allix, Kisub Kim, Xin Zhou, Dongsun Kim, David Lo, Tegawendé F Bissyandé, and Jacques Klein. 2023. Dexbert: Effective, task-agnostic and fine-grained representation learning of android bytecode. *IEEE Transactions on Software Engineering*.

NovaSky Team. 2025. Sky-t1: Train your own o1 preview model within $450. https://novasky-ai.github.io/posts/sky-t1. Accessed: 2025-01-09.

Norbert Tihanyi, Mohamed Amine Ferrag, Ridhi Jain, Tamas Bisztray, and Merouane Debbah. 2024. Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating llms in cybersecurity knowledge. In *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 296–302.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hao Wang, Zeyu Gao, Chao Zhang, Zihan Sha, Mingyang Sun, Yuchen Zhou, Wenyu Zhu, Wenju Sun, Han Qiu, and Xi Xiao. 2024a. Clap: Learning transferable binary code representations with natural language supervision. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 503–515.

Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu, Guoyin Wang, and Eduard Hovy. 2024b. Reinforcement learning enhanced llms: A survey. *arXiv preprint arXiv:2412.10400*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

HanXiang Xu, ShenAo Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and HaoYu Wang. 2024a. Large language models for cyber security: A systematic literature review. *arXiv preprint arXiv:2405.04760*.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. 2024b. SaySelf: Teaching LLMs to express confidence with self-reflective rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5985–5998, Miami, Florida, USA. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2024. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, 55(1):90–112.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR.

Jie Zhang, Haoyu Bu, Hui Wen, Yu Chen, Lun Li, and Hongsong Zhu. 2024a. When llms meet cybersecurity: A systematic literature review. *arXiv preprint arXiv:2405.03644*.

Jie Zhang, Hui Wen, Liting Deng, Mingfeng Xin, Zhi Li, Lun Li, Hongsong Zhu, and Limin Sun. 2023a. Hackmentor: Fine-tuning large language models for cybersecurity. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 452–461.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023b. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Wei Zhang, Ming Li, Hao Wang, and Yang Liu. 2024b. Deepseekmath: Scalable math pre-training and group relative policy optimization for mathematical reasoning. *arXiv preprint arXiv:2402.03300*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Hongjian Zhou, Fenglin Liu, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S Chen, Peilin Zhou, Junling Liu, et al. 2023. A survey of large language models in medicine: Progress, application, and challenge. *arXiv preprint arXiv:2311.05112*.

A FAQs

- **Q1**: ***What are the implementation details, such as the training hyperparameters and the prompts used for the LLM during dataset construction?***

These details are provided in the appendix. The training hyperparameters are listed in Appx.E, and the prompts used for dataset construction are included in Appx.G.

- **Q2**: ***The experiments primarily target 8B models. A natural follow-up is whether these datasets generalize to larger models, i.e., whether they can also improve the cybersecurity performance of larger models?***

Yes, we extended our experiments to a 70B model by further pre-training Llama-3.1-Nemotron-70B-Instruct to obtain Llama-Primus-Nemotron-70B-Base. In addition to the dataset used for the 8B model, we supplemented its pre-training corpus with 7.6B tokens of cybersecurity content filtered from Nemotron-CC (Su et al., 2024) (see Appx.C). The results in Tab.10 show an **11.2%** gain in the aggregated cybersecurity benchmark score. We will also release this model under the MIT license. Due to its high computational cost, we did not conduct the dataset-combination ablation study on the 70B model that we performed on the 8B experiments.

- **Q3**: ***Since LLMs (e.g., Claude) were used during dataset construction, has their reliability been evaluated?***

Yes, we conducted an experiment to measure the discrepancy between human experts and LLM judges under identical prompts. Specifically, in Sec.2.4 we used Claude 3.5 Sonnet to rate the helpfulness of responses in PRIMUS-INSTRUCT, discarding those that were not helpful enough[17]. To validate Claude's reliability as a judge, we randomly selected ten examples per task for human experts to score, then computed the differences between human, GPT-4o, and Claude ratings.

The discrepancies are reported in Tab.11. Since PRIMUS-INSTRUCT's responses were generated by GPT-4o, we found that it tended to favor its own answers, which is consistent with findings in LLM-as-a-Judge (Zheng et al., 2023). This resulted in slightly larger discrepancies compared to Claude. Based on these results, we found that the gap between LLM-based and human scoring remained within an acceptable range.

