Llama-3.1-FoundationAI-SecurityLLM-Base-8B Technical Report

Paul Kassianik[1]*, Baturay Saglam[1,2], Alexander Chen[1], Blaine Nelson[1], Anu Vellore[1], Massimo Aufiero[1], Fraser Burch[1], Dhruv Kedia[1], Avi Zohary[1], Sajana Weerawardhena[1], Aman Priyanshu[1], Adam Swanda[1], Amy Chang[1], Hyrum Anderson[1], Kojin Oshiba[1], Omar Santos[3], Yaron Singer[1], Amin Karbasi[1]

[1]Foundation AI – Cisco Systems Inc.
[2]Yale University
[3]Security & Trust Organization – Cisco Systems Inc.

Abstract

As transformer-based large language models (LLMs) increasingly permeate society, they have revolutionized domains such as software engineering, creative writing, and digital arts. However, their adoption in cybersecurity remains limited due to challenges like scarcity of specialized training data and complexity of representing cybersecurity-specific knowledge. To address these gaps, we present Foundation-Sec-8B, a cybersecurity-focused LLM built on the Llama 3.1 architecture and enhanced through continued pretraining on a carefully curated cybersecurity corpus. We evaluate Foundation-Sec-8B across both established and new cybersecurity benchmarks, showing that it matches Llama 3.1-70B and GPT-4o-mini in certain cybersecurity-specific tasks. By releasing our model to the public, we aim to accelerate progress and adoption of AI-driven tools in both public and private cybersecurity contexts.

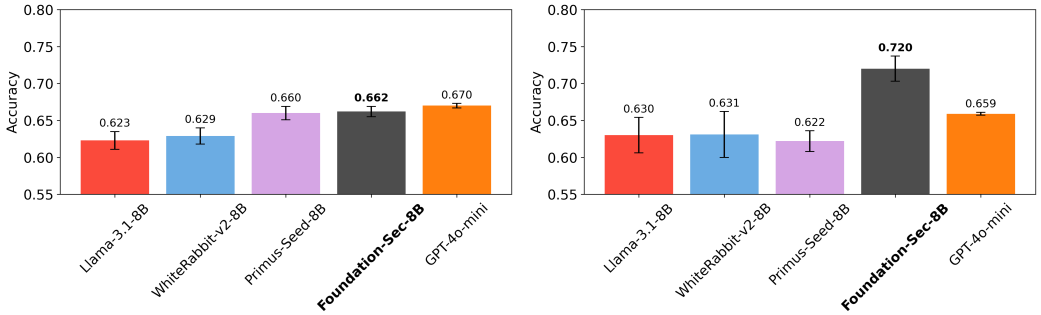
1 Introduction

Artificial intelligence tools have rapidly become essential for boosting productivity and automating tasks across various domains. Frontier large language models (LLMs) such as ChatGPT [43] have accelerated this trend, enabling users to complete in minutes tasks that once took hours or days. Since the release of ChatGPT, AI capabilities have advanced significantly [8, 50, 66]. The rise of highly capable open-source LLMs [22, 62] has further democratized access, allowing researchers to adapt these models through fine-tuning and continued pretraining [20, 37]. These adaptations often yield models that match or exceed proprietary counterparts on specialized tasks [13, 68].

Despite these advancements, the integration of LLMs into standard cybersecurity practices remains limited. Cybersecurity professionals face several barriers to adopting frontier LLMs, including restrictive safety guardrails from commercial providers, a lack of clean and publicly available cybersecurity datasets [2, 40], model hallucinations [29, 61], and challenges from distribution shifts [24, 52]. Meanwhile, serious threat actors have the compute and data resources to train custom, closed-source LLMs for nefarious purposes. The broad and heterogeneous nature of cybersecurity – from phishing detection to cryptographic analysis—makes developing a general-purpose security AI particularly difficult. As a result, the field is dominated by fragmented, task-specific tools [18].

In this work, we introduce **Foundation-Sec-8B**, a cybersecurity-specialized LLM built on Llama 3.1-8B [22]. Our model shows significant performance gains over Llama 3.1-8B across cybersecurity benchmarks. We also release trained checkpoints to the research community, supported by

[*]Correspondence to: Paul Kassianik (paulkass@cisco.com) and Dhruv Kedia (dkedia@cisco.com).



(a) CTIBench-MCQA

(b) CTIBench-RCM

Figure 1: **Overview of core results on the selected cybersecurity benchmarks.** Foundation-Sec-8B shows significant improvement over Llama 3.1-8B while matching or surpassing GPT-4o-mini in cyber threat intelligence knowledge.

comprehensive evaluations that highlight the model's capabilities[2]. With this contribution, we advance research on applying LLMs to cybersecurity tasks and level the playing field for defensive cybersecurity practitioners.

2 Related Works

2.1 Continued Pretraining

Continued pretraining was initially proposed for transformer-based models as a way to improve domain-specific adaptation [11, 23, 25, 27, 35, 44, 68]. Wu et al. [68] showed that it enhances zero-shot and few-shot promptability, while Gururangan et al. [25] demonstrated that task-specific pretraining, even on unlabeled data, can boost model performance.

This approach has also been extended to LLMs, where models are continuously pretrained on billions of tokens to improve performance in specific domains such as medicine [13, 73], law [15, 16], mathematics [3, 53], and code [49]. Ibrahim et al. [28], Parmar et al. [44] further provide guidelines on when domain-specific continued pretraining is most effective.

2.2 Benchmarks

We reviewed a range of benchmarks and evaluation techniques at the intersection of LLMs and cybersecurity. Since our primary objective is to assess the knowledge of pretrained models, we exclude tasks that emphasize behavioral robustness over factual understanding.

We have selected three cybersecurity benchmarks framed as multi-class classification tasks. This design choice was motivated by the observation that pretrained models often fail to reliably follow instructions, rendering open-ended formats such as short answer questions (SAQ) unsuitable. We therefore focus on two formats: *multiple choice question answering* (MCQA) and *root cause mapping* (RCM). Each MCQA question includes four options with a single correct ground truth answer. For the RCM task, each Common Weakness Enumeration (CWE) [14] description maps to exactly one CWE ID.

