A Pretrained Language Model for Cyber Threat Intelligence

Youngja Park
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
young_park@us.ibm.com

Weiqiu You
University of Pennsylvania
Philadelphia, PA, USA
weiqiuy@seas.upenn.edu

Abstract

We present a new BERT model for the cybersecurity domain, CTI-BERT, which can improve the accuracy of cyber threat intelligence (CTI) extraction, enabling organizations to better defend against potential cyber threats. We provide detailed information about the domain corpus collection, the training methodology and its effectiveness for a variety of NLP tasks for the cybersecurity domain. The experiments show that CTI-BERT significantly outperforms several general-domain and security-domain models for these cybersecurity applications, indicating that the training data and methodology have a significant impact on the model performance.

1 Introduction

In response to rapidly growing cyber-attacks, cybersecurity experts publish many CTI reports, detailing on new security vulnerabilities and malware. While these reports help security analysts to better understand the cyber-threats, it is very difficult to digest all the information in a timely manner. Thus, automatic extraction of CTI from text has gained a lot of attention from the cybersecurity community.

However, general-domain language models (LMs) are not effective for cybersecurity text due to differences in terminology and styles. Earlier studies have demonstrated that domain-specific LMs are crucial for domain-specific applications (Beltagy et al., 2019; Lee et al., 2020; Huang et al., 2019; Peng et al., 2019; Gu et al., 2022; Chalkidis et al., 2020; Hu et al., 2022; Priyanka Ranade and Finin, 2021; Aghaei et al., 2023).

Two different approaches have been used to produce domain-specific language models: *continual pretraining* and *pretraining from scratch*. The *continual pretraining* method takes an existing general-domain model and continues training the model using a domain-specific corpus. While this approach is useful, especially when the size of the domain-specific corpus is small, the vocabulary of the new model remains largely same as that of the original model. Most domain-specific terms are thus out of vocabulary. The *pretraining from scratch* approach trains a new tokenizer to construct a domain-specific vocabulary and trains the language model using only its own corpus. Beltagy et al. (2019), Gu et al. (2022), and Hu et al. (2022) have trained BERT models from scratch for the biomedicine, computer science, and political science areas. These studies show that *pretraining from scratch* outperforms the *continual pretraining*.

Recently, a few transformers-based LMs have been built for the cybersecurity domain. CyBERT (Priyanka Ranade and Finin, 2021) trains a BERT model, and SecureBERT (Aghaei et al., 2023) trains a RoBERTa model using the continual pretraining method. jackaduma (2022) introduces SecBERT and SecRoBERTa models trained from scratch. However, these models either do not provide training details or are not evaluated on many cybersecurity tasks.

We present CTI-BERT, a BERT model pretrained from scratch with a high quality cybersecurity corpus containing CTI reports and publications. In CTI-BERT, both the vocabulary and the model weights are learned from our corpus. Further, we introduce a variety of sentence-level and token-level classification tasks and benchmark datasets for the security domain. The experimental results demonstrate that CTI-BERT outperforms other general-domain and security domain models, confirming that training a domain model from scratch with a high quality domain-specific corpus is critical.

To the best of our knowledge, this work provides the most comprehensive evaluations for classification task within the security domain. Accomplishing these tasks is a crucial part of the broader process of automatically extracting CTI, suggesting appropriate mitigation strategies, and implementing counter-measurements to thwart attacks. Thus, we see our work as an essential milestone towards

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 113–122
December 6-10, 2023

more intelligent tools for cybersecurity systems.

The main contributions of our work are the following:

- We curate a large amount of high quality cybersecurity datasets specifically designed for cyber-threat intelligence analysis.
- We develop a pre-trained BERT model tailored for the cybersecurity domain.
- We perform extensive experiments on a wide range of tasks and benchmark datasets for the security domain and demonstrate the effectiveness of our model.

2 Training Datasets

We curated a cybersecurity corpus from various reputable data sources. The documents are professionally written and cover key security topics including cyber-campaigns, malware, and security vulnerabilities. Most of the documents are in HTML and PDF formats. We processed the files using the Apache Tika parsers[1] to extract the file content. Then, we detected sentence boundaries and discarded sentences if the percentage of word tokens is less than 10% in the sentences. Table 1 summarizes our document categories and their statistics.