- **Q4**: ***What is the training objective of* PRIMUS-INSTRUCT*?***

We would like to clarify that our primary goal with the SFT data was ***not*** to further improve the model's cybersecurity capabilities. Instead, our goal was to help the model regain its instruction-following ability ***without forgetting*** the cybersecurity knowledge acquired during pre-training. This can be viewed as a continual learning problem involving two tasks: "retaining cybersecurity knowledge" and "learning instruction following". According to LAMOL (Sun et al., 2019), language models often suffer from catastrophic forgetting when trained sequentially on multiple tasks—learning a new task tends to overwrite knowledge from previous ones.

A common solution is to interleave data from previous tasks into the new task to mitigate forgetting. Inspired by this, we designed our cybersecurity SFT data to combine both instruction-following and domain-specific knowledge, hoping that the model would learn instruction-following while retaining its earlier cybersecurity understanding. As shown in Tab.6, the results suggest that the model was able to recover instruction-following ability without significant loss in cybersecurity performance.

- **Q5**: ***Why does* PRIMUS-INSTRUCT *appear to have a relatively small number of samples (~1k)?***

In fact, PRIMUS-INSTRUCT was selected from a larger pool of data. For each task, we initially generated 300–400 samples and rated their helpfulness (on a scale of 1 to 10) using the judge prompt in Fig.11. Only the top 100 samples with scores of at least 8 were retained (Tab.12).

Since we first performed SFT and then merged the resulting model with Llama-3.1-8B-Instruct to balance cybersecurity capabilities and instruction-following ability (Sec.4.3), the ***SFT and merging steps should be considered as a unified process***. We therefore evaluated the combined effect of both. Specifically, we conducted SFT on Llama-Primus-Base separately using both the unfiltered version (2,239 samples) and the filtered high-quality version (835 samples) from Tab.12. Each resulting SFT model was then merged with Llama-3.1-8B-Instruct for comparison.

The merging process involves subtracting each model's weights from the same base model (Llama-3.1-8B) to obtain two task vectors: one representing cybersecurity knowledge, and the other repre-

---

[17] The judge prompt is provided in the Appx.G (Fig.11)

Model	**CISSP**	**CTI-MCQ**	**CTI-RCM**	**CTI-VSP**	**CTI-ATE**	**CyberMetric**	**SecEval**	***Agg.***
Llama-3.1-Nemotron-70B-Instruct	0.8527	0.6900	0.6590	1.1893	0.3905	**0.9380**	0.7177	3.06
Llama-Primus-Nemotron-70B-Base	**0.8703**	**0.7148**	**0.7410**	**1.0281**	**0.4540**	0.9280	**0.7208**	**3.40↑11.2%**

Table 10: Performance comparison of Llama-3.1-Nemotron-70B-Instruct and Llama-Primus-Nemotron-70B-Base on cybersecurity benchmarks. CTI-VSP is scored using Mean Absolute Deviation ***(lower is better)***, CTI-ATE uses F1 score, and the others use accuracy. The aggregate score *(Agg.)* is the sum of all benchmarks, with CTI-VSP negated. The best results are highlighted in **bold**.

Task	**MAE** (Claude)	**MAE** (GPT-4o)
Alert Explanation	0.8	1.0
Retrieved Security Doc QA	0.7	1.1
Suspicious Command Analysis	0.4	1.0
Security Event Query Generation	1.0	0.8
Terraform Security Misconfiguration Fix	1.1	0.4
Average	0.8	0.86

Table 11: Mean absolute error (MAE) between human expert scores and LLM scores across different PRIMUS-INSTRUCT tasks.

Task	**Samples**	**Accepted**
Alert Explanation	400	100
Retrieved Security Doc QA	400	100
Suspicious Command Analysis	400	100
Security Event Query Generation	400	100
Terraform Security Misconfiguration Fix	300	96
Total	1,900	496
+ General Instruction Following (339)	2,239	835

Table 12: Initially designed (unfiltered) and accepted (filtered) sample counts per task, where accepted refers to the top 100 samples with a judge score ≥ 8.

senting instruction-following ability. The results are shown in Tab.13. We found that applying SFT with a small amount (835) of high-quality data on Llama-Primus-Base before merging yields the best results in both the Cybersecurity Aggregate Score (2.63) and the MT-Bench score (8.29). This is why we chose the filtered high-quality version as PRIMUS-INSTRUCT.

- **Q6**: ***Were more baselines compared?***

As shown in Fig.2, most existing cybersecurity-specific LLMs are fine-tuned for narrow tasks, such as password strength detection or malware detection from assembly code. Studies aimed at improving general cybersecurity domain knowledge in LLMs are relatively rare, and to the best of our knowledge, we are the ***first*** to pursue this through pre-training.