2.2.1 Security Benchmarks

CTIBench Cyber threat intelligence (CTI) plays a key role in understanding and mitigating evolving cyber threats. CTIBench [1] targets practical, CTI-specific tasks and consists of five sections. However, most require advanced, consistent instruction-following capabilities and are not suitable for evaluation with a pretrained model. Therefore, we adapt two of them for base model evaluation: MCQA and RCM tasks. MCQA includes 2,500 questions drawn from CTI frameworks such as NIST [32], the Diamond Model of Intrusion Detection [9]; regulations like GDPR [17]; CTI sharing

[2] `https://huggingface.co/fdtn-ai/Foundation-Sec-8B`

standards such as STIX and TAXII [41]; and taxonomies like the MITRE ATT&CK Framework [56] and CAPEC [10]. The RCM section evaluates a model's ability to identify the root cause of a vulnerability by linking CVE (Common Vulnerability Enumeration) [58] records and bug reports to CWE (Common Weakness Enumeration) entries [14]. This task is highly nuanced, requiring a deep understanding of both CVE descriptions and the CWE taxonomy.

CyberMetric The CyberMetric dataset [59] was built from a collection of documents—such as NIST standards, research papers, public books, RFCs, and other cybersecurity publications—and converted into MCQA format using GPT-3.5 with Retrieval-Augmented Generation (RAG). Human experts spent over 200 hours validating the questions and answers to ensure accuracy, relevance, and topic alignment. CyberMetric is available in four sizes: 80, 500, 2000, and 10,000 samples. We use the 500-sample version, as the larger sets contain only ChatGPT-generated questions on similar topics, without additional human validation.

SecBench SecBench [31] includes both MCQA and SAQ questions, designed to assess LLM knowledge across two dimensions: Knowledge Retention and Logical Reasoning. A key strength of the benchmark is that a large portion of its content was created by human experts through a Cybersecurity Question Design Contest, making it more challenging than benchmarks like MMLU and CyberMetric. It is also the most recent in our collection, released in December 2024. Built using both contest submissions and open-source resources, SecBench includes English and Chinese sections. Since we are developing a monolingual model, we use only the English portion which contains 600 samples.

2.2.2 Non-Security Benchmarks

MMLU To evaluate whether continued pretraining on cybersecurity data affects the model's general knowledge, we evaluate our model on the full Measuring Massive Multitask Language Understanding (MMLU) benchmark [26]. MMLU covers a broad range of topics, including STEM, humanities, and social sciences, and serves as a strong indicator of a model's retained knowledge across diverse domains. This allows us to check for any signs of overfitting or catastrophic forgetting, ensuring that domain specialization does not come at the cost of severe degradations to general reasoning and factual recall.

2.3 LLMs for Security

Several works [70, 75] have reviewed the use of LLMs in cybersecurity. For a broader overview, we refer the reader to those studies and focus here on prior efforts most relevant to our work. While much research has focused on secure code generation [45, 54, 74], our model is not designed or optimized for coding tasks. Instead, our primary goal was to integrate core cybersecurity knowledge into a pretrained language model, making it a strong foundation for downstream use cases (see Section 6). This enables broader applicability across cybersecurity tasks beyond code generation.

Prior to our work, several notable models were trained on general cybersecurity data. We include them as baselines in our evaluations, with the exception of the final model described below.

WhiteRabbitNeo-V2 WhiteRabbitNeo [67] introduced one of the earliest open-source cybersecurity LLMs, based on the Llama 3.1 family. Released in 8B and 70B variants, it focused on building uncensored models tuned for offensive security tasks.

Primus Yu et al. [72] curated a cybersecurity corpus from sources like MITRE, Wikipedia, cybersecurity companies, and manually collected cyber threat intelligence. Their dataset, containing nearly 2 billion tokens, was used to pretrain and instruct-finetune models based on Llama 3.1-8B.

SecurityLLM SecurityLLM[3], though not described in a technical report, is an open-source model based on Hugging Face's Zephyr series [63] and built on Mistral 7B [30]. It was designed as a helpful security-focused assistant and finetuned across 30 domains, including attack surface threats, cloud security, and compliance frameworks like CIS Controls.

[3] https://huggingface.co/ZySec-AI/SecurityLLM

SecGemini SecGemini [51] introduced the only frontier LLM specifically tailored for cybersecurity. SecGemini is a reasoning-enhanced [42, 55] variant of Gemini Flash [21], augmented with live threat intelligence access with a knowledge-based system. As of this writing, it remains in closed preview and is not publicly available.

3 Data Collection and Preparation

One of the key challenges in developing LLMs for cybersecurity is the need for large volumes of high-quality data to drive meaningful downstream improvements. Further pretraining in other domains often relies on tens or hundreds of billions of tokens to effectively infuse domain-specific knowledge [71]. In contrast, current cybersecurity models are typically trained on fewer than 2 billion tokens [67, 72]. Since cybersecurity is a relatively niche topic in the broader landscape of internet content, we developed a two-pronged approach to ensure diverse and high-quality data collection. First, we deployed general-purpose scrapers with a relevancy filter for broad data coverage. Second, we built custom scrapers targeting known high-quality cybersecurity sources and sites that are less accessible to URL-based scrapers.

We chose to build our dataset from scratch rather than filtering existing web-scale datasets like Hugging Face's FineWeb [46]. Security-related content often appears "non-English" due to the presence of code, abbreviations, and fragmented sentences—leading to high perplexity. Datasets such as FineWeb [46] and The Pile [19] apply a definition of "quality" that does not align well with cybersecurity needs. For instance, FineWeb uses a fastText-based English filter with a threshold of 0.65 [33], which we found unsuitable for cybersecurity texts. This setting would exclude valuable content like CVE descriptions (see Appendix A.1). While existing datasets are valuable in their own contexts, we designed a focused data collection and curation pipeline—outlined in Figure 2—tailored to the unique demands of cybersecurity.

3.1 Data Collection via Wide-Net Scraper

To expand our collection of cybersecurity data from sources that are harder to identify or navigate, we deployed a wide-net internet crawler to capture relevant content. The crawler begins by visiting a set of initial seed URLs, fetching and storing the raw page content, then parsing each response to extract new links, which are added to a processing queue. We seeded the crawler with URLs from a diverse set of cybersecurity domains.

The crawler prioritized URLs based on their depth in the search tree, favoring exploration within a domain before moving on to a new one. This produced a hybrid strategy—breadth-first search within a domain and depth-first search across domains.

To keep the crawl focused, we pruned irrelevant branches by applying a relevancy filter to the content of each scraped page. Links were only extracted if the page content passed this filter. Details of the relevancy filter are provided in Section 3.2.2.

3.2 Data Preprocessing

While the top-of-funnel collection (Section 3.1) gathered over 4 TiB of raw data, we built a scalable preprocessing pipeline to filter, clean, and transform the raw data into a high-quality, standardized dataset suitable for pretraining. We leveraged existing primitives to construct our own composable data pipelines, which were deployed across a cluster of virtual machines.

