Knowledge Mining in Cybersecurity: From Attack to Defense

Khandakar Ashrafi Akbar[1], Sadaf Md Halim[1], Yibo Hu[1], Anoop Singhal[2], Latifur Khan[1], and Bhavani Thuraisingham[1]

[1] The University of Texas at Dallas, 800 West Campbell Road, Richardson, Texas 75080, USA

[2] National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA

Abstract. In the fast-evolving world of Cybersecurity, an analyst often has the difficult task of responding to new threats and attack campaigns within a limited amount of time. If an analyst fails to do so, this can lead to severe consequences for the system under attack. In this work, we are motivated to aid the security analyst by introducing a tool which will help to produce a swift and effective response to incoming threats. If an analyst identifies the nature of an incoming attack, our system can produce a ranked list of solutions for the analyst to quickly try out, saving both effort and time. Currently, the security analyst is typically left to manually produce a solution by consulting existing frameworks and knowledge bases, such as the ATT&CK and D3FEND frameworks by the MITRE Corporation. This task is made harder by the fact that existing knowledge bases are not always comprehensive, and so a lot of valuable security knowledge is instead found scattered across the web. To solve these challenges, our tool leverages existing frameworks as well as data crawled from the web. Our tool uses advanced natural language processing techniques, including a large language model (RoBERTa), to derive meaningful semantic associations between descriptions of offensive techniques and defensive countermeasures. Experimental results confirm that our proposed method can provide useful suggestions to the security analyst with good accuracy, especially in comparison to baseline approaches which fail to exhibit the semantic and contextual understanding necessary to make such associations.

Keywords: Cyber Threat Intelligence · Natural Language Processing · Semantic Association

1 Introduction

In the event of an attack, a security analyst has a small window within which to react and produce an effective counter-response. In this setting, time is of the essence. However, the entire process of analyzing the threat and determining the type of offense, and then coming up with the correct measure is often tedious and time consuming. This is compounded by the fact that knowledge in the cybersecurity domain is very scattered, and all the required information can rarely be found in one place. As such, the analyst will have to search across various resources including the web, all while the attacker is actively inflicting damage. Furthermore, the array of attack techniques is rapidly expanding so it

becomes even more difficult to process and respond in real-time. It is therefore imperative that we are able to speed up this process and assist the security analyst to minimize the harm as much as possible.

A security analyst has various resources at their disposal. One example is the National Vulnerability Database (NVD) [6], which provides detailed analysis of CVE [19] vulnerabilities, as well as exploitability and impact scores for those vulnerabilities. This tool is useful for understanding what kind of vulnerabilities might be exploited in the event of an attack. However, should an attack occur, the analyst needs to not only understand the vulnerabilities being exploited, but also be ready to produce an effective counter-response to that attack. Cyber attacks like SolarWinds [14] are inevitable in certain situations when supply chain compromises cannot be handled well. In these cases, root-level system analytics or countermeasures are necessary to find out and solve particular attack tactics or techniques which are possibly being performed behind the scenes, in order to prevent any significant aftermath. Another existing resource for security analysts is a knowledge-base built by the MITRE corporation. In fact, MITRE offers two different frameworks: ATT&CK [20] and D3FEND [21]. ATT&CK is a framework that offers knowledge regarding certain offensive techniques, and D3FEND offers defensive techniques and countermeasures. Both of these offensive and defensive frameworks are hierarchically structured almost the same way: they consist of tactics at the top level and of techniques and sub-techniques at the bottom level. Techniques are composed of sub-techniques (if any) and each technique belongs to a tactic or multiple tactics in the hierarchy. All these tactics, techniques, and sub-techniques have their respective textual descriptions which are presented either from the behavioral or the technical perspective. D3FEND was generated based only on patent information in combination with a few other external resources such as the Cyber Analytic Repository (CAR).

In the current scenario, a security analyst would have to look up resources such as the knowledge bases offered by MITRE, as well as the web, in order to figure out both the nature of the attack being conducted, as well as a potential solution for the attack. This is evidently slow, and given the vast possibilities for attack types and countermeasures, this entire process is akin to finding a needle in a haystack, under severe time constraints. This inevitably increases the chance of extensive harm being done. Furthermore, existing frameworks such as D3FEND [21] provide associations of defensive techniques and offensive techniques, but these knowledge bases are manually generated, with enormous human effort involved. Thus, if new defensive techniques are to be associated with already existing attack techniques or zero-day attack techniques (and vice versa) we will need to manually perform associations in the absence of any automated solutions. Also, the D3FEND framework mainly only provides defensive techniques - which typically provides analytical solutions that help us to identify attacks of a specific type. It does not usually provide countermeasures which will actually mitigate the attack or put an end to the ongoing attack campaign.

It is thus clear that the security analysts' toolkit is presently missing an important tool - a comprehensive and automated system which can quickly provide

the analyst with relevant solutions in the event of an attack. We are therefore motivated to fill this gap and create this tool. But this is not without its own challenges. First, creating this tool using current knowledge bases is not straightforward. Since attack and defense methods are constantly evolving, it is evident that the knowledge bases provided by organizations such as MITRE are not exhaustive. Therefore, to fill this gap and make the system as comprehensive as possible, we must turn to the world wide web. This involves crawling the web and scraping relevant websites to gather as much information as we can regarding attacks, defenses and countermeasures. Much of this information is unstructured text, which must then be processed to extract knowledge and then inform our system. Yet another challenge is filtering out erroneous information and the "junk" which we inevitably come across when crawling the web.

In this paper, we present a tool that overcomes these challenges and automatically identifies potential countermeasures and makes recommendations to the analyst. The recommendations are made by identifying meaningful associations between the ongoing attack tactic or technique and the candidate countermeasure(s). These associations are found through the aid of advanced machine learning and natural language processing techniques. We use the language model called RoBERTa (Robustly Optimized BERT Pre-training Approach) [16], which we show to be similarly effective to classical techniques such as bag of words or word2vec [17].