The primary goal of our comparisons is to demonstrate the effectiveness of our dataset by showing the performance gains of the same base model before and after training on it. Comparisons with other cybersecurity LLMs are difficult to interpret fairly due to differences in training methods and base models. However, to make our findings more convincing, we also identified existing models that incorporate domain knowledge into LLMs via SFT or DPO, and conducted comparisons with them. As shown in Tab.14, our model consistently outperforms these alternatives (Zhang et al., 2023a).

- **Q7**: ***What are the structures of the datasets proposed in this paper?***

The schema for each dataset is provided in Appx.B, including the fields it contains, their descriptions, and license information.

- **Q8**: ***Could you provide some sample responses from Llama-Primus that demonstrate its capabilities?***

In Appx.F, we present and compare the responses of Llama-Primus-Merged and Llama-3.1-8B-Instruct on a question selected from the CTI-MCQ dataset.

- **Q9**: ***Could the improvement in CISSP scores after training on PRIMUS-REASONING be attributed to the inclusion of CISSP answers, thereby leading to potential data leakage?***

To ensure the rigor of our experiments, we applied the standard N-gram decontamination method from EleutherAI's `llm-eval-harness` to identify any overlapping content, following the approach described in the GPT-3 paper (Brown et al., 2020) (with N set to 13 by default). Specifically, we concatenated each CISSP question and answer pair, and likewise concatenated the message contents of each sample in PRIMUS-REASONING, then generated N-grams for both and checked for duplicates. The results are shown in Tab.15.

We found only 5 overlapping samples between PRIMUS-REASONING and the CISSP benchmark when lowering N to 8. However, manual inspection

Model	**Base Model for Merge**	**Merge Model 1 (Task Vector 1)**	**Merge Model 2 (Task Vector 2)**	**Cybersecurity *Agg.* Score**	**MT-Bench**
Llama-Primus-Merged (from unfiltered SFT)	Llama-3.1-8b	Llama-Primus-Base -> SFT (2,239 samples)	Llama-3.1-8b-Instruct	2.44	7.97
Llama-Primus-Merged (from filtered SFT)	Llama-3.1-8b	Llama-Primus-Base -> SFT (835 samples)	Llama-3.1-8b-Instruct	2.63	8.29
Llama-3.1-8b-Instruct	–	–	–	2.29	8.35

Table 13: Comparison of merged PRIMUS models using different versions of the SFT dataset on cybersecurity and MT-Bench benchmarks. The first row refers to applying SFT on Llama-Primus-Base using the unfiltered 2,239 samples from Tab.12 before merging with Llama-3.1-8B-Instruct, while the second row uses the filtered high-quality 835-sample version for SFT prior to merging.

Benchmark	**ZySec-AI/ SecurityLLM**	**HackMentor/ Llama-7b-lora-iio**	**HackMentor/ Vicuna-7B-lora-iio**	**Llama-Primus-Merged**
CISSP	0.6012	0.2908	0.4519	**0.7191**
CTI-MCQ	0.5676	0.4184	0.5104	**0.6656**
CTI-RCM	0.4420	0.2770	0.2810	**0.6620**
CTI-ATE	0.0286	0.2671	0.1411	**0.3387**
CTI-VSP	1.3923	2.1172	1.6205	**1.1233**
CyberMetric	0.8140	0.3640	0.6760	**0.8660**
SecEval	0.4641	0.3640	0.3413	**0.5062**

Table 14: Performance comparison with existing cybersecurity LLMs across benchmarks. CTI-VSP is scored using Mean Absolute Deviation ***(lower is better)***, CTI-ATE uses F1 score, and the others use accuracy. The best results are highlighted in **bold**.

N-gram	**Overlap Count**
13	0
12	0
11	0
10	0
9	0
8	5

Table 15: Counts of overlapping N-grams between PRIMUS-REASONING and CISSP.

revealed that these overlaps were limited to generic question stems such as "*Which of the following best describes how an*" and "*Which of the following is an example of*," rather than actual cybersecurity concepts or substantive content. Therefore, we believe that potential data leakage is negligible.

B Dataset Details

Fields. All datasets of PRIMUS-PRETRAINING (PRIMUS-SEED, PRIMUS-FINEWEB, and PRIMUS-NEMOTRON-CC) have the same structure and set of fields, as shown in Tab.16. Similarly, PRIMUS-INSTRUCT and PRIMUS-REASONING have a unified schema, which is detailed in Tab.17.

