Open Source Intelligence for Cybersecurity Events via Twitter Data

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science

by

Dakota Steven Dale
University of Arkansas
Bachelor of Science in Computer Science, 2021

May 2023
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

Qinghua Li, Ph.D.
Committee Chair

Susan Gauch, Ph.D.
Committee Member

Kevin Jin, Ph.D.
Committee Member

Abstract

Open-Source Intelligence (OSINT) is largely regarded as a necessary component for cybersecurity intelligence gathering to secure network systems. With the advancement of artificial intelligence (AI) and increasing usage of social media, like Twitter, we have a unique opportunity to obtain and aggregate information from social media. In this study, we propose an AI-based scheme capable of automatically pulling information from Twitter, filtering out security-irrelevant tweets, performing natural language analysis to correlate the tweets about each cybersecurity event (e.g., a malware campaign), and validating the information. This scheme has many applications, such as providing a means for security operators to gain insight into ongoing events and helping them prioritize vulnerabilities to deal with. To give examples of the possible uses, we present three case studies demonstrating the event discovery and investigation processes. We also examine the potential of OSINT for identifying the network protocols associated with specific events, which can aid in the mitigation procedures by informing operators if the vulnerability is exploitable given their system's network configurations.

Acknowledgments

This work is supported in part by the National Science Foundation under award 1751255 and by the Department of Energy under award DE-CR0000003. I would also like to express my gratitude to my advisor, Dr. Qinghua Li, for guiding me through this thesis as well as my committee, Dr. Susan Gauch and Dr. Kevin Jin.

Contents

List of Figures

List of Tables

Disclaimer

Portions of the below publication were used in Chapters 1-4 and Chapter 6.

Dakota Dale, Kylie McClanahan, and Qinghua Li. "AI-based Cyber Event OSINT via Twitter Data". In: *2023 International Conference on Computing, Networking and Communications (ICNC)*. 2023, pp. 436–442. DOI: `10.1109/ICNC57223.2023.10074187`

Chapter 1

Introduction

1.1 Impact of Cyberattacks

Cyber attacks on network infrastructures are becoming more frequent and catastrophic. For example, Microsoft was a victim of a data breach discovered in January 2020 that caused over 250 million customer records leaked online [2]. In May 2021, Colonial Pipeline was hit with a ransomware attack [3] that caused gas supply to the east coast of the United States to be completely cut off for days. In 2015, a cyber attack against power grids left hundreds of thousands of people in Ukraine without power for six hours [4]. Not to mention the average cost of a data breach rose from $3.86 million in 2020 to $4.24 million in 2021 [5]. Cyber attacks are also constantly evolving. Sonicwall found 442,151 new malware variants in 2021, an increase of 65% over 2020 [6]. The severity of malware has increased too. In 2020, 9 malicious actions on average were performed by a single piece of malware. This rose to 11 actions per file in 2021 [7].

Energy infrastructure is also affected by these developments. In May of 2021, the largest fuel pipeline in the United States, Colonial Pipeline, was hit with what is considered as the most severe ransomware attack against the energy infrastructure [3]. Colonial was able to resume operations after paying $4.4 million in bitcoin as ransom. However, during this period, gas supply to the east coast was completely cut off. Though the ransom was paid, the hackers still made away with 100 gigabytes of information [8]. In 2015, a cyber attack against power grids left hundreds of thousands of people in Ukraine without power for six hours [4]. Attackers shut down at least 30 substations, disabled backup power, and even sabotaged the doors leading out of the operators' workstations. These attackers used a Trojan called BlackEnergy to access these networks and plant the KillDisk malware on the target systems [9].

1.2 Background

To better protect network and information systems, it is crucial for a system to identify these cyber events as they occur and also aggregate all of the necessary pieces of information. With the rate of these attacks as well as the speed of their evolution, the system must work real-time.

Open-source intelligence (OSINT) is one of the most important tools for securing cyberspace [10]. OSINT refers to the search and collection of intelligence through public resources such as datasets, blogs, or social media sites. This process is also aiding decision-making for policy, foreign affairs, and the economy [11]. There are two main categories of cybersecurity intelligence sources: formal and informal. Formal sources are typically government-sponsored sites that collect technical information on cyber vulnerabilities such as the National Vulnerability Database (NVD) or the Cybersecurity & Infrastructure Security Agency (CISA). Conversely, informal sources are developed by independent entities such as contractors or hobbyists and then made available online through a blog or social media page.

Due to the widespread usage of social media by both governmental organizations and independent entities, as well as the volume and velocity with which data is produced, Twitter seems to be a highly viable data source for cybersecurity intelligence, with 330 million monthly active users posting 500 million tweets per day [12]. The global 2017 "Petya/NotPetya" ransomware attack was discussed on Twitter as early as four months before the attack went public [13]. Hacktivists themselves even take to the social media to disseminate vulnerability information amongst their collective. In one incident, a malicious threat actor known as SandboxEscaper once released a zero-day, or previously unknown, vulnerability as well as linked proof-of-concept code in a public GitHub repository on Twitter. Less than two days later, a group known as PowerPool began exploiting the vulnerability in their own hacking campaign [14].

1.3 Limitations of Existing Work

In recent years, there have been several studies on leveraging Twitter data for cybersecurity OSINT, but these studies are not without limitation. For instance several studies forgo any form of content-based filtering for the tweets themselves, relying on the number of entities extracted, the emotional sentiment, or the account's follower count to determine a tweets usefulness [15, 14, 16]. In practice, this could hinder the coverage of accounts used by relying primarily on professional, well-known accounts and ignoring those of hobbyists and, potentially, the hackers themselves. A secondary limitation is the lack of the ability to connect information across tweets/accounts. Most studies will simply provide the tweet verbatim, output some form of word frequency aggregation, or issue an alert [17, 18]. Part of what makes OSINT effective is the ability to pull from multiple sources. However, not every source may have a complete understanding of the event in question. Thus, the functionality of aggregating information across sources is crucial.

1.4 Overview of Thesis

In this study, we leverage AI and Twitter data for OSINT. We propose a method to automatically collect and correlate the necessary pieces of information about specific cyber events (e.g., a malware campaign or vulnerability) for security operators to better understand them and make more timely and informed decisions. Specifically, the system pulls tweets and passes them through a multi-step AI pipeline. The first step serves as a cyber event detection model to identify not only tweets in the cybersecurity domain, but also to detect those that contain valuable information about specific events. The second step performs named entity recognition (NER), a natural language processing (NLP) technique, so that the names of different malware, threat actors, or companies can be extracted. Next, the named entities are used to create a word co-occurrence network to assess the correlation amongst entities and cluster them around events. Lastly, the subcomponents of the word co-occurrence network are validated against phenomena such as Twitter spam that can hinder the integrity of the information collected. The results of this pipeline are output to the user to show the current sub-topics in the cybersecurity community on Twitter and their potential to represent real events occurring.

In addition to our empirical method evaluations, we also present three case studies. Event discovery is the first of these studies. This use case follows a security operator who is simply checking the twitter stream to see what events may be occuring without a specific one in mind. Second, we evaluate the systems functionality on investigating a specific event. By altering the initial search term, the user can focus the twitter stream to include a vulnerability, a particular product, or even a malware. Third, we test how the pipeline can be used to prioritize vulnerability patching. We take two vulnerabilities and investigate them using the program. Based on the severity of the information found, the user should be able to make a decision on which to prioritize.

Lastly, we evaluate the potential of implementing OSINT for the identification of network protocols. Network protocols provide necessary pieces of information for understanding the mechanics of a given event. However, identification methods vary given different criteria and thus, no one method may work for all targets. To conclude this thesis, we evaluate the efficacy of OSINT for network protocol identification via a keyword-based method and a named entity recognition model.

1.5 Thesis Structure

The rest of this thesis is organized as follows. An examination of related work is provided in chapter 2. Chapter 3 details the methology and evaluations of our method along with three case studies. Chapter 5 describes the methodology and results of leveraging OSINT for network protocol identification. Last, chapter 6 provides our closing statements.