Typical topic modeling might fail in establishing such associations since the dataset in this particular task is limited and unstructured. Moreover, capturing semantics from textual descriptions can be tricky with traditional models. Instead, we use the RoBERTa model which is pre-trained efficiently with a large number of tokens (leading to a bigger vocabulary size) and which can aid zero-shot learning [22], since this model is trained with a large amount of data nearing 160GB of uncompressed text. Zero-shot learning is a type of machine learning technique where the model is used without fine-tuning on a particular task. Thus, models like RoBERTa which are ready-to-use can generate effective representations of texts in order to find semantic similarities among them.

We now present an example use-case for our tool. Our tool works as follows: Suppose the security analyst identifies an offensive technique used to perform an attack, such as "Spearphishing Link". The security analyst can query our tool to find recommendations on what defensive techniques or countermeasures to take. Our tool will provide a ranked list of defensive techniques and countermeasures such as "Homoglyph Detection, File-Hashing, File Carving, Process Spawn Analysis". In this hypothetical list, the recommendations are ranked in descending order of priority. The analyst can then quickly attempt these solutions in that order. We can thus see how this system can largely relive the security analyst of the tedium as well as the stress of having to manually figure out how to respond to a critical system attack. This in turn minimizes the harm to the system by producing a solution faster.

In this work, our contributions can be summarized by the following:

- The creation of a tool which recommends appropriate countermeasures and defensive techniques to the security analyst when queried with an attack technique.
- The automation of associations between offensive and defensive techniques and countermeasures using language models.

This paper is organized as follows. In Section 2, we detail the relevant preliminaries such as the D3FEND and ATT&CK frameworks by MITRE. In Section 3, we formally present the problem statement and provide a relevant case study. In Section 4, we describe our approach in detail. Experimental results are presented in Section 5. In Section 6, we provide a study on related works. Lastly, Section 7 provides the conclusion for this study.

2 Preliminaries

In the past few years, the knowledge of defensive countermeasures as well as the knowledge of adversarial behaviors have evolved. Post-compromise adversarial behavioral analytics are now provided by frameworks like ATT&CK [20] . Defensive countermeasures [21] for different offensive techniques have also been developed and provided through knowledge bases which have been manually built up by MITRE. In the following two subsections, we will discuss briefly about these two types of frameworks.

2.1 Offense: ATT&CK

The offense framework that we use here is called ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge). Generated by MITRE, this framework contains offensive tactics, techniques and sub-techniques. Some of the techniques or sub-techniques can belong to multiple tactics. Figure 1 is the partial illustration of the enterprise matrix of this framework. The entries in the first row (e.g., reconnaissance, resource development, initial access, and execution) define the different tactics. Furthermore, the entries under the same column for each of the tactics define the techniques or sub-techniques which belong to that particular tactic. There are 14 different tactics in this framework which have numerous techniques and sub-techniques under their hood.

2.2 Defense: D3FEND

The defense framework (D3FEND [21]) has certain defensive countermeasures in the form of tactics, techniques and sub-techniques. D3FEND is generated based only on patent information and few other external resources (e.g., Cyber Analytic Repository – CAR). Existing enterprise or web-resource based cybersecurity solutions are out of scope for D3FEND. Figure 2 is the partial pictorial depiction of this particular framework along with its components. The entries in the first row are the tactics, and the entries in the second row are the techniques. Entries under a certain column are sub-techniques belonging to a particular technique and tactic in the upper hierarchy. A single defensive technique or sub-technique can be a countermeasure for different offensive techniques or sub-techniques.

2.3 Association of Offensive and Defensive Techniques

D3FEND framework has certain defensive technique(s) associated with certain offensive techniques which are present in the ATT&CK framework. Compiling information from both the ATT&CK and D3FEND framework can tell us which

defensive technique(s) can be used for the analytics of a particular offensive technique or as its countermeasure.
Thus, associations for offensive and defensive techniques exist through the D3FEND framework by MITRE though this framework is not comprehensive enough to provide all possible associations. A couple of examples of this type of association are discussed as follows:

Association Between Offense and Defense: The offensive sub-technique 'PowerShell Profile' (T1546.013) belongs to the technique 'Event Triggered Execution' in the ATT@CK framework and the following defensive techniques or sub-techniques are associated with it:

- Dynamic Analysis
- Emulated File Analysis
- File Content Rules
- File Hashing
- Executable Denylisting
- Decoy File

These offensive techniques have been assessed to be related to the particular defensive technique based on some digital artifact relationships using some digital artifact objects (DAO). These mappings or associations are inferred, i.e., they are experimentally established from the DAOs and so they will improve as the knowledge graph for D3FEND grows. An example of how DAO connects the offensive and defensive techniques are given below:
The knowledge extracted from D3FEND framework relating to this concept is given in figure 3. In figure 3, the entity enclosed within the blue box is the defensive sub-technique, the entities enclosed within the red boxes are offensive techniques or sub-techniques, and the entities enclosed within the off-white boxes are the digital artifact objects (DAOs). The offensive and defensive techniques or sub-techniques are connected using some relationships which are labeled accordingly. 'Application Configuration Hardening' is a defensive sub-technique which is connected to a digital artifact object (DAO) 'Application Configuration' via a digital artifact relationship 'hardens'. The offensive techniques or sub-techniques such as 'Email Collection', 'Email Forwarding Rule', etc are connected to the same digital artifact object (DAO) 'Application Configuration' or the objects belonging to the same object class as this (e.g., Application Rule, Email Rule, Process Environment Variable) using some other digital artifact relationships such as: 'modifies','may-modify','may-create'. This is how the bridging or association takes place among the offensive and defensive techniques using DAOs.