License. All datasets proposed in this paper are released under the ODC-BY license. Additionally, compliance with the Terms of Use (ToU) or licenses of the original content sources is required. Some datasets are derived from existing ones. For example, PRIMUS-FINEWEB originates from FineWeb, and PRIMUS-NEMOTRON-CC stems from Nemotron-CC. Both of these datasets are in turn based on Common Crawl, which requires compliance with its ToU. The Common Crawl ToU also requires adherence to the ToU of the original content owners.

As indicated in the field descriptions, all datasets of PRIMUS-PRETRAINING include a url field that points to the original content source. We expect users to also respect the ToU or licenses of the original content providers.

C PRIMUS-NEMOTRON-CC

We further extracted cybersecurity-related text from Nemotron-CC (Su et al., 2024), which claims higher quality and more "unique" tokens than FineWeb (i.e., tokens remaining after global fuzzy deduplication). We scored each Nemotron-CC sample using the binary classifier trained in Sec.2.3 and partitioned the scores into multiple intervals. For each score interval, we sampled 1,000 examples, grouped them by length, sent them to

Field	**Description**
url	The original source URL link corresponding to the sample.
source	A coarse category of the sample's source, such as Wikipedia or MITRE.
content	The textual content of the sample.
time	The crawling time of the sample, recorded in ISO 8601 format (e.g., `2024-12-31T00:00:00`). For PRIMUS-FINEWEB and PRIMUS-NEMOTRON-CC, only the year is recorded; to maintain format consistency, we append `-12-31T00:00:00` after the year.

Table 16: Fields contained in each sample of PRIMUS-SEED, PRIMUS-FINEWEB, and PRIMUS-NEMOTRON-CC.

Field	**Description**
messages	The conversation history stored in an alternating `user/assistant` format, e.g., `[{"role": "user", "content": "..." }, {"role": "assistant", "content": "..." }, ... ]`.
prompt	The first prompt from the user, i.e., the content of the first messages entry.
prompt_id	A unique identifier for the sample.

Table 17: Fields contained in each sample of PRIMUS-INSTRUCT and PRIMUS-REASONING.

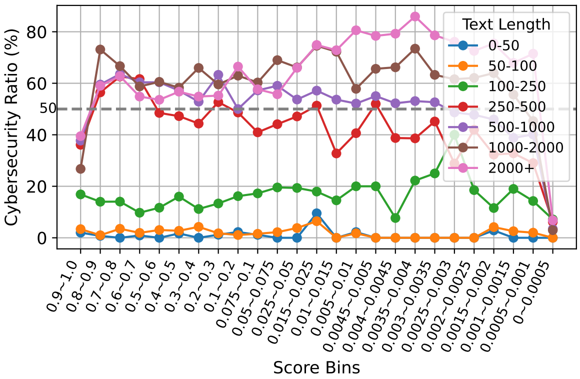


Figure 6: Ratio of cybersecurity-related text across different score bins in NEMOTRON-CC, grouped by sample length.

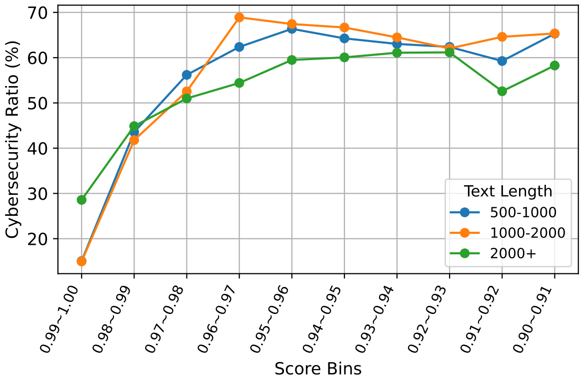


Figure 7: Ratio of cybersecurity-related text across score bins in the 1.0 ~ 0.9 range in NEMOTRON-CC.

GPT-4o-mini[18] to verify whether they were truly cybersecurity-related, and then calculated the proportion of confirmed samples. The results are shown in Fig.6.

We observed that when sample length is under 500 or the score is below 0.003, the proportion of cybersecurity-related samples falls below 50% in most cases. Therefore, we only retain samples that exceed 500 in length and have a score greater than 0.003. Interestingly, the proportion of cybersecurity samples also declines when the score is very high (> 0.9), likely because our classifier was trained on FineWeb. Thus, we performed a finer-grained analysis on the > 0.9 interval, as shown in Fig.7. Once the score exceeds 0.98, the related proportion drops below 50%, so we only keep samples with scores under 0.98.