Chapter 2

Related Work

2.1 Deep Learning for Text Classification

Deep learning encompasses a variety of models all mimicking the structure of the human brain. However, some forms of neural networks are better suited for certain problems than others. Recurrent neural networks (RNN) are particularly suited to handle sequential or time-dependent input data [19], and the use of RNNs in processing or generating natural language is well established [20, 21]. Furthermore, Gated Recurrent Unit (GRU) models are a form of RNN that is well suited for capturing the link between elements in a sequence [22], like words in a tweet. [23] found that for text classification tasks such as sentiment and relation classification, GRU models outperform both convolutional neural networks (CNN) and LSTM models, especially with sentence lengths greater than 10 words. However, RNNs are particularly vulnerable to what is known as the vanishing gradient problem. The vanishing gradient problem essentially causes neural networks to stop learning past a certain point due, this is especially true for RNNs dealing with longer sequences of data. However, contrary to plain RNNs, GRU architectures do not suffer from the vanishing gradient problem [24]. Regardless, some studies take this a step further and create bidirectional models. The bidirectional aspect of the model allows data to be passed both forwards and backwards which should alleviate any performance drops concerning the length of the sequence [25]. [26] found that their bidirectional gated recurrent unit (BiGRU) architecture achieved a testing accuracy of 91.6% when used in conjunction with GloVe embeddings after only 10 epochs for text classification.

2.2 Social Media OSINT

In recent years, social media sites have been investigated as potential sources of open source intelligence. [27] sought to develop an OSINT and social network visualization tool based on Facebook that utilized heuristic and named entity recognition based approaches to investigate organized crime. However, no empirical evaluations of the methods were provided. Another study, [28], compared machine learning and dictionary-based approaches to detecting malicious insiders, a problem effecting both the cybersecurity and broader corporate spaces, from user comments on YouTube. The highest accuracy of these methods was 81%. Twitter specifically has been implemented for OSINT in a number of fields such as natural disaster support [29, 30], analyzing the stock market [31], and predicting elections [32]. Furthermore, the two most prevalent applications of OSINT are in the cybersecurity and social media domains [33].

Loosely speaking, the applications of Twitter for cybersecurity OSINT can be split into two groups: alert-oriented and investigation-oriented. Alert-oriented methods seek to bring specific events to a security operator's attention. CyberTwitter [15] for instance proposes a profiling system that extracts vulnerability information about a user's installed software programs and browser extensions. Through the use of an ontology and concept extractor, the system is able to correlate Twitter information with the derived profile and issue alerts for individual users. [16] also aims to address cyber event detection, but through the use of NER and keyword-based approaches. This system functions as a trend detector, but for cyber threat events specifically. [18] creates a system to identify tweets about cyber threats and extract named entities from each tweet to generate a security alert or fill in an indicator of compromise. Comparatively, investigation-oriented methods seek to aid operators in the investigative process by aggregating Twitter information. Typically, these systems make no decisions themselves. [17] provides cyber intelligence by using NLP techniques and a specialized cyber-domain entity extractor. Through these methods, the system can output information like important tweets and significant keywords. [14] also identifies cyber threat tweets but through keyword heuristics and sentiment analysis with the goal of enhancing preemptive preparedness for cyber events.

2.3 NER for Cybersecurity

One of the primary problems within natural language processing tasks is the ability to identify named entities such as people, companies, or products. There are three primary categories for NER: rule-based, dictionary-based, and machine learning-based. Out of these, supervised machine learning is the most adaptable and provides a reasonably high accuracy [34]. One study that sought to quantify the significance of a given piece of cybersecurity related text implemented the python library spacy‘s en_core_web_lg pipeline for NER. After adding a custom label, “ITProduct”, their model yielded an F1 score of 74.9 [35]. A similar study seeking to extract exploit information from news articles compared two NER architectures: a BiLSTM and a spacy pipeline. The study found that the spacy pipeline vastly outperformed the BiLSTM in every task [35].

Chapter 3

Method

Overall, our method seeks to enable security operators to make more effective and timely decisions by aggregating relevant information from Twitter and presenting it to them in an holistic format.

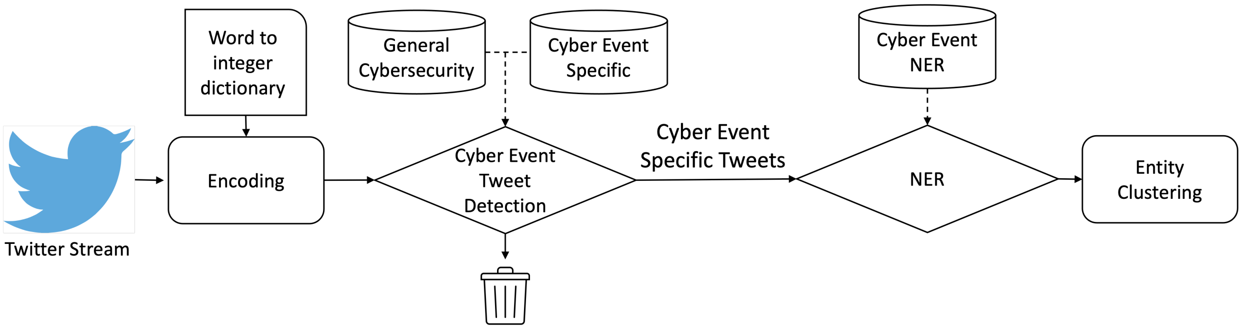


Figure 3.1: System workflow.

3.1 Overview

Our pipeline (see Figure 3.1) contains three main phases: Cyber Event Tweet Detection, NER, and Event-centric Entity Cluster Identification and Validation. The first begins with pulling tweets from Twitter. These tweets are then passed to our Cyber Event Tweet Detection model which we trained using the concept of Transfer Learning. Transfer Learning is used in cases where the target data is too limited to train a classifier, so instead the classifier is trained on a related subject area with more prevalent data, then finalized using the target data. In our case, a cyber event-specific dataset can only be built by hand but a large dataset would be required to build a competent model. Thus, we pre-train the model on a large number of general cybersecurity tweets, which are easier to automatically obtain, and then fine-tune it on a small dataset of manually selected cyber event tweets. Once the model identifies the cyber event tweets, they are passed on to the next phase. The NER phase extracts the valuable information from the tweets, so that they can be clustered by occurrence in the Event Cluster Identification phase. In this last phase, each cluster, representing an independent cyber event, is also validated to provide users with metrics to gauge

Table 3.1: Search & Filtering terms to identify viable Twitter accounts

Filtering	Keywords
Initial Twitter Stream	#cybersecurity, #vulnerability, #cyber, #cyberattack, #infosec, #ransomware, #malware, #hack, #hacker
Account Bio	Founder, Analyst, Scientist, Director, cybersecurity, hacker, Center, Centre, Dr., Doctor, CIO, Chief Innovation Officer, CEO, Chief Executive Officer

the spread and validity of the information. The user is provided with a report of each event with notable tweets for each and an interactive word co-occurrence network.

The primary users of our solution are professionals that oversee the cybersecurity risks of organizations such as security operators. They can use our approach for many purposes, e.g., investigating how on-going cyber attacks could affect their organization and how their un-patched vulnerabilities could soon be exploited.

3.2 Data Collection

3.2.1 Data for Cyber Event Tweet Detection

Due to the use of transfer learning, the cyber event tweet detection module needs general cybersecurity tweets, non-security tweets, and cyber event-specific tweets. To begin building a dataset of general cybersecurity tweets, we elected to first identify accounts belonging to industry professionals, similar to [14]. Through the use of Twint, an open-source Twitter scraping tool, we extracted 119 accounts who had posted a tweet containing keywords pertaining to cybersecurity such as “vulnerability”, “cybersecurity”, “phishing”, etc. To validate that the account is a credible source, we filtered the accounts by ensuring that their bios contain a title granting them some credibility in the field, essentially certifying the account. Table 3.1 shows the list of the search and filtering keywords. These steps derive the 332,518 positive data samples of general cybersecurity tweets.