3 Problem Statement

3.1 A Dive into Statistics: Offense and Defense

A security analyst needs to have the appropriate knowledge or a framework handy in the event of an attack, so that they can quickly respond with defensive measures before any persistent damage is done. The D3FEND framework is generated manually (knowledge extraction with human intervention). Plenty of offensive and defensive techniques evolve everyday, and so it is quite impossible to associate them quickly and effectively (to form meaningful associations). Existing

Reconnaissance	Resource Development	Initial Access	Execution
Active Scanning	Acquire Infrastructure	Drive-by Compromise	Command and Scripting Interpreter
Gather Victim Host Information	Compromise Accounts	Exploit Public-Facing Application	Container Administration Command
Gather Victim Identity Information	Compromise Infrastructure	External Remote Services	Container Administration Command
Gather Victim Network Information	Develop Capabilities	Hardware Additions	Deploy Container
Gather Victim Org Information	Establish Accounts	Phishing	Exploitation for Client Execution
Phishing for Information	Obtain Capabilities	Replication Through Removable Media	Inter-Process Communication
Search Closed Sources	Stage Capabilities	Supply Chain Compromise	Native API
Search Open Technical Databases		Trusted Relationship	Scheduled Task/Job
Search Open Websites/Domains		Valid Accounts	Shared Modules
Search Victim-Owned Websites			Software Deployment Tools
			System Services
			User Execution
			WIndows Management Instrumentation

Fig. 1: Framework for Offensive Techniques (Partial)

Isolate		Deceive		Evict	
Execution Isolation	**Network Isolation**	**Decoy Environment**	**Decoy Object**	**Credential Eviction**	**Process Eviction**
Executable Allowlisting	Broadcast Domain Isolation	Connected Honeynet	Decoy File	Account Looking	Process Termination
Executable Denylisting	DNS Allowlisting	Integrated Honeynet	Decoy Network Resource	Authentication Cache Invalidation	
Hardware-based Process Isolation	DNS Denylisting	Standalone Honeynet	Decoy Persona		
IO Port Restriction	Forward Resolution Domain Denylisting		Decoy Public Release		
Kernel-based Process Isolation	Hierarchical Domain Denylisting		Decoy Session Token		
Mandatory Access Control	Homoglyph Denylisting		Decoy User Credential		
System Call Filtering	Forward Resolution IP Denylisting				
	Encrypted Tunnels				
	Inbound Traffic Filtering				
	Outbound Traffic Filtering				

Fig. 2: Framework for Defensive Techniques (Partial)

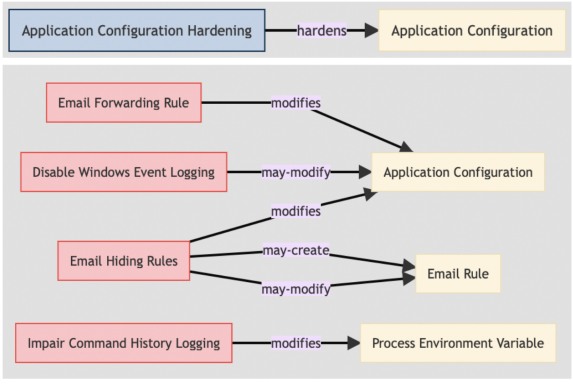


Fig. 3: Association Between Offensive and Defensive Techniques

cybersecurity solutions cannot be used effectively if a security analyst doesn't know when and how to use them.

To remedy these drawbacks of manual knowledge association, we want to semi-automate this process. We will have to investigate some statistics and other facts to emphasize on the necessity of building a language model for associating defensive countermeasures to offensive techniques. These statistics mainly shed light on why we need external resources for this association. These statistics are given in the following two tables: 1a and 1b.

We will use the textual descriptions of the offensive techniques or sub-techniques and their associated defensive techniques or sub-techniques to associate and rank all the defensive techniques or sub-techniques. For this task, we use language models (e.g., BERT [8]) as they can be successfully used to measure semantic similarities among different textual descriptions. Models like Word2Vec [18] and TF-IDF [27] will be used as baselines.

There are at least 259 offensive techniques or sub-techniques which do have at least a single associated defensive countermeasure. On average, a single offensive technique or sub-technique has 3-4 defensive techniques or sub-techniques as countermeasures. In table 1, max and min count indicates the maximum and minimum number of associations that a single offensive or defensive technique or sub-technique has. A count of zero for the number of associations indicates that an offensive technique or sub-technique have no associated countermeasure(s) within the 'D3FEND' framework, and we discovered that there are many such cases. Specifically, there are 287 offensive techniques or sub-techniques have no association which is more than half of the whole set. This tells us that the D3FEND framework is clearly not comprehensive enough. To fill this gap, we need to utilize resources from the web, so that effective countermeasures can be well-associated with the offensive techniques or sub-techniques which may or may not have associations within the D3FEND framework. We provide a case study in the upcoming subsection on how these external resources are effective and suitable for use in creating this association.

Total Offensive Techniques	546
Max Count	28
Min Count	0
Average	3.341
Count of Zero	287

(a) Statistic for Offensive Techniques

Total Defensive Techniques	123
Max Count	173
Min Count	0
Average	23.195
Count of Zero	50

(b) Statistic for Defensive Techniques

Table 1: A Dive into Statistics

3.2 Case Study on Web Data Resources

Following from the discussion on the ATT&CK and D3FEND frameworks statistics, finding associations for the zero-count offensive techniques can be compared to finding a needle in the haystack. To fill this gap, we need external resources which can provide the defensive countermeasures that we will associate with the offensive techniques and sub-techniques.

We need web data resources to have more counteractive solutions for associating them with the offensive techniques as the D3FEND framework is not

comprehensive enough. We crawl cybersecurity product descriptions, solutions from different websites from the web as part of different counteractive solutions. To emphasize on how these crawled data is effective for establishing the associations, we present two case studies as follows:

Example 1 Botnet (denoted as T1583.005 in the ATT&CK framework) is an offensive sub-technique which does not have any relevant defensive countermeasures, at least within the 'D3FEND' framework. Botnet detection is an important part of protecting from 'Phishing' attacks from seemingly legitimate sources. Akamai Technologies provide solutions for 'Botnet Detection' for which they have product briefs as well. This [2] web link contains the product brief for the bot manager from Akamai. We can generate our own corpus (dataset), choosing product descriptions wisely to get countermeasure solutions for such offensive techniques or sub-techniques which have no associations within available frameworks such as D3FEND.

Example 2 Cron (denoted as T1053.003 in the ATT&CK framework) is another offensive sub-technique which can be used to execute programs at system startup or on a scheduled basis for persistence. To stop a cronjob once it has been started, crontab must be edited, removing the line that triggers the job, and then saving the file. Multiple stack-exchange or superuser solutions show how to do it in Linux based systems [31]. This sort of information exists in the web and can be extracted to be used as part of counteractive solutions to different offensive techniques or sub-techniques. A security analyst can easily follow the steps from those solutions and take care of the situation whenever some scheduled adversarial program is set to execute in a system.