Document Set	# Sentences	# Tokens
Attack Description	22,086	544,260
Security Textbook	20,371	438,720
Academic Paper	1,156,026	23,245,317
Security Wiki	298,450	7,338,609
Threat Report	84,639,372	1,195,547,581
Vulnerability Description	598,265	14,123,559
Total	86,734,570	1,241,238,046

Table 1: Summary of our datasets

Attack Description This dataset includes descriptions about known cyber-attack techniques collected from MITRE ATT&CK[2] and CAPEC (Common Attack Pattern Enumeration and Classification)[3]. They are carefully curated glossaries containing the attack technique name, the definition and examples, and potential mitigation approaches.

Security Textbook The dataset contains two online text books for the CISSP (Certified Information Systems Security Professional) certification test. The CISSP textbooks cover all information security topics including access control, cryptography, hardware and network security, risk management and recovery planning.

Academic Paper This dataset contains all the papers in the proceedings of USENIX Security Symposium, a premier security conference, from year 1990 through 2021.

Security Wiki This dataset contains 7,919 Wikipedia pages belonging to the "Computer Security" category. We download the data starting from the 'Computer Security' category and recursively extracting pages from its subcategories. We discarded the subcategories not related to the cybersecurity domain.

Threat Reports This dataset contains news articles and white papers about cyber-campaigns, malware, and security vulnerabilities. These articles provide in-depth analysis on a specific cyber-attack, including the attack techniques, any known characteristics of the perpetrator, and potential mitigation methods. We collected the dataset from security companies and the APTnotes collection[4], which is a repository of technical reports on Advanced Persistent Threat (APT) groups.

Vulnerability This dataset contains records from CVE (Common Vulnerabilities and Exposures)[5] and CWE (Common Weakness Enumeration)[6], which offer the catalogs of all known vulnerabilities and provide information about the affected products, the vulnerability type, and the impact.

3 Training Methodology

We first train the WordPiece tokenizer after lowercasing the security text and produce a vocabulary with 50,000 tokens. Training a tokenizer from scratch is beneficial, as it can recognize domain-specific terms better. Table 13 in Appendix shows examples of our tokenizer and BERT for recognizing security-related words.

Following the observations by RoBERTa (Liu et al., 2019), we trained `CTI-BERT` using only the Masked Language Modeling (MLM) objective using the HuggingFace's MLM training script. The model was trained for 200,000 steps with 15% mlm probability, the sequence length of 256, the total

[1] https://tika.apache.org/
[2] https://attack.mitre.org/
[3] https://capec.mitre.org/
[4] https://github.com/aptnotes/data
[5] https://cve.mitre.org
[6] https://cwe.mitre.org/

batch size of 2,048, the learning rate of 5e-4 with learning rate warm-up to 10,000 steps and weight decay of 0.01. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e-6$.

4 Cybersecurity Applications

We evaluate CTI-BERT using several security NLP applications and compare its results with both general-domain models and other cybersecurity domain models. The baseline models are bert-base-uncased, SecBERT (BERT models) and roberta-base, SecRoBERTa and SecureBERT (RoBERTa models). All the baseline models are downloaded from HuggingFace.

The downstream applications can be categorized as sentence-level classification tasks and token-level classification tasks. The goal of the experiments is to compare different pretrained models rather than optimizing the classification models for individual tasks. Thus, we use the same model architecture and hyper-parameters to fine-tune models for all sub-tasks in each application category.

4.1 Masked Word Prediction

First, we conduct the masked token prediction task to measure how well the models understand the domain knowledge. To ensure that the test sentences are not in the training data, we use five headlines from security news published in January and February, 2023[7]. Table 2 shows the test sentences and the models' predictions. For each sentence, we conduct the masked token prediction twice with different masked words. The upper line shows the predictions for <mask>$_1$, and the lower line shows the predictions for <mask>$_2$ respectively.

The results clearly show that CTI-BERT performs very well in this test; its predictions are either the same words (boldfaced) or synonyms (italicized). Note that CTI-BERT produces RAT for "PlugX <mask>", which is a more specific term than the masked word ('malware'). RAT (Remote Access Trojan) is the malware family which PlugX belongs to. However, both SecBERT and SecRoBERTa do not perform well for this test, even though they were trained with security text. Interestingly, roberta-base performs better than these models and bert-base-uncased.