We still need negative samples of general cybersecurity tweets. A 2009 study developed a means of detecting the sentiment, or the opinions/emotions, of tweets. Due to the lack of large publicly available tweet datasets, researchers in that study had to create one [36]. This dataset, commonly known as the Sentiment-140, contains 1.6 million tweets each labeled as having either negative, neutral, or positive sentiment. Since its release, the Sentiment-140 dataset has been used to assess the opinions of scientific studies online [37], to predict stock movement [38], and even to enhance other datasets [39]. Considering these samples were queried from the Twitter stream according to one of two emoticons, :) and :(, it is reasonable to assume the majority are distributed into categories outside of cybersecurity. After the removal of stop words such as "the" and "at" along with any links present in the tweet, we were able to analyze the word frequencies of the data set. This analysis shows that the most common words used were "good", "day", "get", "like", and "go" thus showing no indication of cybersecurity tweets. Additionally, we found that the words "cybersecurity", "cyber", "hacker", "malware", "vulnerability", and "exploit" only accounted for 219 of the 12,276,829 word occurrences. Thus, any tweets from this dataset that happen to fall in the cybersecurity domain should be overshadowed by the others, and their interference to the model should be minimal. 320,351 tweets are randomly selected from the Sentiment-140 dataset to serve as the negative class in our dataset.

Next, we collected the cyber event-specific dataset. This dataset will be used to fine-tune the cyber event tweet detection model. Due to the nature of cyber security information on Twitter, these tweets were selected by hand. Companies often share basic tips and tricks (i.e. "10 tips on how to avoid phishing scams"), but these are not useful for security operators to understand a specific event. Following a similar methodology as before, we manually searched a series of keywords related to cybersecurity and selected 181 tweets containing valid information such as CVEs (Common Vulnerabilities and Exposures), software assets, or malware families to serve as the positive samples. Again, we appended an equivalent number of tweets from the Sentiment-140 dataset for the negative samples.

Table 3.2: Entity labels for the NER model

Label	Examples
AttackType	Phishing, DDos, SQL Injection
Cardinal	1, two, 3, four
CVE	CVE-2022-23657, cve-2022-23658
Global-Political Entity (GPE)	United States, Russia, China, Canada
MalwareType	Ransomware, Spyware, Ryuk, Petya
Money	$50, six dollars, one-hundred euro
Ordinal	First, 2nd, Third, 4th
Organization (ORG)	Apple, Microsoft, Meta
Nationality, Religious, or Political groups (NORP)	American, Russian, Muslim, Democrat, Republican
Percent	10%, twenty percent
Product	iPhone, Windows, iOS

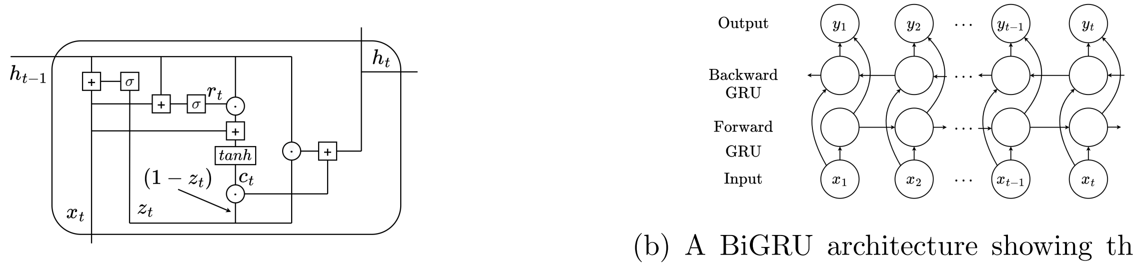
3.2.2 Data for Named Entity Recognition

Again using Twint, we scraped approximately 11,000 new tweets by keywords defined in Table 3.1, and then routed them through the cyber event tweet detection model for filtering. Due to the output of neural networks being a continuous variable, we established a threshold of 0.5 to convert the probabilistic values into binary decisions. Tweets generating a value less than or equal to 0.5 are reassigned to 0 (denoting a non-cyber event tweet) and the rest to 1 (denoting a cyber event tweet). This left around 5,000 tweets containing valuable cyber event information. Using the Prodigy annotation software, we labeled the entities that may be useful for security operators to know, such as types of malware or CVEs. A complete list of entity labels, as well as some examples, are provided in Table 3.2.

3.3 Cyber Event Tweet Detection

3.3.1 Architecture

We adopt a similar neural network architecture to [26] using BiGRU layers, but instead of GloVe embeddings, we use a randomly initialized embedding layer which will convert n integer-based word encodings into vectors of length d, thus creating an $n * d$ matrix representation of the entire tweet. In [26], the authors used GloVe embeddings for any token that was available; however, they approximate that 18% of the tokens from their dataset do not have GloVe mappings, thus warranting the use of random vectors for those tokens. We



(a) Diagram of a GRU unit

(b) A BiGRU architecture showing the direction

Figure 3.2: Diagrams for GRU and BiGRU

found that approximately 44% of our tokens lacked a GloVe mapping. Considering nearly half of our tokens would need random vectors, we elected to use the randomly initialized embedding layer to generate uniformly distributed random vectors for every token.

The equations to compute the output of a GRU unit as shown in Figure 3.2a with input x_t and the output of the previous unit h_{t-1} are shown below, where σ is the sigmoid function and W_i, W_r, and W_c are the weights for the input, the reset gate, and the current memory content, respectively.

$$z_t = \sigma(W_i * [h_{t-1}, x_t]) \tag{3.1}$$

$$r_t = \sigma(W_r * [h_{t-1}, x_t]) \tag{3.2}$$

$$c_t = tanh(W_c * [r_t \cdot h_{t-1}, x_t]) \tag{3.3}$$

$$h_t = (1 - z_t) \cdot c_t + z_t \cdot h_{t-1} \tag{3.4}$$

For a BiGRU architecture, one set of GRU units processes the input from start to end, while a second processes the input in reverse, as shown in Figure 3.2b.

We use an output dimension d of size 25 along with an embeddings regularizer ($L_2 = 0.0438$). The architecture contains a BiGRU layer with 128 state cells. Contrary to [26], we skip the concatenation of the BiGRU outputs and instead route them through a fully connected (FC) layer of 128 units. Lastly, the model includes a dropout layer with a 33% rate and a FC layer with 1 unit to achieve a binary classification.

Table 3.3: Cyber Event Detection Model Architecture

Layer Type	Output Shape
Embedding	(None, 140, 25)
Bidirectional	(None, 256)
Dense	(None, 128)
Dropout	(None, 128)
Dense	(None, 1)

3.3.2 Encoding

Before we could begin training the model, we first need to encode the tweets. The embedding layer present in the architecture detailed in Table 3.3 takes in an array of integers and converts them into vectors of uniform distribution. The maximum number of words possible in a tweet is 140, meaning that our array would contain 140 integers. Thus, we created a dictionary mapping each unique word present in the training data to a specific integer, with a few exceptions.

First, as shown in Table 3.2, one of our primary labels for NER is the CVE. CVEs essentially act as reference tags for vulnerabilities. This also means that every CVE is unique. Mapping every CVE to its own integer is unhelpful, especially since any tweet containing a CVE should automatically be considered a positive case for the cyber event detection algorithm. Instead, we elected to map all CVEs to one integer: 2. Similarly, the set of unique words in the dataset is hardly all-encompassing and new malware and threat groups come out frequently. To remedy this and avoid issues with the model, any words not present in the dictionary, i.e. those without a mapping, are mapped to 1. Lastly, most tweets will not contain 140 words, so we must pad those entries with 0s to maintain a uniform size.

3.3.3 Training

As aforementioned, transfer learning is used. We first train a model to identify general cybersecurity tweets, and then train a model to identify cybersecurity events. To train the model to identify tweets relating to the cybersecurity domain as a whole, we trained it with

50% of the “General Cybersecurity dataset” mentioned previously and validated it with the remaining 50%. We specified 20 epochs and a batch size of 64. Then, using the “Cyber Event Specific” dataset, we focused the pre-trained model to identify tweets containing valuable event-specific information. Because this dataset is rather small, we elected to use a 90/10 training/testing split, 100 epochs, and a batch size of 20.

