A Supervised Machine Learning Based Approach for Automatically Extracting High-Level Threat Intelligence from Unstructured Sources

Yumna Ghazi[*], Zahid Anwar[†], Rafia Mumtaz[‡], Shahzad Saleem[§] and Ali Tahir[¶]
Department of Computing, *SEECS, NUST*
Islamabad, Pakistan
[*]09bicseyghazi@seecs.edu.pk, [†]zahid.anwar@seecs.edu.pk, [‡]rafia.mumtaz@seecs.edu.pk,
[§]shahzad.saleem@seecs.edu.pk, [¶]ali.tahir@seecs.edu.pk

***Abstract*—The last few years have seen a radical shift in the cyber defense paradigm from reactive to proactive, and this change is marked by the steadily increasing trend of Cyber Threat Intelligence (CTI) sharing. Currently, there are numerous Open Source Intelligence (OSINT) sources providing periodically updated threat feeds that are fed into various analytical solutions. At this point, there is an excessive amount of data being produced from such sources, both structured (STIX, OpenIOC, etc.) as well as unstructured (blacklists, etc.). However, more often than not, the level of detail required for making informed security decisions is missing from threat feeds, since most indicators are atomic in nature, like IPs and hashes, which are usually rather volatile. These feeds distinctly lack strategic threat information, like attack patterns and techniques that truly represent the behavior of an attacker or an exploit. Moreover, there is a lot of duplication in threat information and no single place where one could explore the entirety of a threat, hence requiring hundreds of man hours for sifting through numerous sources — trying to discern signal from noise — to find all the credible information on a threat. We have made use of natural language processing to extract threat feeds from unstructured cyber threat information sources with approximately 70% precision, providing comprehensive threat reports in standards like STIX, which is a widely accepted industry standard that represents CTI. The automation of an otherwise tedious manual task would ensure the timely gathering and sharing of relevant CTI that would give organizations the edge to be able to proactively defend against known as well as unknown threats.**

***Index Terms*—Cyber Threat Intelligence; Natural Language Processing; Tactics, Techniques and Procedures (TTPs); STIX; Indicators of Compromise**

I. Introduction

In 2016, there was a massive wave of ransomware attacks on numerous hospitals across the US [1]. The ransomware called Locky [2] was introduced into the systems of healthcare professionals through a spearphishing campaign, where the emails had malicious attachments, e.g. Word documents with macros that caused the encryption of files on the victims' systems. A similar attack was seen in 2017, with WannaCry [3] targeting many hospitals that were a part of Britain's National Health Service. This attack had an identical infection vector, i.e. emails attached with malicious attachments or links. Also common was the fact that they both used Tor network for their command and control communication. These ransomwares moved laterally once they had infected one system to try and infect others in the network. There is a distinct overlap in the kill chains for these attacks, launched almost a year apart, and the former attack could have helped organizations prepare for the latter. Ironically, these are just two in thousands of instances where having knowledge of the previous attack could have been instrumental in circumventing the successive attacks, and yet, despite the information already being out there, organizations were unable to take advantage of that.

Cyber Threat Intelligence (CTI) is contextual knowledge about a threat that includes high-level indicators like campaign, motivation, Tactics, Techniques and Procedures (TTPs), as well as low-level indicators including IPs, hashes, network artifacts, etc. [4]. In recent years, it has become an indispensable part of day to day security operations, to help organizations prioritize threats and detect, mitigate or contain attacks in a timely manner. According to a recent report by SANS [5], almost 60% of organizations are already using CTI and 25% have plans to incorporate it into their security operations soon. It goes on to state that almost 47% of these organizations have dedicated teams for CTI that implement and monitor CTI.

Currently, there are numerous open sources providing periodically updated CTI, both structured (STIX [6], IOC [7], etc.) as well as unstructured (blacklists, etc.). The volume of intelligence gathered from open sources, also known as OSINT, is definitely on the rise. Hailataxii [8], a popular source for STIX-based threat feeds, that aggregates data from around ten other sources, boasts over 800,000 indicators. According to a Ponemon survey [9], 70% of the participants agree that they are overwhelmed by the sheer volume of the CTI data. A Gartner report corroborates this by declaring cyber security a big data problem [10]. Clearly, the issue now is not the lack of data; rather, there is too much data to be sifted through in order to find the needle in the haystack. Another problem with these sources of CTI is their sparse indicator categories that are mostly atomic in nature, including IPs, hashes and domains. There is a distinct lack of strategic threat information, like attack patterns and techniques that truly represent the behavior of an attacker or an exploit,