[7]beepingcomputer.com

4.2 Sentence Classification Tasks

For sentence or document-level classification, we add onto the pretrained language models a classification head, with one hidden layer and one output projection layer connected with tanh activation, which takes the average of the last hidden states of all tokens in sentences as the input. We fine-tune the pretrained models together with the randomly initialized classification layers, using 1,000 warm-up steps, with learning rate varied according to the formula in Vaswani et al. (2017). We use the Adam optimizer with β_1 = 0.9, β_2 = 0.999, and weight decay of 0.01. All the models are trained for 50 epochs with the batch size of 16 and the learning rate of 2e-5.

For the evaluation, we train five models with five different seeds (42, 142, 242, 342, and 442) for each task and report both the micro and macro mean F1 score (Mean) and the standard deviation (Std.) over the five models.

4.2.1 ATT&CK Technique Classification

The key knowledge SoC analysts look for in CTI reports is information about malware behavior and the adversary's tactics and techniques. The MITRE ATT&CK framework[8] offers a knowledge base of these adversary tactics and techniques, which has been used as a foundation for the threat models and methodologies in many security products.

To facilitate research on identifying ATT&CK techniques in prose-based CTI reports, MITRE created TRAM[9], a dataset containing sentences from CTI reports labeled with the ATT&CK techniques. We observe that TRAM contains duplicate sentences across the splits. We remove the duplicates and keep only the classes with at least one sentence in train, development and test splits. The cleaned dataset contains 1,491 sentences, 166,284 tokens, and 73 distinct classes. More detailed statistics of the dataset is shown in Table 15 in Appendix. Note that this dataset is very sparse and imbalanced. Table 3 shows the results of the six models for this task. As we can see, CTI-BERToutperforms all other models by a large margin.

4.2.2 IoT App Description Classification

IoTSpotter is a tool for automatically identifying Mobile-IoT (Internet of Things) apps, IoT-specific library, and potential vulnerabilities in the IoT

[8]https://attack.mitre.org

[9]https://github.com/center-for-threat-informed-defense/tram

Masked Sentence	bert-base-uncased	SecBERT	CTI-BERT	roberta-base	SecRoBERTa	SecureBERT
New Mirai <malware>$_1$ variant infects Linux devices to build DDoS <botnet>$_2$.	linux attacks	. attacks	**malware** **botnets**	worm attacks	this commands	**malware** **botnets**
The <Colonial>$_1$ Pipeline incident is one of the most infamous <ransomware>$_2$ attacks	oil pipeline	it targeted	**colonial** **ransomware**	Pegasus terrorist	the cyber	Olympic cyber
New stealthy Beep malware focuses heavily on <evading>$_1$ <detection>$_2$	intrusion .	antivirus 2009	**evading** **detection**	stealth **detection**	antivirus .	sandbox **detection**
Microsoft Exchange ProxyShell <flaws>$_1$ <exploited>$_2$ in new crypto-mining attack	is resulting	previously resulting	*vulnerability* **exploited**	*vulnerability* **exploited**	Key eavesdrop	Service used
PlugX <malware>$_1$ hides on USB devices to <infect>$_2$ new Windows hosts	also create	is open	*rat* **infect**	11 **infect**	silently communicate	**malware** **infect**

Table 2: Masked Word Prediction (top-1). The actual words, instead of <mask>, are shown for reference.

Model	Micro-F1 Mean	Micro-F1 Std.	Macro-F1 Mean	Macro-F1 Std.
bert-base-uncased	61.13	0.73	38.58	0.70
SecBERT	63.61	0.86	39.56	0.88
CTI-BERT	**69.30**	0.96	**46.62**	1.66
roberta-base	59.44	1.01	37.63	1.06
SecRoBERTa	57.30	0.58	35.61	0.67
SecureBERT	63.61	0.65	41.18	0.69

Table 3: ATT&CK Technique Classification Results

Model	Micro-F1 Mean	Micro-F1 Std.	Macro-F1 Mean	Macro-F1 Std.
bert-base-uncased	83.24	1.40	64.80	3.13
SecBERT	83.82	1.13	70.06	2.69
CTI-BERT	**85.18**	0.98	69.26	2.79
roberta-base	83.30	1.37	66.5	1.44
SecRoBERTa	84.24	1.01	**70.95**	2.04
SecureBERT	83.59	1.14	61.74	6.32

Table 5: Malware Sentence Classification Results

apps (Jin et al., 2022). The authors created a dataset containing the descriptions of 7,237 mobile apps which are labeled with mobile IoT apps vs. non-IoT apps with the distribution of approximately 45% and 55% respectively. They removed stopwords and put together all remaining tokens in the description ignoring the sentence boundaries. We use the datasets[10] without any further processing. The data statistics are shown in Table 16 in Appendix. The models' classification results are shown in Table 4.