3.3 Challenges for Recommendation

A security analyst needs to be directed towards the right path of either performing some kind of analytics or taking any defensive measures which stops an ongoing adversarial campaign. But this sort of initiative requires prompt attention so that proper action can be taken before any permanent damage can take place. Thus, proper attack information should be associated with proper defensive countermeasures so that the exhaustive search for the security analyst can be reduced significantly. For example, if a security analyst has to iterate over a bunch of solutions to find the solution to a particular problem, the damage will be done long before the security analyst engages in the actual work. And it is almost impossible for a security analyst to know the solution to every problem right away given the fast expansion of attack techniques.
Thus, if security analysts are given recommendations on which solutions to try and in which order, quick responses can be ensured. But associating meaningful defensive countermeasures is challenging as the textual descriptions for the defensive countermeasures are not always technically sound. A particular offensive technique can have multiple solutions, each of them being effective in different degrees. A security analyst should know in which order he should try out those countermeasures so that he can stop the ongoing adversarial campaign as early as possible. Structural knowledge is required for such ranked associations. It is

also challenging to always have structural knowledge available for such associations as knowledge-graphs are often constructed manually. Thus, the structural knowledge can be missing due to the tedious processing required, or it can also take a lot of time to be built.

4 Approach

To understand the choices we made for our approach, we must first explore certain baseline approaches and identify their limitations. There are a number of ways to find matches among textual descriptions. We therefore first discuss some of the most widely used techniques.

4.1 Common Techniques

TF-IDF Term Frequency–Inverse Document Frequency, commonly known as TF-IDF, is a statistical measure of how important a specific word or token is to a piece of text, which we can refer to as a *document*. TF-IDF can be used to represent documents as vectors and then the similarity between the two documents can be calculated. Similar to Bag-of-Words, TF-IDF operates based on the counts of words. This means that learning is often "shallow", with little understanding of the actual semantics or the context within which a word is being used. This limits the utility of TF-IDF in applications like ours, where understanding the semantics of the textual descriptions is of paramount importance.

Word2Vec Word embeddings are also very popular. Word2Vec [17], developed at Google, is one of the most popular methods for learning high quality word embeddings. It does so by employing a shallow neural network. Essentially, Word2Vec learns word associations from a large corpus of text. While Word2Vec has proved effective for a number of tasks, it is still not without its own challenges. Word2Vec does not handle out-of-vocabulary words well, which can happen in many cases. Furthermore, Word2Vec embeddings are *context independent*. A word can have multiple meanings depending on the context in which it is used. However, Word2Vec combines all of these different *senses* of the word into one overall embedding. This is a clear limitation which is not present in newer transformer based language models like BERT [8], which can have multiple vector representations of the same word, depending on the context. Lastly, large scale language models, which we will discuss next, are simply known for being able to capture a deeper understanding of the semantics in text.

Language Models This brings us to Transformer-based language models, which have revolutionized many areas of natural language processing. BERT, released by Google, is one such model which achieved state of the art performance on a variety of natural understanding tasks when it was first published. Since then, BERT has become a ubiquitous baseline for a wide range of natural language tasks. An important enhancement on BERT, called RoBERTa [16], was released by Facebook AI. RoBERTa changed both the pretraining objective as well as made adjustments to the hyperparameters, both of which contributed to state of the art performance in a wide range of tasks. RoBERTa is trained on a huge corpus (measured at 160 gigabytes) and it is known for its ability to understand semantics and context at a level that was previously not possible. This is what motivated us to apply RoBERTa to our specific problem setting.

4.2 Proposed Method

Our approach is summarized in Figure 4. There are two distinct parts in our approach. The first part deals with the major task at hand - matching attack techniques with defensive techniques and countermeasures. To do this, we take advantage of the ATT&CK and D3FEND frameworks that we have described earlier. In Figure 4, this part of the approach is represented by the boxes with green text. First, from the ATT&CK framework, we extract the textual descriptions of our attack techniques and tactics. At the same time, we extract textual descriptions of the defensive techniques from the D3FEND framework. Once we have both sets of textual descriptions, we then proceed to the next step where we use RoBERTa. As discussed earlier, RoBERTa is already pretrained on an enormous dataset, and we leverage the deep semantic knowledge present inside the standard RoBERTa model to derive meaningful associations between attack techniques and defensive countermeasures. RoBERTa investigates the textual descriptions of the attack and defense techniques, and provides us a ranked list of D3FEND-based countermeasures for each attack technique in ATT&CK. Note that we do not fine-tune RoBERTa on our dataset of textual descriptions. This is because our dataset is much too small for the model to meaningfully learn from, and it might in fact affect the original RoBERTa model's ability to understand semantics and context. Instead, we directly leverage the deep semantic understanding that the standard RoBERTa model is equipped with, to give us meaningful associations between the descriptions that we have extracted.

The second part of our approach tries to deal with a crucial problem: the fact that current frameworks are not comprehensive. This part of the approach is represented by the boxes with blue text in Figure 4. Attack and defense methods are constantly evolving, which inevitably means that frameworks like D3FEND cannot be exhaustive. To remedy this issue, we turn to the web. The web has a vast range of resources and data, and the challenge here lies in identifying the information which is actually useful. One naive way to go about this might be to run a google search with the keywords relevant to the attack, followed by terms like "mitigation" or "prevention". However, search engines will do a keyword match and return a huge number of results, not all of which might be relevant. Using a straightforward method like this might in fact harm the precision of the overall approach by introducing too many *junk* suggestions.

Instead, our approach is to first crawl the web, and then to extract paragraphs that contain the attack mitigation techniques or countermeasures. Next, these countermeasures collected from the web are again sent to RoBERTa to produce a ranked list for the given attack type. This gives us an alternative ranked-list based on methods collected from the web. This list is important because it might suggest to the security analyst an appropriate countermeasure which may not be available in the D3FEND framework. As attack and defense techniques constantly evolve, we believe it is crucial for any system to evolve with it. Crawling the web and providing these solutions on-the-fly helps keep our system up to date, especially when frameworks like D3FEND might not cover them fast enough, since they involve a lot of manual effort.