After running through this pipeline, the original 4 TiB of raw data was reduced to just under 25 GiB of cleaned text, yielding approximately 0.6%. This low yield reflects both the extraction of only text content from noisy HTML pages and the removal of low-quality or irrelevant files. An analysis of the pipeline stages shows that about 90% of the files were filtered out.

3.2.1 Text Extraction

Text extraction is the most computationally intensive and time-consuming part of the pipeline. This stage involved converting files of varying formats and sizes into clean Markdown files. We evaluated a range of closed and open-source tools on a sample dataset and developed a simple strategy to select the best tool for each file based on its format and size.

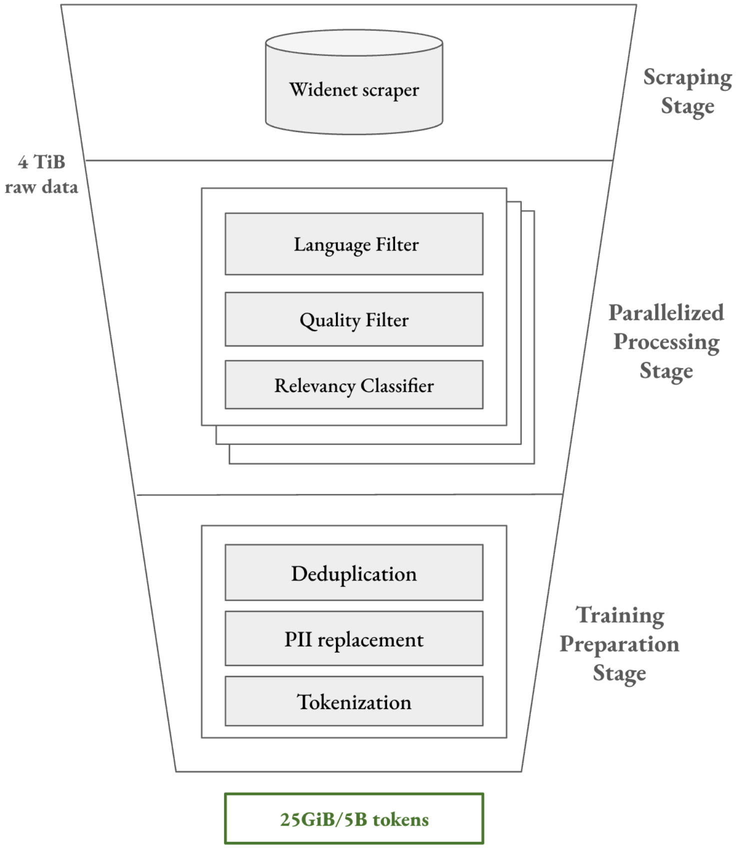


Figure 2: **Three-stage data collection and processing pipeline:** (1) **Scraping Stage**—a widenet scraper gathers 4 TiB of raw web content; (2) **Parallelized Processing Stage**—language filtering, quality filtering, and relevancy classification prune the data; (3) **Training Preparation Stage**—deduplication, PII replacement, and tokenization produce 25 GiB (about 5 billion tokens) of final training data.

3.2.2 Relevancy Filtering

To ensure that our dataset consisted exclusively of cybersecurity-related documents, we implemented a relevance filtering pipeline. The initial filtering stage employed a keyword matching mechanism utilizing a curated list of approximately 800 cybersecurity-related terms and acronyms. The presence of any keyword within a source document was treated as an indicator of relevance. While this approach offered computational efficiency and high recall, it suffered from a significant false positive rate.

To address the limitations of the keyword filter, we developed a small transformer-based classifier [64]. We constructed a labeled dataset comprising 26,000 documents. Labels were generated using Gemini 2.0 Flash-Lite [21] prompted with an instruction to label each document as related/unrelated to cybersecurity. The resulting dataset was balanced, containing 13,000 positive (relevant) and 13,000 negative (irrelevant) samples, each under 1024 tokens long.

We evaluated both the keyword-matching filter and the finetuned classifier on a held-out test set and found that the finetuned classifier significantly improved upon the keyword-matching filter, obtaining an F1 score of 0.924 (see Figure 3).

3.2.3 Filter Evaluation Experiments

In addition to our relevancy filter, we implemented additional filters to further remove low-quality data. In particular, we used a language filter and simple regex-based heuristics to identify files that were deficient or degenerate.

We conducted a series of experiments to assess the effectiveness of these filters within our data processing pipeline. In each experiment, we evaluated how well the filters retained high-quality, cybersecurity-relevant documents.

To carry out this analysis, we randomly sampled 1,000 documents from our raw dataset. Each document was then assessed for quality and topic relevance using a frontier LLM, prompted to provide binary labels based on two criteria: (1) textual quality and (2) relevance to cybersecurity. A document was considered acceptable for inclusion only if it satisfied both criteria; otherwise, it should be filtered out.

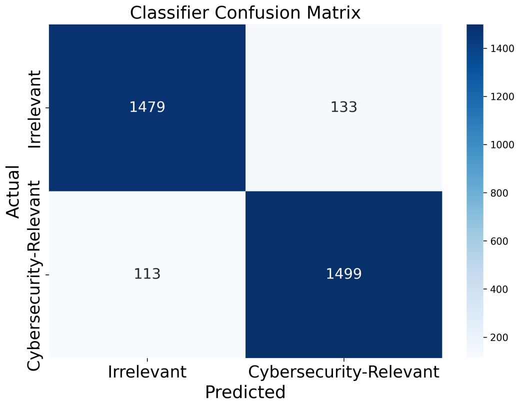


Figure 3: Relevancy classifier on evaluation set

We then applied the different filters independently to this labeled subset and compared their outputs against the ground truth. Some documents from key sources, including pages from the CVE and MITRE ATT&CK databases, were excluded by these filters (see Appendix A.2). Based on these evaluations, we ultimately chose to exclude most of the heuristic-based filters from our final data processing pipeline.

3.3 Data Preparation

Before training, we perform additional steps to clean and format the data for optimal use. To deduplicate our corpus, we applied an n-gram Bloom filter method [7] at the paragraph level. We also followed best practices to filter out personally identifiable information (PII) and prevent training on private data. Following prior work [6, 22, 25, 69], we upsampled known high-quality data related to Tactics, Techniques, and Procedures (TTPs). Finally, we split the resulting 5.1 billion tokens into a 99% training set and a 1% test set.

4 Training and Evaluation

We trained a Llama 3.1-8B model on the curated dataset through a further pretraining approach. Documents were packed into sequences of 4096 tokens for maximum efficiency [47]. Training was performed using DeepSpeed [48] on a multi-node compute cluster. We used the AdamW optimizer [38] with a cosine decay learning rate schedule [39].