Training utilized binary crossentropy loss (Eq. (3.5)) and the Adam optimizer with a learning rate of 0.001. We also elected to implement two functions that augment the training process as a whole, known as callbacks: ReduceLRonPlateau and EarlyStopping. ReduceLRonPlateau keeps track of the validation loss and lower the learning rate by a factor of 0.1 to avoid overshooting the local minima. Similarly, EarlyStopping also keeps track of the validation loss but will stop the training process entirely if it stagnates for longer than 3 epochs.

$$\text{loss} = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \tag{3.5}$$

3.4 Named Entity Recognition

At this point, the Twitter stream has been filtered to only include tweets containing valuable information about cybersecurity events. Thus, it is time to extract the specific entities that may be useful in describing them. The NLP library spaCy provides several pre-trained model pipelines, notably the en_core_web_md pipeline. This pipeline has been used in multiple domains from extracting brand names for sentiment analysis [40] to correlating diseases with specific pre-existing conditions [41]. It takes the raw text as input and passes it through a series of components, each taking in the output of the previous component and passing its own output to the next. Most of these components provide simple but necessary functions, such as word vectorization and part-of-speech tagging. Lastly, the NER component allows for the labeling of non-overlapping spans, so we can extract entities such as companies and malware families from tweets. Overall, this pipeline features 685 thousand keys and 20 thousand vectors of dimension 300. For training the pipeline, we kept the default configuration of an 80/20 training/testing split and the initial learning rate of 0.01.

The components besides tok2vec and NER were frozen to avoid changing their weights.

3.5 Event Cluster Identification and Validation

After the entities are extracted, we aggregate them into specific events and analyze their degree of co-occurrence. For this phase we used three python libraries: Pandas, Pyvis, and Networkx. After iterating through the entities and creating a Pandas DataFrame of entity pairs and their number of co-occurrences, we used Pyvis and Networkx to construct undirected network graphs. Fig. 3.3 shows an example Pyvis graph. The nodes show the entities present in the initial scrape of the program. Each edge connecting two nodes shows the number of co-occurrences, or weight, of the pair. To avoid cluttering the network, we filtered out any edges where the weight was one, as one co-occurrence is not evidence of a strong correlation. Using the Networkx graph, we segmented the graph into its connected subcomponents (i.e., subgraphs) for further analysis. Each subgraph can be understood as a self-contained event and demonstrates the connections between the primary entities.

$$\text{Diffusion Index} = \#\text{Unique Users}/\#\text{Total Tweets} \tag{3.6}$$

$$\text{Spam Index} = \frac{\sum_{n=0}^{\#users}(\frac{1}{\#\text{ nth User Tweets}})}{\#\text{Total Tweets}} \tag{3.7}$$

Due to the varying degree of veracity, or truthfulness, of social media data, it needs some means of validation. Measuring the frequency of tweets about a certain topic is not enough to gauge their validity because in extreme cases, those tweets could be coming from a single user in an attempt to clog or redirect the focus of the Twitter stream. Thus, measuring the number of tweets about a subject while also adjusting for number of accounts participating in the conversation is important. We adopt two metrics from a prior study [42]. The Diffusion Index, given by Eq. (3.6) measures how quickly information has spread. The Spam Index, given in Eq. (3.7), measures repeated tweets from the same user. It can be viewed as inflating the diffusion. In practice, we would grant more trust to topics with high diffusion and spam indices. A low diffusion would signify that a very small number of

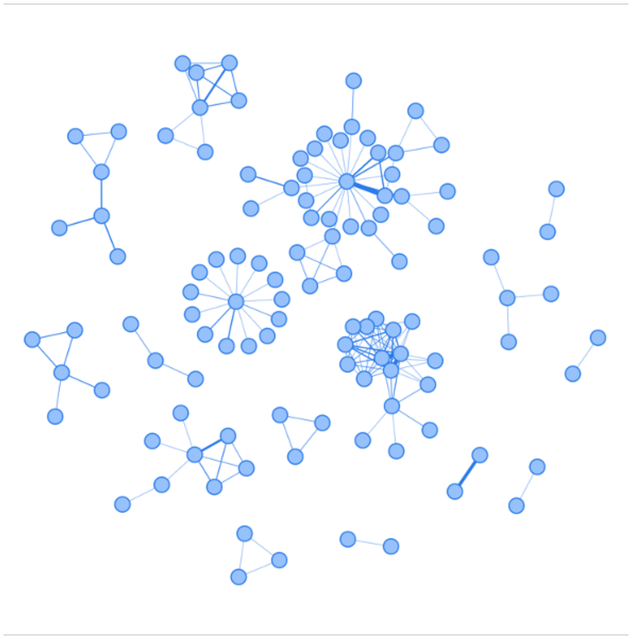


Figure 3.3: An Example Network Graph

users, potentially a community, are discussing the event. Though this is does not inherently make the information false, when coupled with a low spam index (signifying disproportionate tweet contribution), it could be evident of tactics to obscure other, more severe, events.

3.6 Method Evaluations

3.6.1 Evaluation Metrics

We use 6 metrics: Area under the Receiver Operating Characteristic curve (AUC-ROC), Accuracy, Precision, Recall, F_1-score, and Specificity. The AUC-ROC score compares the true and false positive rates across different discrimination thresholds. The curve can be compared to a line from (0,0) to (1,1) which represents an untrained classifier that will label the samples randomly. The greater the ROC curve's deviation from this line, the higher the performance. The remaining metrics all operate on a binary basis, meaning any values above a threshold of 0.5 are converted to 1 (positive) and the rest to 0 (negative). The use of binary classification generates four cases: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy is defined as the number of correct, or true, predictions divided by the total number of predictions. Precision, defined in equation

3.8, focuses on how well the model detects the positive class and does not take into account any values concerning the negative case. Conversely, recall (defined in equation 3.9) accounts for the false negatives, or samples incorrectly classified as negative.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{3.8}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{3.9}$$

F_1-score, defined in equation 3.10 is more robust against class imbalances than standard accuracy is, but it does not demonstrate the models' performance on the negative class.

$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.10}$$

Specificity (equation 3.11) denotes the proportion of true negatives to the number of samples the model classified as negative, and should help demonstrate how well non-cyber event tweets are discarded.

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})} \tag{3.11}$$

3.6.2 Cyber Event Tweet Detection

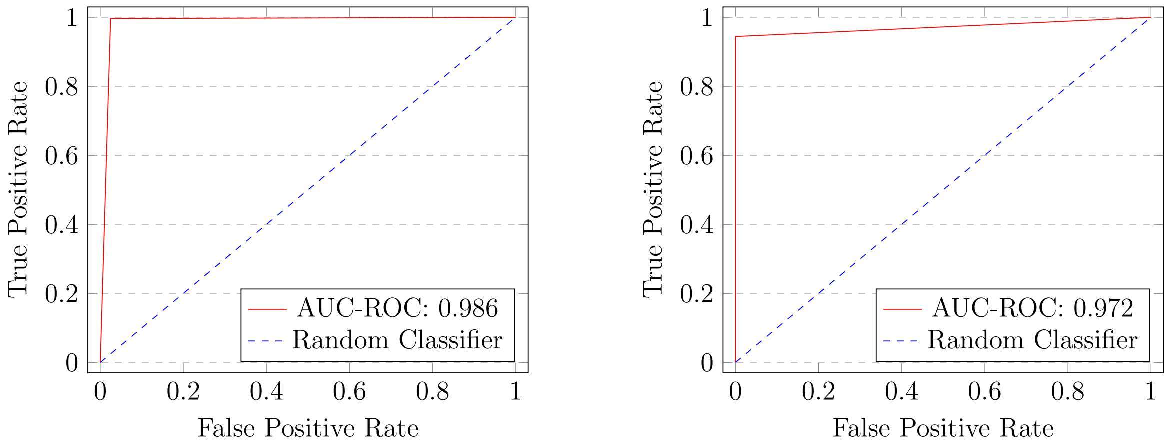
Fig. 3.4 shows the performance of cyber event tweet detection. On a MacBook Pro (2017) with a 3.1 GHz dual-core Intel i5 processor, pre-training and training the model took just over 9 hours (543 minutes) and approximately 90 seconds respectively whereas model predictions take approximately 5ms per sample. The pre-trained model achieved a 98.6% accuracy and a 98.6% AUC-ROC (Fig. 3.4c) score. It yielded 97.7%, 99.6%, 98.7%, and 97.6% for precision, recall, F_1 score, and specificity respectively. The results show that the model can very accurately identify general cybersecurity tweets. The final cyber event tweet detection model achieved a 97.2% AUC-ROC score (Fig. 3.4d) and 96.9% accuracy on its testing set.