Model	Micro-F1 Mean	Micro-F1 Std.	Macro-F1 Mean	Macro-F1 Std.
bert-base-uncased	95.78	0.04	95.70	0.05
SecBERT	94.22	0.21	94.12	0.21
CTI-BERT	**96.40**	0.26	**96.33**	0.26
roberta-base	95.88	0.26	95.82	0.26
SecRoBERTa	94.59	0.39	94.48	0.40
SecureBERT	96.27	0.13	96.19	0.13

Table 4: Performance for IoT App Classification

4.2.3 Malware Sentence Detection

The next two tasks, malware sentence detection and malware attribute classification, are borrowed from the SemEval-2018 Task 8, which consisted of four subtasks to measure NLP capabilities for cybersecurity reports (Phandi et al., 2018). The task provided 12,918 annotated sentences extracted from 85 APT reports, based on the MalwareTextDB work (Lim et al., 2017).

The first sub-task is to build models to extract sentences about malware. The dataset is biased with the ratios of malware and non-malware sentences being 21% and 79% respectively as shown in Table 17 in Appendix. The results are listed in Table 5 which shows that CTI-BERT and SecRoBERTaperform well on this task.

4.2.4 Malware Attribute Classification

This task classifies sentences into the malware attribute categories as defined in MAEC (Malware Attribute Enumeration and Characterization) vocabulary[11]. MAEC defines the malware attributes in a 2-level hierarchy with four high-level attribute types—*ActionName*, *Capability*, *StrategicObjectives* and *TacticalObjectives*—and 444 low-level types. This sub-task was conducted by building models for each of the four high-level attributes. Table 23 in Appendix shows more details of this

[10]https://github.com/Secure-Platforms-Lab-W-M/IoTSpotter/tree/main/data/dataset

[11]https://maecproject.github.io/

dataset for the four high-level attributes. As we can see, the datasets are very sparse with a large number of classes.

Tables 6–9 show the classification results for the four malware attribute types. We can see that `CTI-BERT` performs well, being the best or second best model, for all four attributes types.

4.3 Token Classification Tasks

Here, we compare the models' effectiveness for token-level classification using two security-domain NER tasks and a token type detection task. We use the standard sequence tagging setup and add one dense layer as the classification layer on top of the pretrained language models. The classification layer assigns each token to a label using the BIO tagging scheme. Our system is implemented in PyTorch using HuggingFace's transformers (Wolf et al., 2019). The training data is randomly shuffled, and a batch size of 16 is used with post-padding. We set the maximum sequence length to 256 and use cross entropy loss for model optimization with the learning rate of 2e-5. All other training parameters were set to the default values in transformers. Similarly to the sentence classification tasks, we train five models for each task with the same five seeds for 50 epochs and compare the average mention-level precision, recall and F1-score.

4.3.1 NER1: Coarse-grained Security Entities

Cybersecurity entities have very distinct characteristics, and many of them are out of vocabulary terms. Here, we investigate if domain specific language models can alleviate the vocabulary gap. We collected 967 CTI reports on malware and vulnerabilities. The documents are labeled with the 8 entity types defined in STIX (Structured Threat Information Expression)[12], which is a standard framework for cyber intelligence exchange. The 8 types are *Campaign* (names of cyber campaigns), *CourseOfAction* (tools or actions to take to deter cyber attacks), *ExploitTarget* (vulnerabilities targeted for exploitation), *Identity* (individuals, groups or organizations involved in attacks), *Indicator* (objects used to detect suspicious or malicious cyber activity), *Malware* (malicious codes used in cyber crimes), *Resource* (tools used for cyber attacks); and *ThreatActor* (individuals or groups that commit cyber crimes). The size of the dataset and detailed statistics of the entity types in the corpus are shown in Table 18 and Table 19 in Appendix. Table 10 shows the NER results using the mention-level micro average scores.