Benchmarks We consider the benchmarks detailed in Section 2.2.1 in our experiments. In Appendix B.2, we also discuss other cybersecurity benchmarks from the literature that we reviewed but chose not to include in our evaluation, as they were misaligned with our evaluation objectives.

Baselines The models reviewed in Section 2.3 are included as baselines for comparison. For further details, we refer readers to their respective model cards on Hugging Face.

Input Formatting and Prompting To ensure fair evaluation, we carefully design the prompting strategy, as poor formatting can obscure a model's knowledge and capability. Since log-probabilities are unavailable for closed models, we evaluate based on predicted tokens. Pretrained models, which do not follow instructions directly, are evaluated using 5-shot prompts to help infer both the task and the expected output format. Instruct-finetuned (IFT) models, on the other hand, are evaluated in a zero-shot setting to avoid inconsistent or verbose completions. We prompt them only with the question and extract their answers using regex (regular expressions). Additional implementation details, including templates and regex patterns, are provided in Appendices B.3 and B.4.

Model	CTIBench-MCQA	CTIBench-RCM	CyberMetric-500	SecBench
SecurityLLM	0.601±0.009	N/A[†]	0.807±0.010	0.668±0.012
Llama 3.1-8B	0.623±0.012	0.630±0.024	0.848±0.008	0.735±0.011
WhiteRabbitNeo-V2-8B	0.629±0.011	0.631±0.031	0.846±0.006	0.738±0.010
Primus-Seed-8B	0.660±0.009	0.622±0.014	0.853±0.007	0.732±0.009
Llama 3.1-70B	**0.678±0.008**	0.709±0.010	**0.918±0.008**	0.832±0.008
WhiteRabbitNeo-V2-70B	**0.680±0.007**	0.711±0.008	**0.917±0.009**	**0.836±0.008**
GPT-4o-mini	0.670±0.003	0.659±0.002	0.889±0.002	0.800±0.002
SecGemini	0.8630	0.8610	N/A[‡]	N/A[‡]
Foundation-Sec-8B	0.662±0.007 (↑6.26%)	**0.720±0.017** (↑14.29%)	0.848±0.009 (↔0.0%)	0.723±0.009 (↓1.63%)

[†] Due to its limited context length, SecurityLLM did not produce meaningful results on CTIBench-RCM.
[‡] Results are not reported by the developer.

Table 1: Performance of the models on the selected cybersecurity benchmarks (temperature 0.3). Reported performance differences are relative to the Llama 3.1-8B model, which Foundation-Sec is based on.

Model	CTIBench-MCQA	CTIBench-RCM	CyberMetric-500	SecBench
Llama 3.1-8B	0.641	0.636	0.849	0.744
Llama 3.1-70B	0.689	0.710	0.924	0.839
Foundation-Sec-8B	0.676 (↑5.46%)	0.724 (↑13.84%)	0.851 (↑0.24%)	0.734 (↓1.34%)

Table 2: Performance of the models on the selected cybersecurity benchmarks (temperature 0). Reported performance differences are relative to the Llama 3.1-8B model, which Foundation-Sec is based on.

5 Results

Evaluations were conducted over 10 trials to account for stochasticity in token sampling (with temperature set to 0.3) and in the selection of few-shot examples. We report mean accuracy with the standard deviation over 10 trials. Results are shown in Table 1. We also evaluate the models at temperature 0 and present those results in Table 2.

5.1 Security Benchmark Performance

Foundation-Sec-8B achieves **state-of-the-art performance** for its size class on the selected CTI tasks, performing **on par with GPT-4o-mini**. It trails GPT-4o-mini by only 0.8 points on CTIBench-MCQA while outperforming it by 6.1 points on CTIBench-RCM. Foundation-Sec-8B also surpasses both Llama 3.1-70B and WhiteRabbitNeo-V2-70B by about 1 point on CTIBench-RCM, while falling short by less than 2 points on CTIBench-MCQA.

The model achieves a notable improvement of over 3 points in accuracy compared to its parent model, Llama 3.1-8B-base. We emphasize that, since we evaluate general knowledge rather than a task-specific fine-tuning setup, these gains are substantially large for improvements over a general-purpose model on a specialized, nuanced domain of knowledge—a trend also observed in prior work [13, 67, 72].

Overall, our results position Foundation-Sec-8B as the leading 8-billion parameter model for CTI security tasks.

5.2 General Benchmark Performance

Prior work [12, 57] shows that continued pretraining is typically followed by evaluations on general benchmarks to check for catastrophic forgetting. Following this practice, we verify that our training does not significantly degrade general performance. We evaluate our model on the full MMLU benchmark. Foundation-Sec-8B achieves a score of 0.593±0.004, compared to 0.617±0.004 for Llama 3.1-8B. The observed 2.4 point drop is consistent with prior literature.

6 Use Cases

Large language models in cybersecurity are no longer just theoretical—they are increasingly being integrated into high-value, operational workflows across the security lifecycle. Foundation-Sec was developed with these real-world applications in mind. We outline three core areas where Foundation-Sec is **currently being piloted or deployed**, either in its pretrained form or after fine-tuning.

SOC Acceleration Security Operations Centers (SOCs) face a constant stream of alerts that require triage, enrichment, and contextualization. Foundation-Sec is currently being piloted to automate key parts of this workflow. When finetuned or prompted effectively, it is used to:

- Summarize multi-source alerts into human-readable case notes
- Generate incident timelines and identify relevant entities
- Draft analyst-style reports to support incident resolution and handoffs

These capabilities reduce time-to-triage and enable analysts to handle more alerts with higher accuracy.

Proactive Threat Defense Beyond reactive workflows, Foundation-Sec is being deployed to model and simulate attacker behavior. This includes:

- Extracting Tactics, Techniques, and Procedures (TTPs) from threat intelligence reports
- Prioritizing vulnerabilities based on contextual impact and exploitability
- Generating attack path hypotheses from asset and configuration data
- Drafting penetration test reports with vulnerability details and remediation steps

A particularly effective application has been fine-tuning Foundation-Sec for MITRE ATT&CK Technique extraction from unstructured threat reports. In internal evaluations, Foundation-Sec outperformed a similarly sized non-security-tuned model (Llama 3.1-8B) by over 10% on this classification task, highlighting the value of security-domain pretraining.

Engineering Enablement Security engineering and platform teams often face the challenge of enforcing security standards across rapidly evolving development environments. Foundation-Sec is being used to streamline and enhance these workflows by providing secure development guidance, validating configurations, and supporting compliance efforts.