		Predicted	
		Non-Cyber.	Cyber.
True	Non-Cyber.	0.975	0.025
	Cyber.	0.004	0.996

(a) Confusion Matrix for Pre-Trained Model

		Predicted	
		Non-Event	Event
True	Non-Event	0.938	0.062
	Event	0.00	1.00

(b) Confusion Matrix for Final Model



(c) ROC Curve for Pre-Trained Model

(d) ROC Curve for Final Model

Figure 3.4: Performances of Cyber Event Tweet Detection

After inspecting the confusion matrix in Fig. 3.4b, it is clear that the model can correctly classify cyber event tweets, as it receives a recall of 1. However, a small portion of tweets that did not pertain to a cyber event were not discarded, thus generating false positives and resulting in a 94.1% precision. The model got an F_1 of 97.0% and specificity of 95.2%. Overall, the performance is good, although not as good as the pre-trained model due to insufficient cyber event tweets in the training data.

3.6.3 Named Entity Recognition

Again, on a MacBook Pro (2017) with a 3.1 GHz dual-core Intel i5 processor, training the model took approximately 38 minutes and 17 ms per prediction. Overall, the method received 88.55%, 89.70%, and 87.43% for F_1, precision, and recall respectively. However, further examination revealed that entity labels crucial to the function of the scheme, such as CVE, MalwareType, and AttackType, all receive F-scores above 94.0%. The labels where the model's ability is lacking (F_1 below 85%) are Org, Product, and Cardinal. A full breakdown of the NER model's performance by label is provided in Table 3.4.

Table 3.4: NER Performance Per Label

Label	Precision	Recall	F_1 Score
CVE	98.85	98.85	98.85
MalwareType	94.74	96.64	95.68
Ordinal	100.00	90.00	94.74
AttackType	98.32	90.70	94.74
Money	94.74	90.00	92.31
NORP	97.83	84.91	90.91
GPE	92.02	89.82	90.91
Percent	81.82	90.00	85.71
Org	82.59	76.34	79.34
Product	75.00	84.00	79.25
Cardinal	71.11	78.05	74.42

3.6.4 Entity Clustering

To evaluate the efficacy of clustering named entities by their co-occurrence for cyber event detection (see Fig. 3.3 for examples), we randomly selected nine event clusters as test cases. By manually comparing each edge within the cluster network against what is available online and our own knowledge, we were able to gauge the relevancy/correctness of the edges/connections between different nodes. Since the full clusters contain too many edges to manually verify, for each cluster we only checked the edges connected to the central node (i.e., the node with most edges in the cluster). As shown in table 3.5, the percentage of correct edges ranges from under 70% to 100%. Overall, we found the percentage of correct edges returned by our solution to have a weighted average of approximately 84.41%, showing a good accuracy.

Table 3.5: Event Cluster Validity

Central Node	#Edges	Percentage of Correct Edges
CVE-2022-30129	2	100.00%
CVE-2022-29866	6	100.00%
Spotify	3	100.00%
Facebook	10	100.00%
Apple	65	96.92%
Android	36	88.89%
Microsoft	21	76.19%
MacOS	69	72.46%
cve-2022-32893	19	68.42%

Chapter 4

Case Studies

4.1 Cyber Event Discovery

The first use case is the discovery of new cyber events. Under these circumstances, the user would use the program's default search parameters to capture the largest breadth of Twitter chatter possible. Security officers could in turn use this information to defend their systems, potentially even before major security information providers begin reporting on it. This case study was conducted on May 23, 2022 at approximately 10 a.m. After an initial run, the program revealed 14 events that might be occurring. The top 5 events with the highest weight are provided in Table 4.1 along with their diffusion and spam indices, and a notable tweet. With the tweet column, we are able to examine the events with better context. In fact, all 5 clusters contain a CVE and four of the five refer to a specific version number for the vulnerable asset. This may not always be the case, however. Thus, there are instances in which entity clusters may need further investigation, which is discussed further in section 4.2. We were able to find corroborating articles for each of the clusters [43, 44, 45].

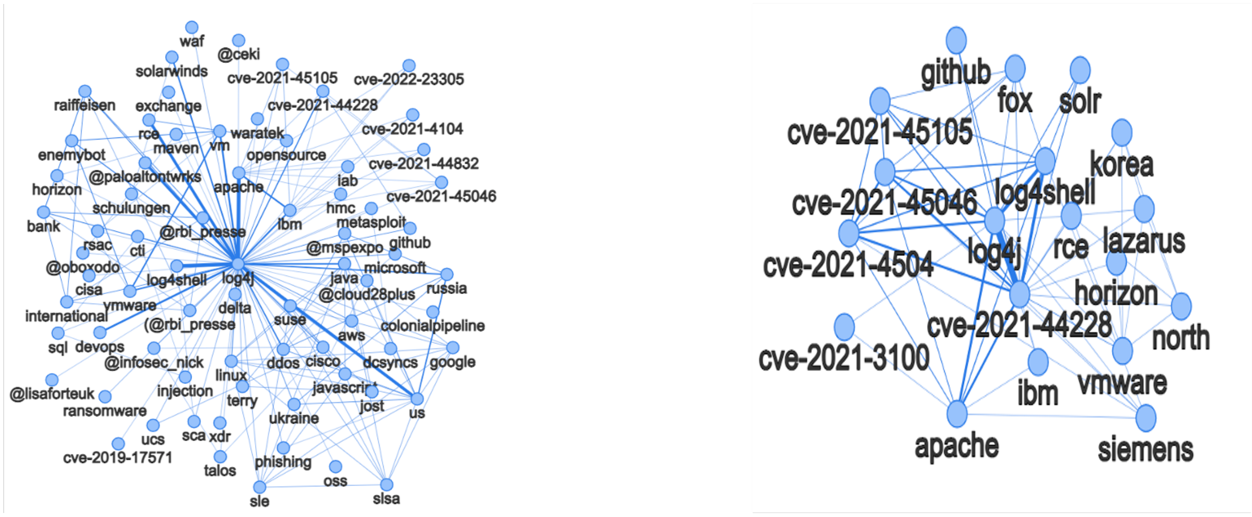
4.2 Cyber Event Investigation

Under some circumstances, the user may already know of an event and would like to investigate it. To simulate this use case, we focus on the "log4j" vulnerability that affected many companies in early 2022. The vulnerability was first disclosed in early December 2021 as a remote code execution (RCE) bug with a critical severity classification. Less than a day later, the vulnerability was being exploited by multiple threat actors, such as the Mirai botnets. Information like this is crucial for security operators to monitor their systems, especially considering Cloudflare and Cisco suffered attacks more than a week before the vulnerability was publicly disclosed [46]. With our proposed solution, professionals would have had access to information about this event and its implication, regardless of whether or not the provider had reported on it.

Table 4.1: Top five entity clusters with their associated outputs. Tweets are provided without modification.