Due to computational constraints, we were unable to include all samples that met the above criteria. Instead, we computed the total number of tokens (for samples with length > 500) within different score ranges, as shown in Tab.18. Given our computing budget, we aimed to limit the 70B model's pretraining dataset to approximately 10B tokens. As a result, we selected the 0.98 ~ 0.0175 score range, which contains 7.6B tokens, for inclusion in PRIMUS-PRETRAINING. This dataset will also be **released**.

Cybersecurity Score Bin	**Filtered Tokens**	***Dedup.***
0.98 ~ 0.85	2.22B	2.05B
0.98 ~ 0.30	4.07B	3.75B
0.98 ~ 0.05	6.02B	5.53B
0.98 ~ 0.0175	8.31B	7.63B
0.98 ~ 0.015	8.89B	8.86B
0.98 ~ 0.01	10.97B	10.05B
0.98 ~ 0.0075	13.10B	11.98B

Table 18: Token counts before and after deduplication for PRIMUS-NEMOTRON-CC samples (length > 500) across different score bins.

---

[18] The prompt is provided in Appx.G (Fig.10)

D CTI-Bench

CTI-Bench is a benchmark for evaluating the reasoning and knowledge capabilities of LLMs in CTI. It consists of several subtasks, including CTI-RCM, CTI-VSP, CTI-ATE, and CTI-MCQ, which assess a model's ability to analyze vulnerabilities, infer security risks, extract attack techniques, and understand cybersecurity concepts. The following paragraphs present a overview of each subtask.

CTI-RCM (Root Cause Mapping). This task maps Common Vulnerabilities and Exposures (CVE) descriptions to Common Weakness Enumeration (CWE) categories, essentially classifying vulnerabilities. CWE consists of over 900 categories, often with subtle differences that make misclassification highly likely. The model must reason about the true root cause of the vulnerability and ***infer*** the most appropriate weakness type rather than relying on textual matches.

CTI-VSP (Vulnerability Severity Prediction). Given a vulnerability description, the task is to calculate its CVSS (Common Vulnerability Scoring System) score, which assesses severity. CVSS scoring dimensions include attack vectors (AV), required privileges, impact scope, and more. However, CVE descriptions often do not explicitly provide this information. The model must understand the vulnerability mechanism, ***infer*** possible exploitation methods and impact scope, and map them to CVSS metrics.

CTI-ATE (Attack Technique Extraction). This task extracts MITRE ATT&CK technique IDs from a given threat behavior description. Threat descriptions are often non-standardized and context-dependent, using different terminology or embedding multiple attack techniques. The model must ***reason*** about the attack process, synthesizing scattered information to identify possible tactics, techniques, and procedures (TTPs) and map them to the correct MITRE ATT&CK technique IDs.

CTI-MCQ. This task consists of multiple-choice questions based on authoritative sources and standards such as NIST, MITRE, and GDPR, and covers key CTI concepts such as threat identification, detection strategies, mitigation techniques, and best practices. While some questions focus on factual recall, our review found many require cross-concept ***reasoning***, such as inferring applicable scenarios for different attack techniques, evaluating the effectiveness of security strategies, or understanding the potential impact of certain vulnerabilities.

E Training Hyperparameters

This section details the hyperparameters used in each training stage of our experiments.

E.1 Pre-Training

[8B Model]

Provider: `AWS`
Framework: `NeMo`
Hardware: *4 nodes, each with 8 × H200*
Training Time: *30 hours (Primus-Seed+Primus-FineWeb)*
Epochs: *2*
Learning Rate: *1e-6*
Pipeline Model Parallel Size: *4*
Tensor Model Parallel Size: *8*
Context Parallel Size: *1*
Global Batch Size: *12*
Micro Batch Size: *12*
Warmup Ratio: *0.05*
Scheduler: *Cosine Annealing*
Sequence Length: *16,384*

[70B Model]

Provider: `NVIDIA`
Framework: `NeMo`
Hardware: *4 nodes, each with 8 × H100*
Training Time: *175 hours*
Epochs: *2*
Learning Rate: *1e-6*
Pipeline Model Parallel Size: *4*
Tensor Model Parallel Size: *8*
Context Parallel Size: *1*
Global Batch Size: *8*
Micro Batch Size: *1*
Warmup Ratio: *0.05*
Scheduler: *Cosine Annealing*
Sequence Length: *11,264*

E.2 Instruction Fine-Tuning

Provider: `Azure`
Framework: `LLaMA-Factory`
Hardware: *4 × A100*
Training Time: *2 hours*
Epochs: *2*
Learning Rate: *1e-6*
Deepspeed: *ZeRO Stage-3 with CPU Offload*
Per Device Train Batch Size: *1*