In this way, we have a developed a two-part approach which leverages a large pre-trained language model like RoBERTa to generate two different ranked-lists for countermeasures: one created from the D3FEND framework and one gathered from the web.

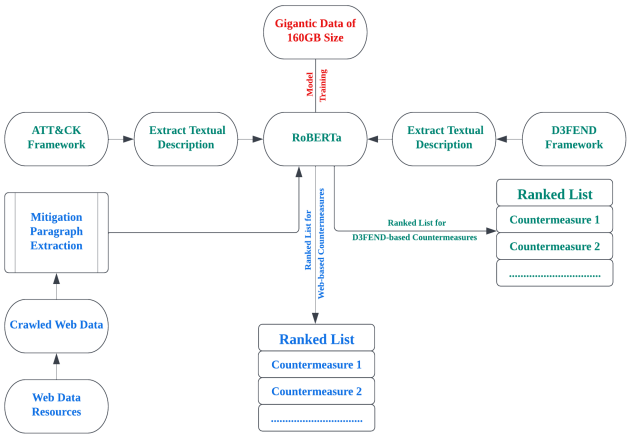


Fig. 4: Work Flow Diagram

5 Experimental Results

5.1 Experimental Setup

We have used the textual descriptions of the offensive and defensive techniques and sub-techniques from the ATT&CK and D3FEND framework as part of our dataset. For the web data resources, we have crawled web data after performing keyword based search for locating the resources within the web. Crawling of web resources are done using the following APIs: justext, newspaper, and trafilatura and we have taken the descriptions with the largest length, amongst the ones that we consider. The RoBERTa model is used from hugging face [9] along with the tokenizer. For Word2Vec, which is a baseline in our study, the glove (Global Vectors for Word Representation) [10, 26] vectors are used for word representations. Pre-trained word vectors with the following characteristics are used: 6B tokens, 400K vocab, uncased, 300 dimensional vectors. The embedding for a whole textual description is calculated by taking the average of the embeddings of all the words belonging to that description. For TF-IDF based experimentation, we have considered each of the textual description as a document and have generated the vectors considering all those descriptions (documents) as the corpus.

5.2 Results

The experimentation is done in two phases. The results of the first phase are tabulated in table 2 and 3 and pictorially depicted in figure 5 and 6.

Precision, recall, and f1-score are reported for RoBERTa-base along with the other baselines. All these three metrics are measured in two different settings, one while considering all possible offensive techniques and sub-techniques and

the other while considering the techniques only (excluding the sub-techniques). RoBERTa-base outperforms the baseline in both these settings. As baselines, we have used Word2Vec [18] and TF-IDF [27] models. Precision, recall and f1-score are reported for the following criteria: top_3, top_5, top_7, and top_10. For all the models including the baselines, it is clear that the RoBERTa base model performs the best in terms of all three metrics.

In Figure 5 and Figure 6, the x axis represents the n value from *top_n* and the y axis represents the precision for part a, recall for part b, and F1-score for part c. Figure 5 has the precision, recall, and f1-score plot while considering the offensive techniques only. Figure 6 has the precision, recall, and f1-score plot while considering all the offensive techniques and sub-techniques. The graphs also support the idea that the our approach using RoBERTa, colored red, outperforms the baselines at every *top_n* value. For example, for top_3 results, we see that RoBERTa is able to produce precision, recall and F1-scores of 0.39, 0.13 and 0.20 respectively. By comparison, for top_3, TF-IDF produces results of 0.25, 0.08 and 0.13 respectively for the same metrics. Word2Vec manages scores of 0.10, 0.03 and 0.05. Evidently, both sets of results trail significantly behind our approach, and this pattern repeats for the other top_n experiment settings.

These results are significant since this is not merely a binary classification where a random guess (50% accuracy, precision or recall) might be enough. The association of offensive techniques to defensive techniques is one to many ranging from 3 (min count) to 28 (max count). There are in total 73 defensive techniques which are associated with the offensive techniques. There are 259 techniques and sub-techniques which are associated with at least a single defensive technique among which 88 of them are techniques only.

		Precision	Recall	F1-Score
RoBERTa Base	top_3	**0.39**	**0.13**	**0.20**
	top_5	**0.35**	**0.20**	**0.26**
	top_7	**0.35**	**0.28**	**0.31**
	top_10	**0.30**	**0.34**	**0.32**
		Precision	Recall	F1-Score
Word2Vec	top_3	0.10	0.03	0.05
	top_5	0.10	0.06	0.07
	top_7	0.12	0.09	0.10
	top_10	0.12	0.14	0.13
		Precision	Recall	F1-Score
TF-IDF	top_3	0.25	0.08	0.13
	top_5	0.22	0.12	0.16
	top_7	0.20	0.16	0.18
	top_10	0.17	0.19	0.18

Table 2: Precision, Recall and F1-Score - For Techniques Only

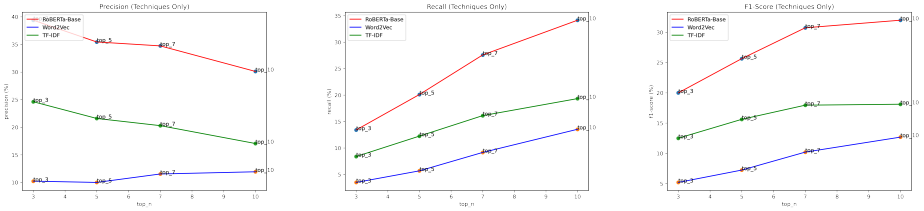
		Precision	Recall	F1-Score
RoBERTa Base	top_3	**0.31**	**0.13**	**0.18**
	top_5	**0.29**	**0.21**	**0.24**
	top_7	**0.28**	**0.28**	**0.28**
	top_10	**0.25**	**0.35**	**0.29**
		Precision	Recall	F1-Score
Word2Vec	top_3	0.09	0.04	0.05
	top_5	0.08	0.06	0.07
	top_7	0.09	0.09	0.09
	top_10	0.08	0.12	0.10
		Precision	Recall	F1-Score
TF-IDF	top_3	0.19	0.08	0.11
	top_5	0.17	0.12	0.14
	top_7	0.16	0.16	0.16
	top_10	0.14	0.2	0.17