Entity Cluster	Diff.	Spam	Sample Tweet
cve-2022-20821, xr, cisco	1.0	1.0	Cisco Warns of Exploitation Attempts Targeting New IOS XR Vulnerabi... (Securityweek) The flaw, tracked as CVE-2022-20821, was discovered by Cisco during the resolution of a s...
cve-2022-29599, maven, commandline	1.0	1.0	CVE-2022-29599 : In #Apache Maven maven-shared-utils prior to version 3.3.3, the Commandline class can emit double-quoted strings without proper escaping, allowing shell injection attacks....
cve-2021-30028 range wi-fi	1.0	1.0	Emerging Vulnerability Found CVE-2021-30028 - SOOTEWAY Wi-Fi Range Extender v1.5 was discovered to use default credentials (the admin password for the admin account) to access the TELNET service, allowing attackers to erase/read/write the firmwar
cve-2021-42863 jerryscript	0.7	0.583	Potentially Critical CVE Detected! CVE-2021-42863 A buffer overflow in `ecma_builtin_typedarray_prototype_filter()` in JerryScript version fe3a5c0 allows an attacker to con... CVSS: 8.80 #CVE #CyberSecurity
cve-2022-1816 zoo	1.0	1.0	CVE-2022-1816 A vulnerability, which was classified as problematic, has been found in Zoo Management System 1.0. Affected by this issue is `/zoo/admin/public_html/view_accounts?type=zookeeper` of the content module....

To begin our investigation we searched "log4j" and found 1017 tweets. Passing them through our cyber event detection model resulted in 487 tweets. An initial analysis is shown in Table 4.2. The word co-occurence network, provided in Figure 4.1a, shows a strong correlation between log4j, VMware, Apache, and RCE. There is also a number of CVE tags present in the network. To continue our investigation, we included CVE-2021-44228 in



(a) log4j

(b) log4j cve-2021-44228

Figure 4.1: Word Co-occurence networks for investigation search terms

the search parameters as it has the highest co-occurence with “log4j”. This small change dramatically reduced the network as shown in Figure 4.1b. For the final run we wanted to see if we could gather more information about what Lazarus may refer to. With the addition of this entity, the program was able to find a tweet containing all three entities, provided in Table 4.2. Of the entities found, North Korea was among the nation-states seen exploiting log4j [47], CVE-2021-44228 was the first of the CVEs revealed during this event [46], and Lazarus was a threat group targeting VMware servers [48]. In this process, many of the key entities of this event were identified in less than a minute.

4.3 Vulnerability Prioritization

In this case study, we adopted the perspective of a security operator who just became aware of two vulnerabilities in our company’s system, namely CVE-2022-26862 and CVE-2022-26717, but we did not know how to prioritize patching them. Our OSINT tool can provide intelligence that complements existing AI-based vulnerability management solutions [49, 50, 51, 52]. Specifically, we followed a similar methodology as section 4.2. After passing CVE-2022-26862 through our OSINT system, we found that the diffusion and spam indices of this topic are rather low, 0.6 and 0.467, meaning that the conversations about

Table 4.2: Diffusion and Spam Index Values from Example Entity Clusters. Tweets are provided without modification.

Entity Cluster	Diff.	Spam	Sample Tweet
log4j vmware us	0.923	0.885	#Log4Shell reminded us how important it is to have a trusted open-source software provider. This blog post by @mpermar explains the vulnerability in detail and how VMware Application Catalog brings confidence to developers and operators. `https://t.co/OhLawPexfJ` #Log4j #CVE
cve-2021-4422 lazarus north	1.0	1.0	North Korea-linked group Lazarus is exploiting the Log4J RCE #vulnerability (CVE-2021-44228) to compromise VMware Horizon servers. If not already do the right thing: Patch yours! #becybersmart `https://t.co/EYFNTcEfOz`

this vulnerability are fairly isolated. Our system also found two notable tweets showing that the vulnerability affects certain versions of Dell BIOS, and allows a locally authenticated malicious user to bypass security controls.

An initial run of our OSINT tool found a strong correlation between CVE-2022-26717 and “safari”. Once we included “safari” in our search terms, we found diffusion and spam indices of 1.0 and two notable tweets. The first provided a link to an exploit dated May 8th. The second also linked this exploit, but provided the patched version number as well.

We found CVE-2022-26717 the more dangerous vulnerability and should be prioritized as such. Not only were there fewer conversations about CVE-2022-26862, it also required a local user. This concludes our investigation. In fact, CVE-2022-26717 is much more widespread and has a publicly available exploit circulating. Thus, it indeed deserves a higher priority, validating our tools recommendation.

Chapter 5

OSINT for Network Protocol Identification

5.1 Overview

Overall, network protocols simply define the data formatting, sending, and receiving rules for a network endpoint. HTTP, TCP, and SMTP are examples of such protocols. However, they can also provide crucial information in the domain of vulnerability management, specifically whether or not the vulnerability is even exploitable. With the adoption of wireless networking, firewalls became commonplace. Broadly speaking, firewalls filter incoming/outgoing network traffic according to some criteria, typically via port. These ports are actually assigned to traffic via a specific protocol. Thus, by knowing what protocol a vulnerability relies on, security operators can better determine if their systems are at risk, even if a vulnerability is present.

In this section we detail the methodology and evaluations of an alternative method of extracting cybersecurity information from Twitter data. Though NER has yielded impressive results, both in this study and others, it may prove less effective than or need to be coupled with other methods, such as keyword identification, to achieve optimum performance. One such situation is the identification of network protocols. Keyword identification methods search a given string of text, such as a tweet, for a given sub-string, such as ”HTTP”. However, the protocols are not always mentioned. Thus, we built a mapping that will allow us to match certain products that are mentioned in a tweet to their associated protocols. In the following sections we describe the datasets used, the methodology of building the mapping, and the evaluations of the method with a comparison to NER performance.

5.2 Datasets

5.2.1 Protocol Mapping

To build this dataset, we employed the use of the national vulnerability database’s API. With this API, we pulled every CVE from January 15, 2019 to September 14, 2022. We then had nearly 130,000 records detailing different characteristics of the vulnerability such as status and published date, but most importantly we had the vulnerability's description. Next, we took a random sample of 50% of the CVEs, leaving us with 62,837. We went through a few pre-processing steps before proceeding. First, we converted the descriptions in to their lowercase forms. Next, we removed any from of website link through the use of a regular expression provided in equation 5.1. Following a similar methodology as in section 3.5, we leveraged the co-occurrence of a product and a protocol. By iterating through the CVE descriptions we were able to asses what products and protocols occured in the same CVE description, thus, insinuating a connection. The products were identified using the NER model described in section 3.4 and the protocols were identified by their keyword occurrence. These co-occurrences are then aggregated into a python dictionary mapping the lower-case string representation of the product, "axis m1125" for example, to a lower case string of its associated protocols separated by spaces such as "http ftp". It is worth noting the majority of the mappings only map products to a single protocol, but a few do contain multiple, which is possible. The resulting 567 mappings were then manually validated. A sample of these mappings are provided in table 5.1

$$\text{https?://\textbackslash S+} \tag{5.1}$$

Table 5.1: Sample of Product-to-Protocol Mapping

Product	Protocol	Product	Protocol
Zyxel armor	http	siplus s7-1500	tcp
zte psirt	telnet	http::daemon	http
advantech scada	http	vaadin designer	http
simatic et200s	udp	horde Groupware	http
apache wicket	dns	jabber	smtp
bacnetstac	ftp telnet	haproxy	http
snapdragon compute	tcp	kibana	http
adblock plus	http	phabricator	dns
somatom x.cite	tcp	openldap	dns
honeywell experion lx	tcp	totolink a3100r	telnet

Table 5.2: List of Protocols Searched for

Protocols
http, tcp, ftp, telnet, dns, arp, udp, smtp

5.2.2 Ground Truth

Before we can asses the efficacy of our approach, we need a high-quality ground truth to compare to. Once again, through the use of the NVD API, we pulled any CVE from 2019 to present whose description mentions one of the protocols provided in table 5.2. Following the pre-processing steps used in section 5.2.1, we lower-case standardized the descriptions and removed any external links. The label for these records is the network protocol mentioned in their descriptions. Before moving further, we validated the approach by taking a random sample of 100 of these records and manually analyzing them. We found that 90% of the CVEs analyzed were correctly associated to a protocol. Additionally, we applied the previously developed mapping to the CVE descriptions, effectively extending the number of protocols each CVE may have been mapped to.