When applied effectively, Foundation-Sec helps teams:

- Interpret and apply security policies during development and deployment
- Validate configuration files and infrastructure setups against best practices
- Assess whether submitted evidence meets compliance control requirements
- Analyze security policies for inconsistencies and outdated controls

These tasks typically require a combination of context awareness and domain-specific knowledge—areas where Foundation-Sec excels.

If you are interested in using Foundation-Sec for your own cybersecurity applications or research, please reach out to **Paul Kassianik** (`paulkass@cisco.com`) or **Dhruv Kedia** (`dkedia@cisco.com`) at Foundation AI.

7 Conclusion and Future Work

In this work, we introduced Foundation-Sec, a cybersecurity-specialized large language model built upon Llama 3.1. Addressing key limitations that have hindered LLM adoption in cybersecurity, we curated a high-quality cybersecurity dataset and demonstrated significant improvements in task-specific performance. Our evaluations show that Foundation-Sec achieves capabilities competitive with much larger models, such as Llama 3.1-70B, without compromising general-purpose functionality.

We highlight several promising directions for further research and development:

- Scale up Foundation-Sec by increasing its parameter count and expanding the training corpus.
- Extend Foundation-Sec to handle cybersecurity-related coding tasks.
- Integrate Foundation-Sec into tool-calling and agentic systems for more interactive applications.

By releasing Foundation-Sec publicly, we aim to support both the cybersecurity and AI research communities, fostering broader experimentation, advancement, and practical deployment of AI-driven security tools. We envision Foundation-Sec accelerating LLM adoption among cybersecurity professionals and enabling new lines of security research.

We also hope this work inspires future studies on optimizing specialized pretraining techniques, expanding coverage of cybersecurity domains, and investigating how smaller models can rival or surpass larger general-purpose LLMs in domain-specific tasks. Ultimately, Foundation-Sec highlights the impact of targeted, domain-aware training in making LLMs more effective for secure, intelligent cybersecurity operations.

Acknowledgements

We are grateful to Zane Mogannam, Divit Rawal, Erica Liu, Arjun Banerjee, Neel Kolhe, Alena Subach, and Kathryn Sun from UC Berkeley Launchpad for their help with data collection and processing.

We thank Sasha Sinkevich and Jon McLachlan from YSecurity for their early efforts in kick-starting this project.

We also sincerely thank Cisco's Security & Trust Organization (S&TO) for their ongoing partnership in identifying and validating impactful use cases for this model. In particular, we acknowledge Omar Santos, Robert Kerby, Vinay Bansal, Aaron Carter, Chalamaiah Gupta Reddy, and Sam Cosentino, whose early input shaped our understanding of real-world security workflows and informed our development strategy.

Finally, we thank Kamile Lukosiute for her valuable feedback during the preparation of this report.

References

[1] Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL `https://openreview.net/forum?id=iJAOpsXo2I`.

[2] Dure Adan Ammara, Jianguo Ding, and Kurt Tutschku. Synthetic network traffic data generation: A comparative study. *arXiv [cs.CR]*, October 2024.

[3] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

[4] Manish Bhatt, Sahana Chennabasappa, Cyrus Nikolaidis, Shengye Wan, Ivan Evtimov, Dominik Gabi, Daniel Song, Faizan Ahmad, Cornelius Aschermann, Lorenzo Fontana, Sasha Frolov, Ravi Prakash Giri, Dhaval Kapil, Yiannis Kozyrakis, David LeBlanc, James Milazzo, Aleksandar Straumann, Gabriel Synnaeve, Varun Vontimitta, Spencer Whitman, and Joshua Saxe. Purple llama cyberseceval: A secure coding benchmark for language models, 2023. URL `https://arxiv.org/abs/2312.04724`.

[5] Dipkamal Bhusal, Md Tanvirul Alam, Le Nguyen, Ashim Mahara, Zachary Lightcap, Rodney Frazier, Romy Fieblinger, Grace Long Torales, Benjamin A. Blakely, and Nidhi Rastogi. SECURE: Benchmarking Large Language Models for Cybersecurity . In *2024 Annual Computer Security Applications Conference (ACSAC)*, pages 15–30, Los Alamitos, CA, USA, December 2024. IEEE Computer Society. doi: 10.1109/ACSAC63791.2024.00019. URL `https://doi.ieeecomputersociety.org/10.1109/ACSAC63791.2024.00019`.

[6] Cody Blakeney, Mansheej Paul, Brett W Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training. *arXiv preprint arXiv:2406.03476*, 2024.

[7] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[8] 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. Language models are few-shot learners. *arXiv [cs.CL]*, May 2020.

[9] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. The diamond model of intrusion analysis. Technical Report 298-0704, ThreatConnect, 2013. URL `https://www.threatconnect.com/wp-content/uploads/The-Diamond-Model-of-Intrusion-Analysis.pdf`.

[10] CAPEC. Common attack pattern enumerations and classifications (capec). `https://capec.mitre.org/`, 2024. Available at `https://capec.mitre.org/`.

[11] Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. IMHO fine-tuning improves claim detection. In *Proceedings of the 2019 Conference of the North*, pages 558–563, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics.

[12] Pin-Er Chen, Da-Chen Lian, Shu-Kai Hsieh, Sieh-Chuen Huang, Hsuan-Lei Shao, Jun-Wei Chiu, Yang-Hsien Lin, Zih-Ching Chen, Eddie TC Huang, Simon See, et al. Continual pretraining is (not) what you need in domain adaption. *arXiv preprint arXiv:2504.13603*, 2025.

[13] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. MEDITRON-70B: Scaling medical pretraining for large language models. *arXiv [cs.CL]*, November 2023.

[14] Steve Christey, J. Kenderdine, J. Mazella, and B. Miles. Common weakness enumeration. Technical report, The MITRE Corporation, 2013. URL `https://cwe.mitre.org/`.

[15] Pierre Colombo, Telmo Pires, Malik Boudiaf, Rui Melo, Dominic Culver, Sofia Morgado, Etienne Malaboeuf, Gabriel Hautreux, Johanne Charpentier, and Michael Desa. SaulLM-54B & SaulLM-141B: Scaling up domain adaptation for the legal domain. *arXiv [cs.CL]*, July 2024.

[16] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F T Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. SaulLM-7B: A pioneering large language model for law. *arXiv [cs.CL]*, March 2024.

[17] European Union. General Data Protection Regulation (GDPR). `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679`, 2024. Available at `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679`.

[18] Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, and Norbert Tihanyi. Generative ai and large language models for cyber security: All insights you need. *arXiv [cs.CR]*, June 2024.