978-1-5386-9355-1/18/$31.00 ©2018 IEEE
DOI 10.1109/FIT.2018.00030

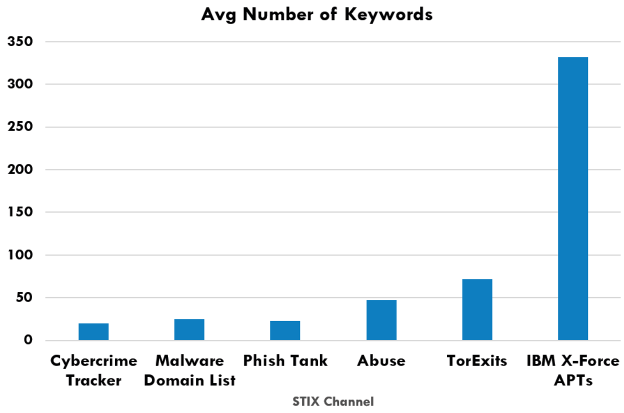


Fig. 1: Average number of keywords in well-known CTI sources, which is directly proportional to information quality

which is rated by 47% organizations as the most important information in CTI [5]. Figure 1 represents the quantity of relevant information on largely subscribed open threat sources, which directly correlates with the quality of CTI. X-Force [11] has the highest keyword density, and therefore, the best quality of CTI of the sources we have evaluated.

This lack of contextual and strategic information can be overcome by using blogs and articles that provide extensive information regarding attacks, including indicators of compromise, as well as high-level information like the attack patterns and the kill chain. However, extracting CTI from these sources manually is not possible when there is simply too much data and not enough time if the information is to be extracted in a timely manner. Automating the process is the only way to go about it, but it is not without its challenges. Another issue with this can be dealing with the signal to noise ratio: dealing with the actual instances of CTI and indicators against false positives, advertisements of security products and various other words that may be related to security but do not provide information about the attack. Moreover, there is a lot of redundancy in such texts, but also inconsistency if one source gives a different account of an attack, there is no way to truly determine which source is more credible. It is a consensus among industry professionals that automation and machine learning improve cyber resilience, whereas the lack of investment in such tools and technologies hinders it [12].

To overcome such challenges, we present our solution that uses tried and tested Natural Language Processing (NLP) techniques including Named Entity Recognition (Stanford NLP) [13] to first extract data from unstructured sources of Open Source Intelligence (OSINT) like blogs and articles, enriches if required and normalizes them into STIX for storage and sharing. For extracting CTI, we have used a supervised learning approach and annotated documents that contain relevant data taken from over 20 different sources. Conversion to STIX makes it readily consumable, so that it can be fed into various defensive and detective mechanisms to immediate effect. We also tag and index the data for easier retrieval.

Our solution is based on a microservice architecture, where every component or service is self-contained and independent in its functionality. We have developed a RESTful web service for performing annotation and for extracting terms from textual content, which include not only low-level indicators but also high-level information, particularly TTPs and attack patterns. We have also created a correlation and aggregation service that takes in the staged data extracted from numerous sources and correlates and aggregates them to remove redundancy and to group together information that pertains to the same attacks. The processed data is finally normalized to STIX and pushed into Elasticsearch and a pub-sub mechanism, TAXII, is being used to send feeds to subscribers.

The rest of the paper is structured as follows: In Section 2, we discuss the state-of-the-art in the CTI community, where the research is headed in terms of data-driven solutions for CTI, the open source tools that are being used, and the standards for representing CTI. Following that, we describe the architecture and implementation of our solution in Section 3. Section 4 is dedicated to the evaluation of our system. We conclude the paper with Section 5, where we discuss the merits of the solution and future work that can be derived from this.

II. Literature Review

CTI has been gaining traction in the research community as well, with increasing number of publications dedicated to the domain that focus on using machine learning and big data analytics to solve the identified issues. In [14], authors highlight the fact that cyber security has become a big data problem. Thuraisingham et al. [15] argue that the future of cyber security lies in embracing a data-driven approach to problem solving and categorize the challenges broadly as attack detection and mitigation, data trustworthiness and policy-based sharing and risk-oriented security metrics. The paper further discusses the state of the art and the future directions in each category. Harel et al. [16] describe the growing trend of intelligent systems in cyber security in terms of automation, collaboration, and how the industry has been equipping itself for the fight against proliferating attackers. Authors specifically discuss supervised and unsupervised learning and give examples of how they are being used in determining the risk factor of a particular user using behavior analysis and anomaly detection, respectively. A data-driven approach to CTI is described in [17], where the authors experiment with using the vulnerability exploitation data found on Twitter to create a model for predicting future exploits. Another such work related to this is presented in [18], which goes further in improving the predictive model, especially by being meticulous about the training dataset.