Warmup Ratio: *0.1*
Scheduler: *Cosine*
Cutoff Length: *16,384*

E.3 Reasoning Fine-Tuning

Provider: `Azure`
Framework: `LLaMA-Factory`
Hardware: *4 × A100*
Training Time: *2.5 hour*
Epochs: *3*
Learning Rate: *1e-6*
Deepspeed: *ZeRO Stage-3 with CPU Offload*
Per Device Train Batch Size: *3*
Warmup Ratio: *0.1*
Scheduler: *Cosine*
Cutoff Length: *8,192*

F Sample Outputs

In this section, we present an example of the output of Llama-Primus-Merged for a question selected from the CTI-MCQ dataset. We removed the answer choices and prompted the model to respond freely. Then, we compared its response with that of Llama-3.1-8B-Instruct.

Question. *What mitigation technique can help prevent the installation of unauthorized browser extensions as per the MITRE ATT&CK framework?*

Llama-Primus-Merged. *As per the MITRE ATT&CK framework, the technique to help prevent the installation of unauthorized browser extensions is T1176: Software Extensions. This technique involves monitoring and restricting the installation of browser extensions to prevent malicious extensions from being installed.*

- *Whitelisting: Maintain a list of approved browser extensions (. . .)*
- *Blacklisting: Maintain a list of known malicious browser extensions (. . .)*
- *Group Policy: Use group policy settings to restrict the installation of browser extensions.*
- *Browser Settings: Configure browser settings to disable the installation of extensions or to only allow extensions from trusted sources.*
- *Regular Security Audits: Regularly audit browser extensions to identify and remove any unauthorized extensions.*

By implementing these controls, you can (. . .)

Llama-3.1-8b-Instruct. *The MITRE ATT&CK framework recommends the following mitigation technique to prevent the installation of unauthorized browser extensions:*

- *Restrict Browser Extensions: (. . .)*
- *Use a Secure Browser: (. . .)*
- *Monitor Browser Extensions: (. . .)*
- *Use a Browser Extension Manager: (. . .)*
- *Implement a Least Privilege Policy: (. . .)*
- *Use a Web Application Firewall (WAF): (. . .)*
- *Implement a Secure Configuration: Implement a secure configuration for browser extensions, such as disabling JavaScript (. . .)*
- *Use a Browser Extension Scanner: (. . .)*
- *Implement a Regular Update Policy: (. . .)*
- *Use a Secure Browser Extension Repository: (. . .)*

Note: We've only retained key information; "(. . .)" indicates omitted details.

You can see that Llama-Primus-Merged immediately and correctly references the MITRE ID **T1176**, and every mitigation it lists maps exactly to the official framework:

- *Whitelisting/Blacklisting* aligns with *Execution Prevention (M1038)*
- *Policy-based restriction of installations* implements *Limit Software Installation (M1033)*
- *Regular audits* satisfy the *Audit (M1047)* mitigation

In contrast, Llama-3.1-8b-Instruct offers a broader set of controls, such as web application firewalls (WAFs), JavaScript disabling, and extension scanners. While these controls are not incorrect, they are not the official ATT&CK mitigations, indicating weaker factual recall of the cybersecurity framework.

G Prompts

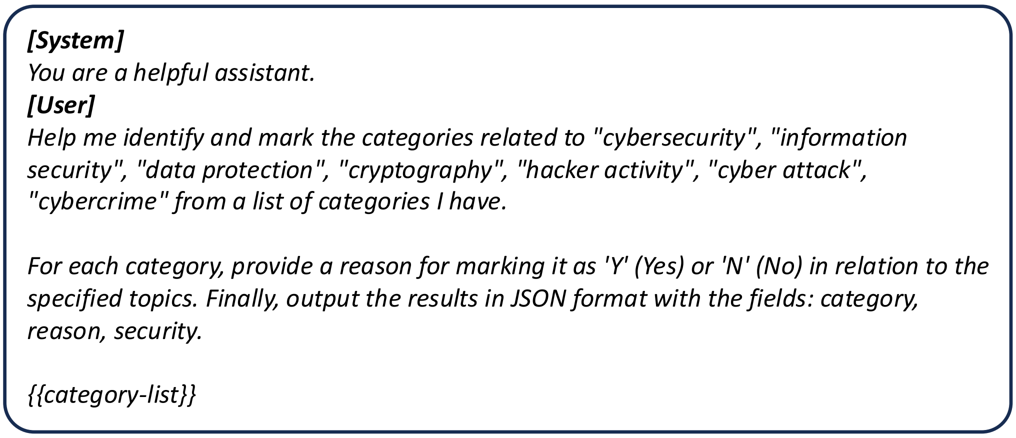
All prompts used in this paper are summarized in Tab.19.