[19] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL `https://arxiv.org/abs/2101.00027`.

[20] Aryo Pradipta Gema, Pasquale Minervini, Luke Daines, Tom Hope, and Beatrice Alex. Parameter-efficient fine-tuning of LLaMA for the clinical domain. *arXiv [cs.CL]*, July 2023.

[21] Google DeepMind. Gemini 2.0 flash-lite. `https://developers.google.com/gemini/2-0-flash-lite`, 2025. Model card retrieved from Google Developers Blog and Google Cloud Vertex AI.

[22] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang,

Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models. *arXiv [cs.AI]*, July 2024.

[23] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthc.*, 3(1):1–23, January 2022.

[24] Ragini Gupta, Shinan Liu, Ruixiao Zhang, Xinyue Hu, Pranav Kommaraju, Xiaoyang Wang, Hadjer Benkraouda, Nick Feamster, and Klara Nahrstedt. Generative active adaptation for drifting and imbalanced network intrusion detection. *arXiv [cs.NI]*, March 2025.

[25] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 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, Stroudsburg, PA, USA, 2020. Association for Computational Linguistics.

[26] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[27] Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv [cs.CL]*, May 2019.

[28] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pretrain large language models. *arXiv [cs.LG]*, March 2024.

[29] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Stroudsburg, PA, USA, December 2023. Association for Computational Linguistics.

[30] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[31] Pengfei Jing, Mengyun Tang, Xiaorong Shi, Xing Zheng, Sen Nie, Shi Wu, Yong Yang, and Xiapu Luo. Secbench: A comprehensive multi-dimensional benchmarking dataset for llms in cybersecurity. *arXiv preprint arXiv:2412.20787*, 2024.

[32] Chris Johnson, Lee Badger, David Waltermire, Julie Snyder, and Clem Skorupka. Guide to cyber threat information sharing. Technical Report 800-150, National Institute of Standards and Technology (NIST), 2016. URL https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-150.pdf.

[33] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification, 2016. URL https://arxiv.org/abs/1607.01759.

[34] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[35] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, February 2020.

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

[37] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. StarCoder: may the source be with you! *arXiv [cs.CL]*, May 2023.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[39] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1608.03983.

[40] Innocent Mbona and Jan H. P. Eloff. Data sets for cyber security machine learning models: A methodological approach. In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS)*, pages 149–156. SCITEPRESS, 2024.

[41] OASIS Open. Threat intelligence standards. https://oasis-open.github.io/cti-documentation/, 2024. Available at https://oasis-open.github.io/cti-documentation/.

[42] OpenAI. Openai o1 system card, December 2024. URL https://cdn.openai.com/o1-system-card-20241205.pdf.

[43] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[44] Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv [cs.CL]*, July 2024.

[45] Hammond Pearce, Benjamin Tan, Baleegh Ahmad, Ramesh Karri, and Brendan Dolan-Gavitt. Examining zero-shot vulnerability repair with large language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2339–2356. IEEE, 2023.

[46] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the

web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

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

[48] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, August 2020. ACM.

[49] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[50] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv [cs.CL]*, February 2023.

[51] SecGemini. Google announces sec-gemini v1, a new experimental cybersecurity model. https://security.googleblog.com/2025/04/google-launches-sec-gemini-v1-new.html, 2025. Accessed: 2025-4-17.

[52] Ibrahim Shaer and Abdallah Shami. Thwarting cybersecurity attacks with explainable concept drift. *arXiv [cs.CR]*, March 2024.

[53] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[54] André Silva, Sen Fang, and Martin Monperrus. Repairllama: Efficient representations and fine-tuned adapters for program repair. *arXiv preprint arXiv:2312.15698*, 2023.

[55] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

[56] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. MITRE ATT&CK: Design and Philosophy. Technical report, The MITRE Corporation, 2018. URL https://attack.mitre.org/resources/enterprise-introduction/.

[57] Jianwei Sun, Chaoyang Mei, Linlin Wei, Kaiyu Zheng, Na Liu, Ming Cui, and Tianyi Li. Dial-insight: Fine-tuning large language models with high-quality domain-specific data preventing capability collapse. *arXiv preprint arXiv:2403.09167*, 2024.

[58] The MITRE Corporation. Cve® – common vulnerabilities and exposures program. https://cve.mitre.org/, 2025.

[59] Norbert Tihanyi, Mohamed Amine Ferrag, Ridhi Jain, Tamas Bisztray, and Merouane Debbah. 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, 2024. doi: 10.1109/CSR61664.2024.10679494.

[60] Ryan Tolboom. *Computer Systems Security: Planning for Success*. Open Textbook Collaborative, Newark, NJ, 2022. URL https://open.umn.edu/opentextbooks/textbooks/computer-systems-security-planning-for-success.

[61] S M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv [cs.CL]*, January 2024.

[62] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein,

Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv [cs.CL]*, July 2023.

[63] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[65] Shengye Wan, Cyrus Nikolaidis, Daniel Song, David Molnar, James Crnkovich, Jayson Grace, Manish Bhatt, Sahana Chennabasappa, Spencer Whitman, Stephanie Ding, Vlad Ionescu, Yue Li, and Joshua Saxe. Cyberseceval 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models, 2024. URL `https://arxiv.org/abs/2408.01605`.

[66] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H Chi, F Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *Neural Inf Process Syst*, abs/2201.11903, January 2022.

[67] WhiteRabbitNeo. AI for DevSecOps, WhiteRabbitNeo. `https://www.whiterabbitneo.com/`, 2024. Accessed: 2025-4-17.

[68] Zhaofeng Wu, Robert L Logan, IV, Pete Walsh, Akshita Bhagia, Dirk Groeneveld, Sameer Singh, and Iz Beltagy. Continued pretraining for better zero- and few-shot promptability. *arXiv [cs.CL]*, October 2022.

[69] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

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

[71] Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. September 2024. URL `http://arxiv.org/abs/2409.07431`.

[72] Yao-Ching Yu, Tsun-Han Chiang, Cheng-Wei Tsai, Chien-Ming Huang, and Wen-Kwang Tsao. Primus: A pioneering collection of open-source datasets for cybersecurity LLM training. *arXiv [cs.CR]*, February 2025.

[73] Dong Yuan, Eti Rastogi, Gautam Naik, Sree Prasanna Rajagopal, Sagar Goyal, Fen Zhao, Bharath Chintagunta, and Jeff Ward. A continued pretrained llm approach for automatic medical note generation. *arXiv preprint arXiv:2403.09057*, 2024.