4.3.2 NER2: Fine-grained Security Entities

We note that some STIX entity types (esp. *Indicator*) are very broad containing many different sub-types and, thus, are difficult to be directly used by automatic threat investigation applications. We redesigned the type system into 16 types by dividing broad categories into their subcategories and annotated the test dataset from the NER1 task. We then split the dataset into a 80:10:10 ratio for the train, dev and test sets. Table 20 and Table 21 in Appendix show the statistics of this dataset. The NER results in Table 11 show that most models perform better for the finer-grained types, and especially `CTI-BERT` outperforms all other models by a large margin.

4.3.3 Token Type Classification

The token type detection task is the sub-task2 from SemEval2018 Task8 which aims to classify tokens to *Entity*, *Action* and *Modifier*, and *Other* categories. *Action* refers to an event. *Entity* refers to the initiator of the *Action* (i.e., Subject) or the recipient of the Action (i.e., Object). *Modifier* refers to tokens that provide elaboration on the *Action*. All other tokens are assigned to *Other*. More details on the dataset are shown in Table 22 in Appendix.

Even though the categories are not semantic types as in NER, this task can also be solved as a token sequence tagging problem, and, thus, we apply the same system used for the NER tasks. The classification results are shown in Table 12. Overall, the models don't perform very well likely because the mentions are long and semantically heterogeneous. The results show that the BERT based models perform better than the RoBERTa-based models.

5 Related Work

Motivated by the large-scale foundational models's successes in many general domain NLP tasks, several domain-specific language models have been developed (Roy et al., 2017, 2019; Mumtaz et al., 2020). In scientific and bio-medical domains, there are SciBERT (Beltagy et al., 2019), BlueBERT (Peng et al., 2019), ClinicalBERT (Huang et al., 2019), BioBERT (Lee et al., 2020) and PubMedBERT (Gu et al., 2022). In political and legal

[12] https://stixproject.github.io/releases/1.2

Model	Micro-F1		Macro-F1	
	Mean	Std.	Mean	Std.
bert-base-uncased	38.79	19.68	30.37	15.79
SecBERT	43.64	3.09	33.25	2.97
CTI-BERT	**55.76**	4.92	43.37	4.92
roberta-base	56.36	4.11	**44.04**	3.41
SecRoBERTa	40.00	2.27	29.03	2.39
SecureBERT	52.12	2.97	39.97	3.32

Table 6: Performance for *ActionName* attributes

Model	Micro-F1		Macro-F1	
	Mean	Std.	Mean	Std.
bert-base-uncased	60.68	3.91	51.51	5.41
SecBERT	53.18	1.82	43.39	1.9
CTI-BERT	60.91	2.34	52.23	4.39
roberta-base	59.77	3.71	50.86	3.80
SecRoBERTa	46.82	1.96	37.70	4.26
SecureBERT	**61.59**	2.73	**54.12**	4.66

Table 7: Performance for *Capability* attributes

Model	Micro-F1		Macro-F1	
	Mean	std	Mean	std
bert-base-uncased	43.71	1.46	29.31	2.27
SecBERT	38.57	2.86	21.12	2.42
CTI-BERT	45.14	4.30	28.11	4.69
roberta-base	**47.14**	2.02	**33.22**	3.81
SecRoBERTa	37.71	4.00	22.42	4.76
SecBERT	44.00	4.98	30.74	6.98

Table 8: Performance for *StrategicObjective* attributes

Model	Micro-F1		Macro-F1	
	Mean	std	Mean	std
bert-base-uncased	36.19	1.56	19.00	1.55
SecBERT	35.24	2.54	19.58	2.75
CTI-BERT	**49.84**	1.62	**31.49**	2.24
roberta-base	42.54	0.63	23.95	1.15
SecRoBERTa	35.87	3.27	20.37	4.18
SecureBERT	40.32	4.33	24.37	4.38

Table 9: Performance for *TacticalObjective* attributes

Model Type	Precison	Recall	F1
bert-base-uncased	72.04	68.67	70.31
SecBERT	69.74	63.98	66.73
CTI-BERT	**75.63**	**75.88**	**75.75**
roberta-base	72.52	68.99	70.70
SecRoBERTa	68.00	59.46	63.44
SecureBERT	73.47	72.51	72.99