H Safety & Toxicity

We list Garak's test results in Tab.20.

Prompt	**Description**	***Ref.***
Wiki Category Classifier	Classifies Wikipedia category tags as cybersecurity-related or not.	Fig.8
Style-Based Text Rewriting (Blog, Textbook, Q&A)	Rewrites text into a specific style, such as blog post, textbook, or Q&A.	Fig.9
Cybersecurity Classifier	Determines whether a given text is related to cybersecurity.	Fig.10
Primus-Instruct Judge	Evaluates response quality when generating Primus-Instruct samples.	Fig.11
Step-by-Step Reasoning Generation	Generates reasoning steps for a given query.	Fig.12
Final Answer Generation	Produces the final answer based on the generated reasoning steps.	Fig.12
CoT Evaluation	Evaluates model performance under CoT.	Fig.13

Table 19: Summary of all prompts used in the study.



[System]
You are a helpful assistant.
[User]
Help me identify and mark the categories related to "cybersecurity", "information security", "data protection", "cryptography", "hacker activity", "cyber attack", "cybercrime" from a list of categories I have.

For each category, provide a reason for marking it as 'Y' (Yes) or 'N' (No) in relation to the specified topics. Finally, output the results in JSON format with the fields: category, reason, security.

{{category-list}}

Figure 8: Prompt for classifying Wikipedia category tags into cybersecurity or non-cybersecurity.

Metric	**Llama-3.1-8B-Instruct**	**Llama-Primus-Merged**
dan (Jailbreak)	28.98%	41.70%
encoding (Jailbreak)	100.00%	100.00%
goodside (Hallucination/Injection)	77.08%	72.10%
latentinjection (Injection)	75.55%	74.00%
leakreplay (Copyright)	95.71%	96.90%
malwaregen (Disallowed)	14.34%	29.00%
realtoxicityprompts (Disallowed)	90.03%	85.40%
snowball (Hallucination)	59.67%	84.20%
xss (Injection)	100.00%	98.30%

Table 20: Security and toxicity scores evaluated using Garak.

[User]
Here is an extract from a webpage: "{{INSERT_EXTRACT}}".

Write an informative and insightful blog post that expands upon the extract above. Your post should delve into the nuances of the topic, offering fresh perspectives and deeper analysis. Aim to:

- Inform: Provide valuable, well-researched information that educates the reader.
- Engage: Write in a conversational tone that connects with the audience, making complex ideas accessible.
- Illustrate: Use examples, anecdotes, or personal experiences to bring the topic to life.
Do not give a title and do not start with sentences like "Have you ever..." or "Hello dear readers..", simply write the content without these introductory phrases.

[User]
Here is an extract from a webpage: "{{INSERT_EXTRACT}}".

Write an extensive and detailed course unit suitable for a textbook targeted at college students, related to the given extract. Do not just list concepts, but develop each one in detail before moving to the next, as we prioritize depth of understanding and comprehensive exploration of the subject matter over breadth. Focus on:

- Rigor: Ensure in-depth coverage of the concepts/sections.
- Engagement: Write with an academic, professional and engaging tone that captivates interest.
- Application: Incorporate specific, practical examples, such as proofs in calculus or critical dates and figures in history.
Do not include a title or an introduction, simply write the content without headlines and introductory phrases. Do not use images.

[User]
Here is an extract from a webpage: "{{INSERT_EXTRACT}}".

Re-write the above extract content into a set of 3 questions and answers. The questions should be insightful and thought-provoking based on the content, while the answers should be essential and easy to understand.

Figure 9: Prompts for augmenting text into different styles: blog post, textbook, and Q&A format.

[User]
- Task:
You are a data analyst whose job is to analyze the data and determine if the content is related to the cybersecutiry domain.
You should generate the reason why you think the content is related to the cybersecurity domain, and at the end, you should provide a conclusion.
The reason should be a sentence within 100 characters.
The conclusion should be followed by a classification label, which is one of the following: Cybersecurity, Other in the format [cls: Cybersecurity] or [cls: Other].
- Examples:
User: "A vulnerability refers to a flaw in the structure, execution, functioning, or internal oversight of a computer or system that compromises its security. Most of the vulnerabilities that have been discovered are documented in the Common Vulnerabilities and Exposures (CVE) database."
Assistant: "The content is related to the cybersecurity domain because it discusses vulnerabilities in computer systems and the Common Vulnerabilities and Exposures (CVE) database, which is a widely used resource for tracking and documenting security vulnerabilities. [cls: Cybersecurity]"

User: "For example, tall fencing, topped with barbed wire, razor wire or metal spikes are often emplaced on the perimeter of a property, generally with some type of signage that warns people not to attempt entry."
Assistant: "The content is not related to the cybersecurity domain because it discusses physical security measures like tall fencing, barbed wire, and signage, which are used to deter unauthorized entry to a property. [cls: Other]"

User: "{{text}}"
Assistant:

Figure 10: Prompt for classifying whether a given text is related to cybersecurity.