[74] Jie Zhang, Hui Wen, Liting Deng, Mingfeng Xin, Zhi Li, Lun Li, Hongsong Zhu, and Limin Sun. 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. IEEE, November 2023.

[75] Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. When llms meet cybersecurity: A systematic literature review. *Cybersecurity*, 8(1):1–41, 2025.

A Data Processing Details

A.1 Example of CVE Description

Below is an example of a CVE description that scores below 0.65 on the fastText language filter [33] but is considered a high-quality cybersecurity document.

```
Advisory ID: CVE–2016–6335

Summary: No summary provided.

Details:
MediaWiki before 1.23.15, 1.26.x before 1.26.4, and 1.27.x before 1.27.1 does not generate head
items in the context of a given title, which allows remote attackers to obtain sensitive information via
a parse action to api.php.

Published: 2017–04–20T17:59:00Z
Modified: 2024–09–18T02:36:06.797830Z

Affected Packages:
– mediawiki (Debian:11)
   Introduced in: 0
   Fixed in: 1:1.27.1–1
   Source: https://storage.googleapis.com/cve–osv–conversion/osv–output/CVE–2016–6335.json
– mediawiki (Debian:12)
   Introduced in: 0
   Fixed in: 1:1.27.1–1
   Source: https://storage.googleapis.com/cve–osv–conversion/osv–output/CVE–2016–6335.json
– mediawiki (Debian:13)
   Introduced in: 0
   Fixed in: 1:1.27.1–1
   Source: https://storage.googleapis.com/cve–osv–conversion/osv–output/CVE–2016–6335.json
– Unknown package (Unknown ecosystem)
   Introduced in: 0
   Source: https://storage.googleapis.com/cve–osv–conversion/osv–output/CVE–2016–6335.json
 References:
– https://lists.wikimedia.org/pipermail/mediawiki–announce/2016–August/000195.html
– https://phabricator.wikimedia.org/T139570
– https://bugzilla.redhat.com/show_bug.cgi?id=1369613
– https://phabricator.wikimedia.org/T139565
– https://security–tracker.debian.org/tracker/CVE–2016–6335
```

A.2 Example of a MITRE ATT&CK Page

Below is an example of a page from the MITRE ATT&CK database that fails the quality filter of a well-known text filtering system but is considered a high-quality cybersecurity document.

```
Description: Adversaries may gather credentials from the proc filesystem or `/proc`. The proc
filesystem is a pseudo–filesystem used as an interface to kernel data structures for Linux based
systems managing virtual memory. For each process, the `/proc/<PID>/maps` file shows how
memory is mapped within the process's virtual address space. And `/proc/<PID>/mem`, exposed
for debugging purposes, provides access to the process's virtual address space.Huseyin Can YUCEEL
 & Picus Labs. (2022, March 22).baeldung. (2022, April 8). Understanding the Linux /proc/id/maps
File. When executing with root privileges, adversaries can search these memory locations for all
processes on a system that contain patterns indicative of credentials. Adversaries may use regex
patterns, such as <code>grep –E ”^[0–9a–f–]∗ r” /proc/”\$pid”/maps | cut –d' ' –f 1</code>, to
look for fixed strings in memory structures or cached hashes.Atomic Red Team. (2023, November).
T1003.007 – OS Credential Dumping: Proc Filesystem. When running without privileged access,
processes can still view their own virtual memory locations. Some services or programs may save
credentials in clear text inside the process's memory.Gregal, H. (2017, May 12). MimiPenguin.Carlos
 Polop. (2023, March 5). Linux Privilege Escalation. If running as or with the permissions of a web
browser, a process can search the `/maps` & `/mem` locations for common website credential
```

patterns (that can also be used to find adjacent memory within the same structure) in which hashes or cleartext credentials may be located.

Domain: Enterprise Attack

Tactics: Credential Access

Detection: To obtain the passwords and hashes stored in memory, processes must open a maps file in the `/proc` filesystem for the process being analyzed. This file is stored under the path `/proc/PID/maps`, where the `PID` directory is the unique pid of the program being interrogated for such authentication data. The AuditD monitoring tool, which ships stock in many Linux distributions, can be used to watch for hostile processes opening this file in the proc file system, alerting on the pid, process name, and arguments of such programs.

Platforms: Linux

Data Sources: Command: Command Execution, File: File Access

Sub−Technique Of: T1003

B Evaluation Details

B.1 Implementation

We used the vLLM framework [34], an open-source, high-throughput engine optimized for inference. vLLM introduces paged attention, a memory management technique that decouples attention computation from memory layout, enabling efficient dynamic batching and reducing memory fragmentation. This design improves GPU utilization by supporting continuous-token streaming and lowering overhead in multi-query processing. The system also supports asynchronous execution and tensor parallelism, allowing it to scale across multiple GPUs. Compatible with Hugging Face, vLLM offers an optimized runtime with low latency, making it well-suited for production-grade LLM serving. Additional details are available on their website[4].

B.2 Additional Benchmarks

The following benchmarks, although recognized in the literature, were excluded from our evaluations as they were either out of scope for our study or could have led to misleading conclusions. The specific justifications are detailed below.

MMLU – computer security Although highly relevant, we did not report results on the computer security section of MMLU as it contains only 100 samples, which could lead to statistically unreliable conclusions.

CyberSecEval-3 The CyberSecEval benchmarks [65], part of the Purple Llama suite [4], form a comprehensive collection of tasks aimed at evaluating the cybersecurity vulnerabilities of LLMs. CyberSecEval-3 assesses eight types of risks across two broad categories: risks to third parties, and risks to application developers and end users. While we recognize the significance of this suite, we excluded it from our evaluation because it focuses on model robustness—such as resistance to prompt injections, spear phishing, and autonomous offensive cyber operations—rather than assessing cybersecurity knowledge.

SecEval SecEval [36] provides over 1200 MCQA questions (English section) spanning nine cybersecurity domains. The dataset is generated by prompting GPT-4 with content from authoritative sources, including open-licensed textbooks, official documentation, and industry standards. We excluded SecEval from our evaluation as it appears highly *saturated*—many models, regardless of size, architecture, or pretraining corpus (e.g., Llama 3.1-8B achieves 86% while GPT-4 achieves 90%), exhibit similar performance levels.

[4] https://docs.vllm.ai/en/stable/serving/offline_inference.html

SECURE **SEC**urity Extraction, **U**nderstanding & **R**easoning **E**valuation [5] was developed to assess model performance in realistic cybersecurity scenarios. It includes six datasets focused on the Industrial Control System (ICS) domain, evaluating knowledge extraction, comprehension, and reasoning using industry-standard sources. We also omit SECURE due to its saturation; for instance, Llama 3.1-8B scores 83% and GPT-4o scores 90%.