5.3 Protocol Extraction via Tweet

The application of the product-to-protocol mapping and the keyword identification would likely occur within the NER phase discussed in section 3.4. By this point in the pipeline, the set of tweets will have already been filtered to include only those pertinent to cybersecurity. First, the tweet is lower-case standardized. Next, the tweet is searched for each of the 8 protocols mentioned in table 5.2. If a match is found, that protocol is added to the list of entities that need to be correlated. If no match is found, we continue with the NER model. From here, we check to see if any of the products found by the NER model are keys in the Product-to-protocol mapping. The protocols associated with any valid mappings are then added to the list of entities.

5.4 Evaluations

5.4.1 Metrics

For measuring the effectiveness of the approach, we use two primary metrics: coverage and accuracy. Coverage measures the prevalence of the terms of interest in the Twitter dataset. For instance, if we are investigating a vulnerability associated with HTTP, but none of the tweets actually contain the keyword ”HTTP”, we will have a coverage of 0. If the tweets do mention the protocol, however, we will have a coverage of 1. This value is averaged over the CVEs investigated. Lastly, we implemented accuracy. In this case, accuracy is the number of covered CVEs where we were able identify the correct protocol though the process described in section 5.3.

5.4.2 Comparison of Methods

By taking into account the products present within our Product-to-Protocol mapping, we found a coverage of approximately 77.7%, as opposed to 54.1% without the mapping. This means that of the CVEs we investigated, Twitter data contained some network protocol information for 77.7% of the CVEs. These tweets were then used to evaluate the accuracy of our method for those vulnerabilities. Overall, the Keyword/Mapping approach outperforms the NER model on a per-protocol basis and yields an average of 90.6% when weighted against the number of samples per-protocol. A complete breakdown of the performance of both the NER and Keyword/Mapping aproaches are provided in table 5.3

5.5 Approach Limitations

Though the Keyword/Mapping approach outperforms the NER model, there are still limitations to the approach. First, our tests focused on the identification of eight protocols, but in practice the list of protocols used to build the Product-to-Protocol mapping and the keyword search itself would need to be exhaustive. Considering the rate that techonology progresses and the computational complexity costs of adding new items to this list, it may

Table 5.3: Accuracy Comparison of Methods

Protocol	NER	Keyword & Mapping
http	46.0	88.3
ftp	57.1	97.0
smtp	78.9	95.8
telnet	82.8	94.3
tcp	55.7	97.4
arp	19.4	100.0
dns	60.0	96.4
udp	77.7	91.1
Weighted Average	50.1	90.6

not always be feasible. Secondly, keyword-based approaches typically have a high false-positive rate. For instance, one of the products we identified when building the mapping was called ClearPass. Through our qualitative investigations into what might be limiting the methods performance, we discovered that CVEs involving ClearPass were being associated with the protocol ARP without it ever being mentioned. We quickly realized that the program was identifying ARP as a substring of the product name: Cle-arP-ass. The simplest solution would be to append spaces to the beginning and end of the search terms so that the method only identifies protocols if they are stand alone words and in the middle of a token. However, we believe this ability to identify protocols mid-token may be why the keyword-based approach outperforms the NER. Take the product “httpdaemon” for instance. To human readers, it is understood that this product deals with the HTTP protocol and thus that may be the only mention of the protocol in the description at all. The tokenization rules of most NER models would leave this token as is and not be able to see it contains a protocol within. Keyword-based approaches however would. An empirical study would be needed to both asses both the false-positive rate of the approach as well as the benefit of allowing mid-token identification.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we proposed a scheme of discovering and analyzing cyber events through the use of OSINT based on Twitter data. Through a multi-model pipeline, our system can filter the Twitter stream to identify cyber event tweets, extract the valuable information about specific cyber events, and validate their veracity. The results of the project are promising and show that the pipeline could be used in practice to inform professionals during investigations. We also presented three case studies to demonstrate a few scenarios in which the system could be helpful. Lastly, we examined the potential of Twitter OSINT for the identification of network protocols. The results of this branch of work further support the use of Twitter for cybersecurity intelligence gathering.

6.2 Future Work

For future work, there are two primary directions we would like to take: usability and refinement of entities. In regard to usability, we would like to enhance the resulting co-occurence network produced at the end of the pipeline. As shown in figure 3.3, some clusters can become very convoluted, making it difficult for the users to evaluate the associated event. Further research into network analysis techniques for how to trim the network could remedy this. Additionally, the use of more user-friendly icons could help the user sift through the information. For instance, instead of every node being a blue circle, nodes denoting a CVE could be displayed as a yellow square and those denoting products could be green stars. For the refinement of our entities, the labels we used for the NER model are rather broad. Prodigy offers a train-curve function, which trains a model on 25% increments from 0% to 100% of a dataset. It allows the user to observe the plateau of the model's performance and in our case our model would likely benefit from further training. Future work should include diversifying the labels. For instance, the product label could be divided into operating systems, mobile devices, websites, databases, etc. This in-turn should help the model distinguish between organization and the many product classes as well as provide a better definition for the entities.

Bibliography

[1] Dakota Dale, Kylie McClanahan, and Qinghua Li. "AI-based Cyber Event OSINT via Twitter Data". In: *2023 International Conference on Computing, Networking and Communications (ICNC)*. 2023, pp. 436–442. DOI: 10.1109/ICNC57223.2023.10074187.

[2] Davey Winder. *Microsoft Security Shocker As 250 Million Customer Records Exposed Online*. Forbes.

[3] Gloria Gonzalez, Ben Lefebvre, and Eric Geller. *'Jugular' of the U.S. fuel pipeline system shuts down after cyberattack*. POLITICO.

[4] Julia E Sullivan and Dmitriy Kamensky. "How cyber-attacks in Ukraine show the vulnerability of the U.S. power grid". In: *Electr. J.* 30.3 (2017), pp. 30–35.

[5] *Cost of a Data Breach Report 2021*. 2021. URL: https://www.ibm.com/downloads/cas/OJDVQGRY.

[6] *2022 SonicWall Cyber Threat Report*. 2022.

[7] *The Red Report 2021*. Picus. Dec. 2021. URL: https://www.picussecurity.com/resource/blog/red-report-2021-top-ten-attack-techniques.

[8] Ido Kilovaty. "Cybersecuring the Pipeline". In: *Houston Law Review* 60 (2023). URL: https://ssrn.com/abstract=4070074.

[9] Marie Baezner. "Cyber and Information warfare in the Ukrainian conflict". In: *ETH Zurich* (2018).

[10] Dmytro Lande, Igor Subach, and Alexander Puchkov. "A system for analysis of big data from social media". In: *Information & Security* 47.1 (2020), pp. 44–61.

[11] Agata Ziółkowska. "Open Source Intelligence (OSINT) as an Element of Military Recon". In: *Security and Defence Quarterly* 2 (2018), pp. 65–77.

[12] Abdullah Talha Kabakus and Resul Kara. "A survey of spam detection methods on twitter". In: *International Journal of Advanced Computer Science and Applications* 8.3 (2017).

[13] Anna Sapienza, Sindhu Kiranmai Ernala, Alessandro Bessi, Kristina Lerman, and Emilio Ferrara. "DISCOVER: Mining Online Chatter for Emerging Cyber Threats". In: *The Web Conference 2018*. 2018, pp. 983–990.

[14] Ariel Rodriguez and Koji Okamura. "Social media data mining for proactive cyber defense". In: *Journal of Information Processing* 28 (2020), pp. 230–238.

[15] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. "CyberTwitter: Using Twitter to generate alerts for cybersecurity threats and vulnerabilities". In: *2016 IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining.*

[16] Avishek Bose, Vahid Behzadan, Carlos Aguirre, and William H Hsu. "A novel approach for detection and ranking of trendy and emerging cyber threat events in twitter streams". In: *Int'l Conf. on Advances in Social Networks Analysis and Mining.* 2019.

[17] Satyanarayan Raju Vadapalli, George Hsieh, and Kevin S Nauer. "Twitterosint: automated cybersecurity threat intelligence collection and analysis using twitter data". In: *International Conference on Security and Management (SAM).* 2018, pp. 220–226.