Table 3: Precision, Recall and F1-Score - For Techniques and Sub-Techniques

Table 4: Precision, Recall, and F1-score for the whole ontology

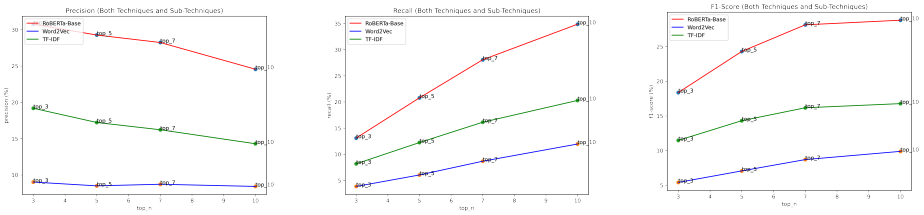
5.3 Results With Pruned Ontology

We have done another set of experiments with a pruned ontology of the D3FEND framework. In this special experimental setup, we have considered only the relevant defensive tactics from the D3FEND framework. This narrows down the associated techniques for an offensive technique or sub-technique which fall under the relevant defensive tactics only. For any offensive technique or sub-technique the possible list of associations are brought down from the count of 73 (all existing defensive techniques in D3FEND framework) to a certain number depending



(a) Precision for Techniques Only (b) Recall for Techniques Only (c) F1-Score for Techniques Only

Fig. 5: Evaluation Metrics for Techniques Only



(a) Precision for Both Techniques and Sub-Techniques (b) Precision for Both Techniques and Sub-Techniques (c) Precision for Both Techniques and Sub-Techniques

Fig. 6: Evaluation Metrics for both Techniques and Sub-Techniques

on the defensive tactics which they can be associated to. For example, the offensive technique "Spearphishing Attachment" can only have associations under the hood of the following defensive tactics according to the D3FEND framework: Network Traffic Analysis, File Analysis, Network Isolation, Message Analysis, Identifier Analysis, User Behavior Analysis, Decoy Object instead of all the 17 different defensive tactics in the D3FEND framework. This way we narrow down the search scope which eventually leads to better suggestions by producing an improved and more relevant ranked list. This is validated through the tabulated results listed in table 5 and table 6.

		Precision	Recall	F1-Score
RoBERTa Base	top_3	**0.60**	**0.26**	**0.36**
	top_5	**0.54**	**0.38**	**0.45**
	top_7	**0.47**	**0.47**	**0.47**
	top_10	**0.41**	**0.59**	**0.48**
		Precision	Recall	F1-Score
Word2Vec	top_3	0.28	0.12	0.17
	top_5	0.27	0.19	0.22
	top_7	0.28	0.27	0.27
	top_10	0.28	0.38	0.32
		Precision	Recall	F1-Score
TF-IDF	top_3	0.39	0.17	0.24
	top_5	0.36	0.26	0.30
	top_7	0.35	0.35	0.35
	top_10	0.32	0.45	0.37

Table 5: Precision, Recall, and F1-Score - For Techniques and Sub-Techniques

		Precision	Recall	F1-Score
RoBERTa Base	top_3	**0.65**	**0.22**	**0.33**
	top_5	**0.56**	**0.32**	**0.41**
	top_7	**0.50**	**0.40**	**0.44**
	top_10	**0.46**	**0.52**	**0.49**
		Precision	Recall	F1-Score
Word2Vec	top_3	0.31	0.10	0.15
	top_5	0.28	0.16	0.20
	top_7	0.29	0.23	0.26
	top_10	0.30	0.34	0.32
		Precision	Recall	F1-Score
TF-IDF	top_3	0.44	0.15	0.22
	top_5	0.4	0.23	0.29
	top_7	0.39	0.31	0.35
	top_10	0.36	0.41	0.38

Table 6: Precision, Recall, and F1-Score - For Techniques Only

Table 7: Precision, Recall, and F1-score for pruned ontology

Model	Count of Non-Associated Techniques or Sub-Techniques	Precision
RoBERTa-base	5	**0.56**
Word2Vec	5	0.38
TF-IDF	5	0.34

Table 8: Precision for the web crawled resources

5.4 Results With Web Crawled Resources

For this part of our experiment, we use the resources crawled from the web to produce defensive countermeasures. At first, we have generated a list of the offensive techniques or sub-techniques from the ATT&CK framework which have no known associations. Next, we create and inspect the top 10 list from the web-crawled resources, which is generated by sorting by the similarity scores for each of those resources with the offensive technique or sub-technique's description. We present the precision that was achieved in table 8. The recall cannot be calculated in this scenario as we do not know the exact association lists for these offensive techniques or sub-techniques.

5.5 Use Case

Certain iterative projects, e.g. [30], exist for finding cyber threats with ATT&CK-based analytics. Behavioral identification, or the detection of anomalies and outliers, is the key part to identifying cyber-threats within a system or network according to this analytics method. Methodologies based on system telemetry or network data inspection [4, 34] can identify certain ongoing offensive campaigns within a system and define the tactics or techniques which are being used to execute the campaign. But as soon as an anomalous behavior or particular vulnerability is spotted, a prompt and proper reaction is necessary from the defensive end. If appropriate actions are suggested, relevant measures can be taken in time to battle the cyber-threat. The use-case of our contribution in this paper applies to a scenario such as the following:

A security analyst might find that a vulnerability has been detected, or they may want to identify an ongoing attack campaign within a system. To do so, the analyst has to perform certain analytics. Knowing what analytics needs to be performed for a specific scenario requires a lot of domain knowledge or iterative search. For example, an adversary might communicate using the Domain Name System (DNS) application layer protocol to avoid detection or being filtered at the network level by blending in with existing traffic. Commands sent to a remote system and the results of those commands can be embedded with the protocol traffic between the client and server. This sort of technique can be used to maintain an established Command and Control (C&C) server. This phenomenon can be identified using certain analytics based approaches such as: protocol metadata anomaly detection, remote terminal session detection, client-server payload profiling, etc. The analyst can use such tools to identify the nature of the attack. Next, in this context, our tool helps the security analyst specifically because of the following: the particular offensive technique we are concerned with, i.e., 'Domain Name System (DNS)' is associated with 18 different defensive countermeasures of the D3FEND framework. Using our tool, when we sort the ranked list of all possible defensive solutions (a count of 73), the top 10 from

that sorted list contains 5 of those 18 associated defensive measures. Thus, a security analyst can search for solutions amongst all the possible cybersecurity solutions and can still find problem-specific suggestions using our tool, quickly and efficiently.