SecQA The MCQA dataset SecQA was generated by GPT-4 using content from the textbook *Computer Systems Security: Planning for Success* [60]. Although it is a recognized resource in the field, we excluded it from our evaluation due to its small size (around 120 samples).

B.3 Prompt Design

Designing an effective prompting strategy is essential to fairly evaluate a model's knowledge, as improper phrasing or formatting can obscure what the model actually knows and lead to misleading conclusions. Since log-probabilities are not accessible in closed models, we evaluate based on the model's predicted next tokens.

B.3.1 Pretrained Models

Base pretrained LMs do not follow instructions explicitly but instead complete the given input text. Therefore, we leverage their few-shot learning capabilities. We use 5-shot prompts to help the model infer both the task and the expected answer format (e.g., MCQA or CWE ID mapping).

When available, we construct these examples from the benchmark's development (dev) set. Among our selected benchmarks, only MMLU includes a dev set; for the others, we sample examples directly from the dataset. Since we run multiple trials and report the average performance, the effect of this stochasticity is also averaged out. In each trial, the model sees a different set of examples, ensuring prompt diversity and randomness.

Lastly, although pretrained models are not instruction-following by design, we append a brief sentence—"`The following are multiple choice questions about computer security.`"—to help them infer the task context. This is a common practice in evaluations, as seen in benchmarks for models like Llama 3.1.

The input templates for pretrained models are provided for both MCQA and CWE ID mapping (i.e., CTIBench-RCM) in Figures 4 and 5.

B.3.2 Instruction-Finetuned Models

Instruct-finetuned models are designed to follow specific prompt formats but often produce inconsistent outputs in few-shot settings. Instead of returning a clean "`Answer: `" format, they may prepend phrases like "Sure, here's the answer...," which breaks format consistency and complicates evaluation.

To mitigate this, we evaluate IFT models in a zero-shot setting—prompting only with the question and extracting the answer using regex. That is, no examples are included in the input. To guide the model's output format, we append a short instruction of 1–2 sentences. If the benchmark provides its own instruction, we use it directly. Otherwise, we adopt the fusion of the instructions used in Llama 3.1 evaluations (based on MMLU) and OpenAI Simple Evals[5]: "`Given the following question and four candidate answers (A, B, C and D), choose the best answer. Your response should be of the following format: 'Answer: $LETTER' (without quotes) where LETTER is one of A, B, C, or D.`" Among our benchmarks, only CTIBench provides its own instruction. No system prompts are used in any evaluation.

Example input prompts for chat models in MCQA and CWE ID mapping tasks are shown in Figures 6 and 7.

B.4 Postprocessing and Answer Extraction

We apply several regular expression (regex) operations to the model's output string responses.

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

```
The following are multiple choice questions about computer security.

The ____________ is anything which your search engine cannot search.
A. Haunted web
B. World Wide Web
C. Surface web
D. Deep Web
Answer: D

Exploitation of the Heartbleed bug permits
A. overwriting cryptographic keys in memory
B. a kind of code injection
C. a read outside bounds of a buffer
D. a format string attack
Answer: C

.
.
.

Three of the following are classic security properties; which one is
not?
A. Confidentiality
B. Availability
C. Correctness
D. Integrity
```

Figure 4: Few-shot prompt format used with pretrained models for MCQA tasks. The model is expected to respond in the format "`Answer: X`," where X is one of A, B, C, or D.

B.4.1 MCQA Tasks

Correct Format We start by matching a case-insensitive "`answer:`"—allowing optional spaces and an optional opening parenthesis—followed by a letter A–D, and optionally ending with a closing parenthesis or a word boundary. This captures variations such as "`Answer: C`" or "`answer: (B)`". If no match is found in this format, we flag the sample as "misformatted" and proceed with the "misformatted patterns" described below. Note that pretrained models almost never misformat their responses, so these operations are primarily applied to chat models.

1. **Responses in "`answer is`":** Matches case-insensitive patterns like "`Answer is A`" or "`answer is: (D)`", allowing an optional colon, optional "(", and an uppercase A–Z, optionally ending with ")" or a word boundary.
2. **Single Letter Responses:** Captures standalone uppercase letters A–Z, optionally enclosed in parentheses, such as "`C`" or "`(B)`".
3. **Use of "`option`" instead of "`answer`":** Matches case-insensitive "option" followed by an uppercase A–Z at a word boundary, e.g., "`Option A`".

B.4.2 CWE ID Mapping Task

Since this task is part of CTIBench, we use the authors' codebase[6] to extract CWE IDs. The extraction process is similar to that of MCQA answers, where we search for the pattern "`CWE-X`," with X representing a sequence of digits.

[6] `https://github.com/xashru/cti-bench`

```
The following is a CVE description. Map it to the appropriate CWE ID.

CVE Description: A SQL injection vulnerability exists in Novel-Plus v4
.3.0-RC1 and prior. An attacker can pass specially crafted offset,
limit, and sort parameters to perform SQL injection via /novel/
userFeedback/list.
Answer: CWE-89

CVE Description: Cross-Site Request Forgery (CSRF) vulnerability in
Doofinder WP & WooCommerce Search.This issue affects Doofinder WP &
WooCommerce Search: from n/a through 2.0.33.
Answer: CWE-352

.
.
.

CVE Description: Tenda AX1803 v1.0.0.1 contains a stack overflow via
the iptv.city.vlan parameter in the function getIptvInfo.
```

Figure 5: Few-shot prompt format used with pretrained models for the CWE ID mapping task (CTIBench-RCM). The model is expected to respond in the format "`Answer: CWE-X`," where X is a unique CWE ID.

```
Given the following question and four candidate answers (A, B, C and D)
, choose the best answer. Your response should be of the following
format: 'Answer: $LETTER' (without quotes) where LETTER is one of A, B,
 C, or D.

Three of the following are classic security properties; which one is
not?
A. Confidentiality
B. Availability
C. Correctness
D. Integrity
```

Figure 6: Zero-shot prompt format used with instruct-finetuned models for MCQA tasks.

```
Analyze the following CVE description and map it to the appropriate CWE
. Provide a brief justification for your choice. Ensure the last line
of your response contains only the CWE ID.

CVE Description: Tenda AX1803 v1.0.0.1 contains a stack overflow via
the iptv.city.vlan parameter in the function getIptvInfo.
```

Figure 7: Zero-shot prompt format used with instruct-finetuned models for the CWE ID mapping task (CTIBench-RCM).