[18] Nuno Dionísio, Fernando Alves, Pedro M Ferreira, and Alysson Bessani. "Cyberthreat detection from twitter using deep neural networks". In: *2019 International Joint Conference on Neural Networks (IJCNN).* IEEE, pp. 1–8.

[19] Kanchan M.Tarwani and Swathi Edem. "Survey on Recurrent Neural Network in Natural Language Processing". In: *International Journal of Engineering Trends and Technology* 48 (2017), pp. 301–304.

[20] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. "Image Captioning with Semantic Attention". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition.* 2016, pp. 4651–4659.

[21] Jie Zhou and Wei Xu. "End-to-end learning of semantic role labeling using recurrent neural networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.* Vol. 1. 2015, pp. 1127–1137.

[22] Jaouhar Fattahi, Marwa Ziadia, and Mohamed Mejri. "Cyber Racism Detection Using Bidirectional Gated Recurrent Units and Word Embeddings". In: *New Trends in Intelligent Software Methodologies, Tools and Techniques.* IOS Press, 2021, pp. 155–165.

[23] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. "Comparative Study of CNN and RNN for Natural Language Processing". In: (2017).

[24] A Rehman Khan, A Tamoor Khan, Masood Salik, and Sunila Bakhsh. "An optimally configured HP-GRU model using hyperband for the control of wall following robot". In: *Int. J. Robot. Control Syst* 1.1 (2021), pp. 66–74.

[25] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. "Dipole: Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recur-

rent Neural Networks". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 1903–1911.

[26] Catherine Lee, Jacob Shiff, and Sridatta Thatipamala. *Predicting US Political Party Affiliation on Twitter*. Stanford University, 2018.

[27] Carlo Aliprandi, Antonio E. De Luca, Giulia Di Pietro, Matteo Raffaelli, Davide Gazzè, Mariantonietta N. La Polla, Andrea Marchetti, and Maurizio Tesconi. "CAPER: Crawling and analysing Facebook for intelligence purposes". In: *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*. 2014, pp. 665–669. DOI: 10.1109/ASONAM.2014.6921656.

[28] Miltiadis Kandias, Vasilis Stavrou, Nick Bozovic, and Dimitris Gritzalis. "Proactive Insider Threat Detection through Social Media: The YouTube Case". In: *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*. WPES '13. Berlin, Germany: Association for Computing Machinery, 2013, pp. 261–266. ISBN: 9781450324854. DOI: 10.1145/2517840.2517865. URL: https://doi.org/10.1145/2517840.2517865.

[29] Amir Karami, Vishal Shah, Reza Vaezi, and Amit Bansal. "Twitter speaks: A case of national disaster situational awareness". In: *Journal of Information Science* 46.3 (2020), pp. 313–324.

[30] Krishna Kanth A, Abirami S, Chitra P, and Gayathri Sowmya G. "Real Time Twitter Based Disaster Response System for Indian Scenarios". In: *2019 26th International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW)*. 2019, pp. 82–86. DOI: 10.1109/HiPCW.2019.00029.

[31] Christoph G Schmidt, David A Wuttke, George P Ball, and Hans Sebastian Heese. "Does social media elevate supply chain importance? An empirical examination of supply chain glitches, Twitter reactions, and stock market returns". In: *Journal of Operations Management* 66.6 (2020), pp. 646–669.

[32] Prabhsimran Singh, Yogesh K. Dwivedi, Karanjeet Singh Kahlon, Annie Pathania, and Ravinder Singh Sawhney. "Can twitter analytics predict election outcome? An insight from 2017 Punjab assembly elections". In: *Government Information Quarterly* 37.2 (2020), p. 101444. ISSN: 0740-624X. DOI: https://doi.org/10.1016/j.giq.2019.101444. URL: https://www.sciencedirect.com/science/article/pii/S0740624X19303521.

[33] João Rafael Gonçalves Evangelista, Renato José Sassi, Márcio Romero, and Domingos Napolitano. "Systematic Literature Review to Investigate the Application of Open Source Intelligence (OSINT) with Artificial Intelligence". In: *Journal of Applied Security Research* 16.3 (2021), pp. 345–369. DOI: 10.1080/19361610.2020.1761737.

eprint: https://doi.org/10.1080/19361610.2020.1761737. URL: https://doi.org/10.1080/19361610.2020.1761737.

[34] Chen Gao, Xuan Zhang, Mengting Han, and Hui Liu. "A review on cyber security named entity recognition". In: *Frontiers of Information Technology & Electronic Engineering* 22.9 (Sept. 2021), pp. 1153–1168.

[35] Otgonpurev Mendsaikhan, Hirokazu Hasegawa, Yukiko Yamaguchi, and Hajime Shimada. "Quantifying the Significance of Cybersecurity Text through Semantic Similarity and Named Entity Recognition." In: *ICISSP*. 2020, pp. 325–332.

[36] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N project report* 1.12 (2009).

[37] Natalie Friedrich, Timothy D Bowman, Wolfgang G Stock, and Stefanie Haustein. "Adapting sentiment analysis for tweets linking to scientific papers". In: *CoRR* abs/1507.01967 (2015).

[38] Sai Vikram Kolasani and Rida Assaf. "Predicting stock movement using sentiment analysis of Twitter feed with neural networks". In: *Journal of Data Analysis and Information Processing* 8.4 (2020), pp. 309–319.

[39] Dilara Torunoğlu, Gürkan Telseren, Özgün Sağtürk, and Murat C. Ganiz. "Wikipedia based semantic smoothing for twitter sentiment classification". In: *2013 IEEE INISTA*, pp. 1–5.

[40] Puti Cen. "Predicting Consumers' Brand Sentiment Using Text Analysis on Reddit". undergraduate. University of Pennsylvania, 2020.

[41] Dhwani Dholakia, Ankit Kalra, Bishnu Raman Misir, Uma Kanga, and Mitali Mukerji. "HLA-SPREAD: a natural language processing based resource for curating HLA association from PubMed abstracts". In: *BMC genomics* 23.1 (2022), pp. 1–14.

[42] T.K. Ashwin Kumar, Prashanth Kammarpally, and K.M. George. "Veracity of information in twitter data: A case study". In: *2016 International Conference on Big Data and Smart Computing*, pp. 129–136.

[43] Sergiu Gatlan. *Cisco urges admins to patch IOS XR zero-day exploited in attacks.* BleepingComputer. May 2022.

[44] *Security Bulletin 08 Jun 2022.* Singapore Computer Emergency Response Team. June 2022.

[45] *Security Bulletin 25 May 2022.* Singapore Computer Emergency Response Team. May 2022.

[46] Raphael Hiesgen, Marcin Nawrocki, Thomas C Schmidt, and Matthias Wählisch. “The Race to the Vulnerable: Measuring the Log4j Shell Incident”. In: (2022).

[47] *Hackers used the Log4j flaw to gain access before moving across a company’s network, say security researchers.* ZDNet.

[48] Pierluigi Paganini. *North Korea-linked Lazarus APT uses Log4J to target VMware servers.* Security Affairs.

[49] Philip Huff, Kylie McClanahan, Thao Le, and Qinghua Li. “A Recommender System for Tracking Vulnerabilities”. In: *Int’l Conf. on Availability, Reliability and Security (ARES).* ACM, 2021.

[50] Kylie McClanahan and Qinghua Li. “Automatically Locating Mitigation Information for Security Vulnerabilities”. In: *IEEE SmartGridComm.* 2020, pp. 1–7.

[51] Fengli Zhang, Philip Huff, Kylie McClanahan, and Qinghua Li. “A Machine Learning-based Approach for Automated Vulnerability Remediation Analysis”. In: *IEEE Conference on Communications and Network Security (CNS).* 2020, pp. 1–9.

[52] Philip Huff and Qinghua Li. “Towards Automated Assessment of Vulnerability Exposures in Security Operations”. In: *Int’l Conf. on Security and Privacy in Communication Networks (SecureComm).* 2021.