6 Related Works

In D3FEND, the offensive techniques are associated with defensive technique based on some digital artifact relationships using some digital artifact objects (DAO). These associations are inferred, i.e., they are experimentally established from the DAOs. When security analysts are searching for countermeasures to particular problems, the D3FEND framework can be handy in terms of analytics. But as per as the discussion section in 1, this framework is not comprehensive. If a novel attack is happening, relevant countermeasures cannot be found easily, and thus we need a dynamic or robust framework. This motivates us to also crawl the web. However, crawling the web to associate defensive knowledge with offensive techniques inevitably leads to a lot of junk. Thus, it is also necessary to clean this data.

Various industry efforts have been carried out to provide threat sharing formats that can be applied by security professionals to share threat informatics. They include the Open Indicator of Compromise format (openIOC) [23], Structured Threat Information eXpression (STIX) [29], Trusted Automated Exchange of Intelligence Information (TAXII) [32], CVEs, and CWE. These formats leverage machine-readable formats to exchange threat indicators like the skill level of attackers involved in an attack, the tools used, the attack phases, and the attack tactics used. MITRE ATT&CK covers why and how attackers perform advanced persistent attacks. Attackers use a variety of approaches to achieve their end goal including deploying CVE vulnerabilities.

Some valuable resources have been built based on CVE vulnerabilities. One such example is the National Vulnerability Database (NVD) [6]. This was developed by the National Institute of Standards and Technology (NIST). Once a CVE is published to the list of CVEs, NVD is tasked with analyzing each CVE. The vulnerabilities are then categorized with a common weakness identifier (CWE). They are also given CVSS scores, which are metrics that characterize the exploitability and the impact of the vulnerability. When a malicious entity tries to launch an attack campaign against a system, they will often try to exploit vulnerabilities in the system such as those identified in CVE. Therefore, resources such as NVD can be used to establish associations between the CVE vulnerabilities and the type of offenses that typically exploit them. This is out of scope for our paper because we focus on associating the offenses to the countermeasures. However, if the links between CVE vulnerabilities and the offense techniques were to exist, it could be used in conjunction with our system to create associations between the CVE vulnerabilities and the countermeasures themselves, by using the offensive techniques as an intermediary. This is an interesting direction for future work.

In recent times, researchers have focused on techniques that automatically extract useful threat information from data available online from blogs and threat

report websites. Hutchins et al. [13] provided a technique to categorize advanced persistent threat attacks to kill chain phases. By classifying the attacker's actions into phases, defenders can comprehend attacker steps and seek to understand the attacker's motives.

A similar work, TTPDrill, by Ghaith et al. [12] applied NLP techniques to extract threat actors, threat indicators and generate STIX standard formatted reports from unstructured data. TTPDrill uses a simple lazy classification technique based on calculating similarity scores between two documents. TTPDrill focuses on extracting threat indicators from documents with short sentences of less than 900 words. Xiaojing et al. [15] applied NLP techniques to automatically extract indicators of compromise such as botnet IPs, malware names from unstructured text to a more standardized format. Burger et al. [7] classified various threat sharing technologies on how they interoperate. By considering the different uses of cases of the various threat sharing technologies, they propose a way to unify these techniques for wider usage and adoption by security professionals. Furthermore, some works have developed approaches to detect and recognize threats [11, 24, 25, 33, 35], as well as created benchmark datasets for advanced persistent threats [1]. Some work, such as [28], have proposed creating intelligent assistants for specific cyber-resources - their suggested architecture is based on using features such as boolean or non-boolean events, traffic and so on, to classify alerts. Furthermore, ontologies have been proposed earlier as a way to integrate all the different concepts in the security domain [3].

Lastly, we have also previously worked [5] on leveraging NLP techniques to extract attacker actions from threat report documents generated by different organizations and automatically classifying them into standardized tactics and techniques. This was based on MITRE data. All of the related work described here seeks to extract information or classify based on these information sources, but our proposed method is the first that attempts to establish connections without any preliminary knowledge (i.e. labeled data) between offensive and defensive techniques. To the best of our knowledge, there is no comprehensive study that attempts to establish these links across such varying types of data at all levels. Furthermore, the unstructured nature of the data includes new challenges that we will aim to tackle with advanced machine learning models.

7 Conclusion

Our study shows the effectiveness of language models in suggesting the right set of solutions to the security analyst. It is clear that traditional models cannot be used effectively for this task as those models cannot understand the context as well as the language models. Along with better natural language processing techniques, quality data is also required for making this knowledge association more comprehensive. Filling out the gaps within the framework to associate each of the offensive techniques with some countermeasures can be done if proper counteractive solutions and their descriptions exist in a structured way. To extend this work, we plan to build a large corpus of our own to accommodate all possible in-the-wild counteractive solutions. We also plan to automate the collection of such counteractive solutions for further association and automatically extract the part of text which talks about 'mitigation'.