Table 10: NER1 Results (mention-level micro average)

Model	Precison	Recall	F1
bert-base-uncased	73.44	68.23	70.73
SecBERT	68.58	60.90	64.43
CTI-BERT	**83.35**	**78.62**	**80.91**
roberta-base	72.17	73.51	72.80
SecRoBERTa	71.91	55.01	62.34
SecureBERT	76.66	75.98	76.30

Table 11: NER2 Results (mention-level micro average)

Model Type	Precison	Recall	F1
bert-base-uncased	**22.97**	44.51	30.27
SecBERT	21.63	36.20	27.02
CTI-BERT	22.67	**47.77**	**30.70**
roberta-base	15.05	17.44	15.97
SecRoBERTa	14.18	20.71	16.81
SecureBERT	22.58	46.97	30.46

Table 12: Token Type Classification Results (mention-level micro average)

domains, there are ConflictBERT (Hu et al., 2022) and LegalBERT (Chalkidis et al., 2020). These domain models have shown to improve the performance of downstream applications for the domain.

There have been several attempts to construct language models for the cybersecurity domain. Roy et al. (2017, 2019) propose techniques to efficiently learn domain-specific language models with a small-size in-domain corpus by incorporating external domain knowledge. They train `Word2Vec` models using malware descriptions. Similarly, Mumtaz et al. (2020) train a `Word2Vec` model using security vulnerability-related bulletins and Wikipedia pages.

Recently, transformers-based models have been built for the cybersecurity domain: CyBERT (Priyanka Ranade and Finin, 2021), SecBERT (jackaduma, 2022) and SecureBERT (Aghaei et al., 2023). CyBERT is trained with a relatively small corpus consisting of 500 security blogs, 16,000 CVE records, and the APTnotes collection. Further, CyBERT applies the continual pretraining and uses the BERT model's vocabulary after adding 1,000 most frequent words in their corpus which do not exist in the base vocabulary. SecBERT provides both BERT and RoBERTa models trained on a security corpus consisting of APTnotes, the SemEval2018 Task8 dataset and Stucco-Data[13] which contains security blogs and reports. However, the details about the data and any experimental results are not available. SecureBERT trains a RoBERTa model using security reports, white papers, academic

[13]https://stucco.github.io/data/

books, etc., which are similar to our dataset both in terms of the size and document type. However, the model is built using the continual pretraining method while CTI-BERT is trained from scratch. We believe that the main difference comes from CTI-BERT being trained from scratch and having the vocabulary specialized to the domain, compared to the extended vocabulary used in CyBERT and SecureBERT. Table 14 compares different training strategies used for these models.

6 Conclusion

We presented a new pretrained BERT model tailored for the cybersecurity domain. Specifically, we designed the model to improve the accuracy of cyber-threat intelligence extraction and understanding, such as security entity (IoCs) extraction and attack technique (TTPs) classification. As demonstrated by the experiments in Section 4, our model outperforms existing general domain and other cybersecurity domain models with the same base architecture. For future work, we plan to collect more documents to improve the model and also to train other language models to support different security applications.

Limitations

The model is pretrained using only English data. While the majority of cybersecurity-related information is distributed in English, we consider adding support for multiple languages in the future work. Further, while we demonstrate that CTI-BERT outperforms other security-specific LMs for a variety of tasks, the benchmark datasets are relatively small. Thus, the findings may not be conclusive, and further evaluations with more data are needed.

Ethical Considerations

To our knowledge, this research has a very low risk for ethical perspectives. All datasets were collected from reputable sources, which are publicly available. The only person information in our corpus is the authors' names and their affiliations in the USENIX Security proceedings. However, we do not expose their identities nor use the information in this work.

References

Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. 2023. *SecureBERT: A Domain-Specific Language Model for Cybersecurity*, pages 39–56.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Heal.*, 3(1):2:1–2:23.

Yibo Hu, MohammadSaleh Hosseini, Erick Skorupa Parolin, Javier Osorio, Latifur Khan, Patrick Brandt, and Vito D'Orazio. 2022. Conflibert: A pre-trained language model for political conflict and violence. In *The Conference of the North American Chapter of the Association for Computational Linguistics NAACL*.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *CoRR*.

jackaduma. 2022. SecBERT. https://github.com/jackaduma/SecBERT.