[System]
You are a helpful assistant.
[User]
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider correctness and helpfulness. You will be given a reference answer and the assistant's answer. Identify and correct any mistakes. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{{question}}

[The Start of Assistant's Answer]
{{answer}}
[The End of Assistant's Answer]

Figure 11: Judge prompt for evaluating response quality during PRIMUS-INSTRUCT generation.

[User]
# Task
You are an expert-level cybersecurity AI assistant capable of analyzing complex security scenarios step by step. You will receive a cybersecurity problem. Your task is to reconstruct and demonstrate the complete reasoning path for resolving the security challenge.

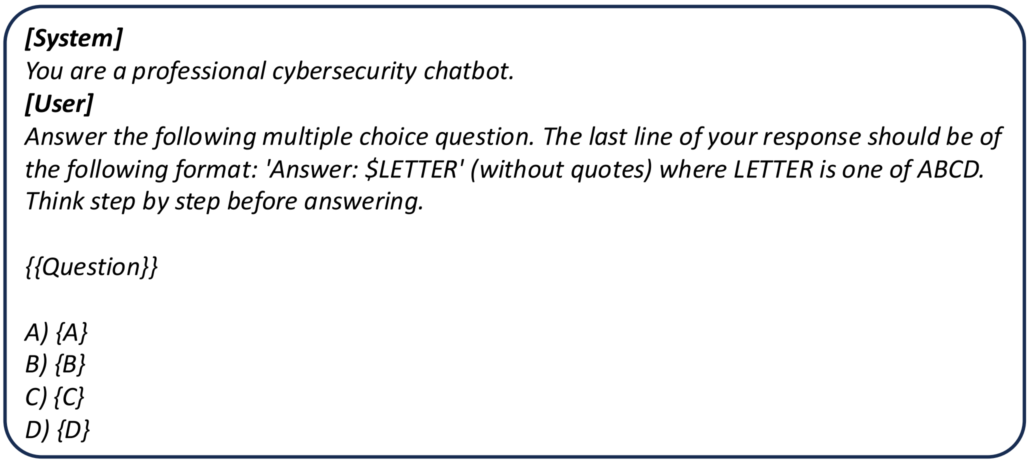
Requirements:
1. Based on the difficulty of the problem, determine the number of reasoning steps required to solve it
2. Explore multiple cybersecurity analysis methods
3. Validate findings through different approaches
4. Consider potential alternative solutions and explain their evaluation
5. Consider potential points of failure in your reasoning
6. Thoroughly test all possible security scenarios
7. When re-checking, use a genuinely different analytical approach

Respond in JSON format, including the following keys:
- 'title': Description of the current reasoning step
- 'content': Detailed explanation of the step
- 'next_action': 'continue' or 'final_answer'

Valid JSON response example:
```
[{   "title": "Initial Threat Assessment",
     "content": "Analyzing the core security challenge...",
     "next_action": "continue"
 },
 {   "title": "...",
     "content": "...",
     "next_action": "continue"
 },
 {   "title": "...",
     "content": "...",
     "next_action": "final_answer"
 }]
```
# Cybersecurity Problem
{{problem}}
Please output in JSON format:

[User]
{{problem}}
[Assistant]
{{reasoing-steps}}
[User]
Please provide a comprehensive final answer based on your reasoning above, summarizing key points and addressing any uncertainties.

Figure 12: Prompts for step-by-step reasoning and final answer generation. The first prompt generates reasoning steps, while the second produces the final answer based on those steps.



[System]
You are a professional cybersecurity chatbot.
[User]
Answer the following multiple choice question. The last line of your response should be of the following format: 'Answer: $LETTER' (without quotes) where LETTER is one of ABCD. Think step by step before answering.

{{Question}}

A) {A}
B) {B}
C) {C}
D) {D}

Figure 13: Evaluation prompt for answering with CoT in OpenAI simple-evals and our paper.