Disclaimer Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

1. Dapt 2020 - constructing a benchmark dataset for advanced persistent threats. In: Wang, G., Ciptadi, A., Ahmadzadeh, A. (eds.) Deployable Machine Learning for Security Defense - 1st International Workshop, MLHat 2020, Proceedings. pp. 138–163. Communications in Computer and Information Science, Springer Science and Business Media Deutschland GmbH, Germany (2020). https://doi.org/10.1007/978-3-030-59621-7_8
2. Akamai: Bot manager: Product brief. https://www.akamai.com/resources/product-brief/bot-manager-product-brief (2021), [Online: Last accessed 9-March-2022]
3. Aviad, A., Wecel, K., Abramowicz, W.: The semantic approach to cyber security. towards ontology based body of knowledge. vol. 2015, pp. 328–336 (01 2015)
4. Ayoade, G., Akbar, K.A., Sahoo, P., Gao, Y., Agarwal, A., Jee, K., Khan, L., Singhal, A.: Evolving advanced persistent threat detection using provenance graph and metric learning. In: 2020 IEEE Conference on Communications and Network Security (CNS). pp. 1–9 (2020). https://doi.org/10.1109/CNS48642.2020.9162264
5. Ayoade, G., Chandra, S., Khan, L., Hamlen, K., Thuraisingham, B.: Automated threat report classification over multi-source data. In: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC). pp. 236–245 (2018). https://doi.org/10.1109/CIC.2018.00040
6. Booth, H., Rike, D., Witte, G.: The national vulnerability database (nvd): Overview (2013-12-18 2013), https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=915172
7. Burger, E.W., Goodman, M.D., Kampanakis, P., Zhu, K.A.: Taxonomy model for cyber threat intelligence information exchange technologies. In: Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security. p. 51–60. WISCS '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2663876.2663883, https://doi.org/10.1145/2663876.2663883
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
9. Face, H.: Robertae. https://huggingface.co/docs/transformers/model_doc/roberta (2019), [Online: Last accessed 26-March-2022]
10. GloVe: Global vectors for word representation. https://nlp.stanford.edu/projects/glove/ (2014), [Online: Last accessed 21-March-2022]
11. Han, X., Pasquier, T., Bates, A., Mickens, J., Seltzer, M.: Unicorn: Runtime provenance-based detector for advanced persistent threats. Proceedings 2020 Network and Distributed System Security Symposium (2020). https://doi.org/10.14722/ndss.2020.24046, http://dx.doi.org/10.14722/ndss.2020.24046
12. Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., Niu, X.: Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of

cti sources. In: Proceedings of the 33rd Annual Computer Security Applications Conference. p. 103–115. ACSAC 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3134600.3134646, https://doi.org/10.1145/3134600.3134646
13. Hutchins, E.M., Cloppert, M.J., Amin, R.M.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research **1**, 80 (2011)
14. Jibilian, I., Canales, K.: The us is readying sanctions against russia over the solarwinds cyber attack. here's a simple explanation of how the massive hack happened and why it's such a big deal. https://www.businessinsider.com/solarwinds-hack-explained-government-agencies-cyber-security-2020-12 (2021), [Online: Last accessed 13-April-2022]
15. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 755–766. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978315, https://doi.org/10.1145/2976749.2978315
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. ArXiv **abs/1907.11692** (2019)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR **abs/1310.4546** (2013), http://arxiv.org/abs/1310.4546
19. MITRE: Cve, https://cve.mitre.org/
20. MITRE: Enterprise matrix. https://attack.mitre.org/matrices/enterprise/ (2015-2021), [Online: Last accessed 10-March-2022]
21. MITRE: D3fend. https://d3fend.mitre.org (2021), [Online: Last accessed 10-March-2022]
22. OpenCV: Zero-shot learning : An introduction. https://learnopencv.com/zero-shot-learning-an-introduction/ (2020), [Online: Last accessed 13-March-2022]
23. OpenIOC: Open indicator of compromise. https://www.fireeye.com/blog/threat-research/2013/10/openioc-basics.html (2013), [Online: Last accessed 18-June-2021]
24. Oprea, A., Li, Z., Norris, R., Bowers, K.: Made: Security analytics for enterprise threat detection. In: Proceedings of the 34th Annual Computer Security Applications Conference. p. 124–136. ACSAC '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3274694.3274710, https://doi.org/10.1145/3274694.3274710
25. Pei, K., Gu, Z., Saltaformaggio, B., Ma, S., Wang, F., Zhang, Z., Si, L., Zhang, X., Xu, D.: Hercule: Attack story reconstruction via community discovery on correlated log graph. In: Proceedings of the 32nd Annual Conference on Computer Security Applications. p. 583–595. ACSAC '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2991079.2991122, https://doi.org/10.1145/2991079.2991122
26. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), http://www.aclweb.org/anthology/D14-1162

27. Sammut, C., Webb, G.I. (eds.): TF–IDF, pp. 986–987. Springer US, Boston, MA (2010). https://doi.org/10.1007/978-0-387-30164-8_832, https://doi.org/10.1007/978-0-387-30164-8_832
28. Sayan, C.M.: An intelligent security assistant for cyber security operations. In: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self Systems (FASW). pp. 375–376 (2017). https://doi.org/10.1109/FAS-W.2017.179
29. STIX: Structured threat information expression. https://oasis-open.github.io/cti-documentation (2021), [Online: Last accessed 18-June-2021]
30. Strom, B.E., Battaglia, J.A., Kemmerer, M.S., Kupersanin, W., Miller, D.P., Wampler, C., Whitley, S.M., Wolf, R.D.: Finding cyber threats with att&ck – based analytics (June 2017), https://www.mitre.org/publications/technical-papers/finding-cyber-threats-with-attck-based-analytics
31. superuser: How to stop a currently running cron job? https://superuser.com/questions/232144/how-to-stop-a-currently-running-cron-job/834110 (2011), [Online: Last accessed 9-March-2022]
32. TAXII: Trusted automated exchange of intelligence information. https://oasis-open.github.io/cti-documentation (2021), [Online: Last accessed 18-June-2021]
33. Zou, Q., Singhal, A., Sun, X., Liu, P.: Automatic recognition of advanced persistent threat tactics for enterprise security. In: Proceedings of the Sixth International Workshop on Security and Privacy Analytics. p. 43–52. IWSPA '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3375708.3380314, https://doi.org/10.1145/3375708.3380314
34. Zou, Q., Singhal, A., Sun, X., Liu, P.: Deep learning for detecting network attacks: An end to end approach. No. 12840, DBSec 2021: Data and Applications Security and Privacy XXXV, Virtual, US (2021-07-19 04:07:00 2021). https://doi.org/https://doi.org/10.1007/978-3-030-81242-3_13, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930878
35. Zou, Q., Sun, X., Liu, P., Singhal, A.: An approach for detection of advanced persistent threat attacks (53) (2020). https://doi.org/https://doi.org/10.1109/MC.2020.3021548