Xin Jin, Sunil Manandhar, Kaushal Kafle, Zhiqiang Lin, and Adwait Nadkarni. 2022. Understanding iot security from a market-scale perspective. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS*, pages 1615–1629. ACM.

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

Swee Kiat Lim, Aldrian Obaja Muis, Wei Lu, and Ong Chen Hui. 2017. MalwareTextDB: A database for annotated malware articles. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1557–1567.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sara Mumtaz, Carlos Rodriguez, Boualem Benatallah, Mortada Al-Banna, and Shayan Zamanirad. 2020. Learning word representation for the cyber security vulnerability domain. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL*, pages 58–65.

Peter Phandi, Amila Silva, and Wei Lu. 2018. Semeval-2018 task 8: Semantic extraction from cybersecurity reports using natural language processing (securenlp). In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018*, pages 697–706.

Anupam Joshi Priyanka Ranade, Aritran Piplai and Tim Finin. 2021. CyBERT: Contextualized Embeddings for the Cybersecurity Domain. In *IEEE International Conference on Big Data*.

Arpita Roy, Youngja Park, and Shimei Pan. 2017. Learning domain-specific word embeddings from sparse cybersecurity texts. *CoRR*.

Arpita Roy, Youngja Park, and Shimei Pan. 2019. Incorporating domain knowledge in learning word embedding. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 1568–1573. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

A Details on Model Training

Term	CTI-BERT	bert-base-uncased
apt*	apt, apt1, apt10, apt28, apt29, apt41, apts	apt
backdoor*	backdoor, backdoored, backdoors	–
*bot	abbot, agobot, bot, gaobot, ircbot, ourbot, qakbot, qbot, rbot, robot, sabot, sdbot, spybot, syzbot, trickbot, zbot	abbot, bot, robot, talbot
crime	crime, crimes, crimeware, cybercrime	crime, crimea, crimean, crimes
crypto*	crypto, cryptoc, cryptocurr, cryptocurrencies, cryptocurrency, cryptograph, cryptographers, cryptographic, cryptographically, cryptography, cryptojacking, cryptol, cryptolocker, cryptology, cryptom, cryptomining, cryptosystem, cryptosystems, cryptow, cryptowall	–
cyber*	cyber, cyberark, cyberattack, cyberattackers, cyberattacks, cyberb, cybercri, cybercrime, cybercrimes, cybercriminal, cybercriminals, cyberdefense, cybere, cybereason, cyberespionage, cybers, cybersec, cybersecurity, cyberspace, cyberthre, cyberthreat, cyberthreats, cyberwar, cyberwarfare, cyberweap	cyber
dark*	dark, darknet, darkreading, darks, darkside	dark, darkened, darkening, darker, darkest, darkly, darkness
hijack*	[hijack, hijacked, hijacker, hijackers, hijacking, hijacks	–
key*	key, keybase, keyboard, keyboards, keychain, keyctl, keyed, keygen, keying, keylog, keylogger, keyloggers, keylogging, keynote, keypad, keyring, keyrings, keys, keyspan, keyst, keystone, keystore, keystream, keystro, keystroke, keystrokes, keytouch, keyword, keywords	key, keyboard, keyboardist, keyboards, keynes, keynote, keys, keystone
*kit	applewebkit, bootkit, kit, rootkit, toolkit, webkit	bukit, kit
malware*	malware, malwarebytes, malwares	–
*net	botnet, cabinet, cnet, darknet, dotnet, ethernet, fortinet, genet, honeynet, inet, internet, intranet, kennet, kinet, kuznet, magnet, monet, net, phonet, planet, stuxnet, subnet, technet, telnet, vnet, x9cinternet, zdnet	barnet, baronet, bonnet, cabinet, clarinet, ethernet, hornet, internet, janet, magnet, net, planet
trojan*	trojan, trojanized, trojans	trojan
virus	antivirus, coronavirus, virus, viruses, virusscan, virustotal	virus, viruses
web*	web, webapp, webapps, webassembly, webc, webcam, webcams, webcast, webcasts, webclient, webcore, webd, webdav, webex, webgl, webhook, webin, webinar, webkit, webkitbuild, webkitgtk, weblog, weblogic, webm, webmail, webmaster, webpage, webpages, webresources, webroot, webrtc, webs, websense, webserver, webshell, website, websites, websocket, webspace, websphere, webtools, webview	web, webb, webber, weber, website, websites, webster
*ware	adware, antimalware, aware, beware, coveware, crimeware, delaware, designware, firmware, foxitsoftware, freeware, hardware, malware, middleware, radware, ransomware, scareware, shareware, slackware, software, spyware, unaware, vmware, ware, x9cmalware	aware, delaware, hardware, software, unaware, ware

Table 13: Comparison of Vocabulary. For a fair comparison, we generated our tokenizer with 30,000 tokens.

Model	Base	Training mode	Vocab.	Seq.	Batch	Train Steps
CTI-BERT	BERT-base	scratch	50,000	256	2,048	200,000
CyBERT	BERT-base	continual	29,996 (base+1,000 security)	128	–	1 epoch
SecBERT	BERT-base	scratch	52,000	–	–	–
SecRoBERTa	RoBERTa-base	scratch	52,000	–	–	–
SecureBERT	RoBERTa-base	continual	50,265 (base+17,673 security)	512	144	250,000

Table 14: Comparison of Model Training.
“– indicates the information is not available.

B Details on Experiment Datasets

	Train	Dev.	Test	Total
# Sentences	754	355	382	1,491
# Tokens	138,721	19,578	7,985	166,284

Table 15: Summary of TRAM Data

	Train	Dev	Test	Total
# Documents	5,214	1,058	965	7,237
# Tokens	635,220	133,546	106,084	874,850

Table 16: Summary of IoTSpotter Data

	Train	Dev.	Test	Total
# Sentences	9,424	1,213	618	11,255
# Tokens	1,020,655	146,362	56,216	1,223,233

Table 17: Summary of the Malware Sentence Data

	Train	Dev	Test	Total
# Documents	667	167	133	967
# Sentences	38,721	6,322	9,837	54,880
# Tokens	465,826	92,788	119,613	678,227

Table 18: Summary of the NER1 Dataset

Entity Type	Train	Dev	Test
Campaign	247	27	85
CourseOfAction	1,938	779	329
ExploitTarget	5,839	1,412	1,282
Identity	6,175	1,262	1,692
Indicator	3,718	1,071	886
Malware	4,252	776	1,027
Resource	438	91	114
ThreatActor	755	91	144

Table 19: Entity Types and Distributions in the NER1 Dataset

	Train	Dev	Test	Total
# Documents	106	14	13	133
# Sentences	5,206	561	671	6,438
# Tokens	75,969	8,106	9,984	94,059

Table 20: NER2 Dataset

Entity Type	Train	Dev	Test	Total
Campaign	39	0	4	43
SecurityAdvisory	54	12	30	96
Vulnerability	401	50	86	537
DomainName	169	3	16	188
EmailAddress	6	1	1	8
Endpoint	3	0	0	3
FileName	210	37	24	271
Hash	93	5	3	101
IpAddress	37	0	2	39
Network	3	0	0	3
URL	181	20	27	228
WindowsRegistry	9	0	0	9
AvSignature	99	13	10	122
MalwareFamily	554	53	47	654
Technique	334	39	76	449
ThreatActor	89	4	7	100

Table 21: Entity Types and Distributions in the NER2 Dataset

	#Doc.	#Sent.	#Action	#Entity	#Mod.
Train	65	9,424	3,202	6,875	2,011
Dev	5	1,213	122	254	79
Test	5	618	125	249	79
Total	75	11,255	3,449	7,378	2,169

Table 22: Dataset for Token Type Classification

Split	*ActionName*			*Capability*		
	#Doc.	#Sent.	#Class	#Doc.	#Sent	#Class
Train	65	1,154	99	65	2,817	20
Dev.	5	46	20	5	102	13
Test	5	33	18	5	88	14

Split	*StrategicObjectives*			*TacticalObjectives*		
	#Doc.	#Sent.	#Class	#Doc.	#Sent.	#Class
Train	65	2,206	53	65	1,783	93
Dev.	5	77	28	5	63	26
Test	5	70	21	5	63	27

Table 23: Data Statistics for Malware Attribute Classification