While the aforementioned studies definitely highlight the recent hype about machine learning and cyber security, they do not resemble our work in terms of the goals. Liao and Beyah et al. [19], however, have similar goals: they have developed a tool for automatic extraction and retrieval of CTI from online sources. To that end, they have developed a vocabulary of context terms, based on OpenIOC's Indicators of Compromise (IOC), that make it easier to predict the presence of an IOC in a sentence, and use a combination of regexes, graph mining and NLP techniques including dependency parsing, topic filtering and content term extraction to extract information from blogs and other online sources to generate an IOC file, as per OpenIOC schema. While their approach is indeed novel, their focus remains solely on providing context to IOCs that remain primarily atomic and not high-level strategic aspects of an attack, like TTPs or phases of the kill chain, that we are focusing on.

A. *CTI Tools: Aggregation and Exchange*

Over the last few years, a lot of tools have been created that deal with different formats of CTI to collect, store and exchange them. A few honorable mentions include CTX/Soltra Edge [20], Collaborative Research into Threats (CRITS) [21], Collective Intelligence Frameworks (CIF) [22], Malware Information Sharing Platform (MISP) [23], Facebooks ThreatExchange [24], IBMs X-Force [11] and ThreatConnect [25].

There are other tools as well that focus primarily on the parsing and analysis of CTI, and that are more relevant to what we are proposing. ActorTrackr [26], for instance, is one such tool that stores and links APT actors information and it includes other information about the actual exploits executed by said actors. It relies on users and certain repositories for the data insertion, and does not automate the process. Automater [27] is an analysis tool that includes support only for atomic indicators like IP addresses, hashes and URLs. A Google powered APT Groups, Operations and Malware Search Engine [28], which fetches threat information matching given keywords from certain specified sources, but it will only retrieve pertinent links. Cacador [29], Forager [30] and Jager [31] are tools that take in textual reports about incidents and retrieve indicators from them, and while these are the most similar in terms of what our solution does, they do not cater to the more important high-level TTPs that we want to extract from text.

III. Architecture and Implementation

In this section, we present an in-depth description of the architecture and implementation of our solution. Since the primary focus is the NLP-based learning of an NER model for extracting CTI from textual content, this section is further separated into two subsections where the former explains the process and design choices in learning and generating the model, whereas the latter describes the production system that uses this model to extract relevant information and processes it before storing and disseminating.

A. *Learning to Extract CTI*

One of the main objectives of our research was to extract high-level CTI indicators from unstructured sources. To that end, we have used a Natural Language Processing (NLP) technique called Named Entity Recognition (NER). Named Entity Recognition is essentially the process of extracting named entities from text. Most NLP libraries have default named entity classifiers that are trained to recognize generic classes, like names, locations, organizations. NER is incredibly domain-specific, in that a model trained for one domain will definitely not be suitable to use for another. To the best of our knowledge, none of the proof-of-concepts implemented so far in this domain have used and created a sequential model trained specifically for CTI. The output of this process is the trained model, which is then fed into the main architecture for further processing as can be seen in Figure 2.

1) Collecting CTI Documents: We collected a number of CTI documents containing relevant information about cyber threats as candidate documents for our training set from well-known security blogs and threat reports from 4 different sources that include FireEye [32], Kaskpersky Security Lab [33], and a curated list of APT reports from a Github repository [34]. The criteria for choosing these sources was based on: i) the frequency with which they post incident reports, ii) the thorough research they are known to do on threats, and iii) their reputation in the information security industry based on their well-known detective and defensive mechanisms.

2) Annotating CTI Documents: For an NLP model to learn to recognize the required terms, we need to provide it with supervised data that contains continuous text labeled with the classes that need to be recognized. To prepare such a dataset, we created a web-based user interface that would make the manual task of annotation easier. We also devised a set of terms that we would like the model to be able to recognize. For this purpose, we took into account the vast vocabulary of STIX and narrowed down the main building blocks of TTPs — victim targeting, resources, behavior and intended effect — to generate a focused dataset that includes:

1) **Actor**: The team of hackers attributed to a certain attack, e.g. Carbanak cybergang, APT30, etc.
2) **Targeted Industry**: The industry targeted by the attack, e.g. financial institutions, government, military, etc.
3) **Targeted Location**: The specific geographical target of the attack, e.g. South Asia, Turkey, US, etc.
4) **Intended Effect**: The goal of the attacker, e.g. cyber espionage, financial gain, steal information, etc.
5) **Technique (TTPs)**: The high-level techniques employed by the attacker, e.g. spearphishing emails, social engineering, watering hole, etc.
6) **Tool Used**: The tools used by the attacker, e.g. backdoor, reverse shell, Mimikatz, etc.
7) **Targeted Application**: The applications whose vulnerabilities the attackers wish to exploit, e.g. MS Word, PowerShell, etc.

This list also included other, low-level indicators, that were parsed using regexes, as they always follow certain patterns: *IP Addresses*, *Hashes*, *Domain Names*, *URLs*, *Registry Keys*, *Files*, and *Vulnerabilities*.

From the user interface, the annotation is made by selecting a group of words and allocating the corresponding tag to it, as shown in Figure 3. Once the annotated document is saved, the backend service ensures that the document is readied to

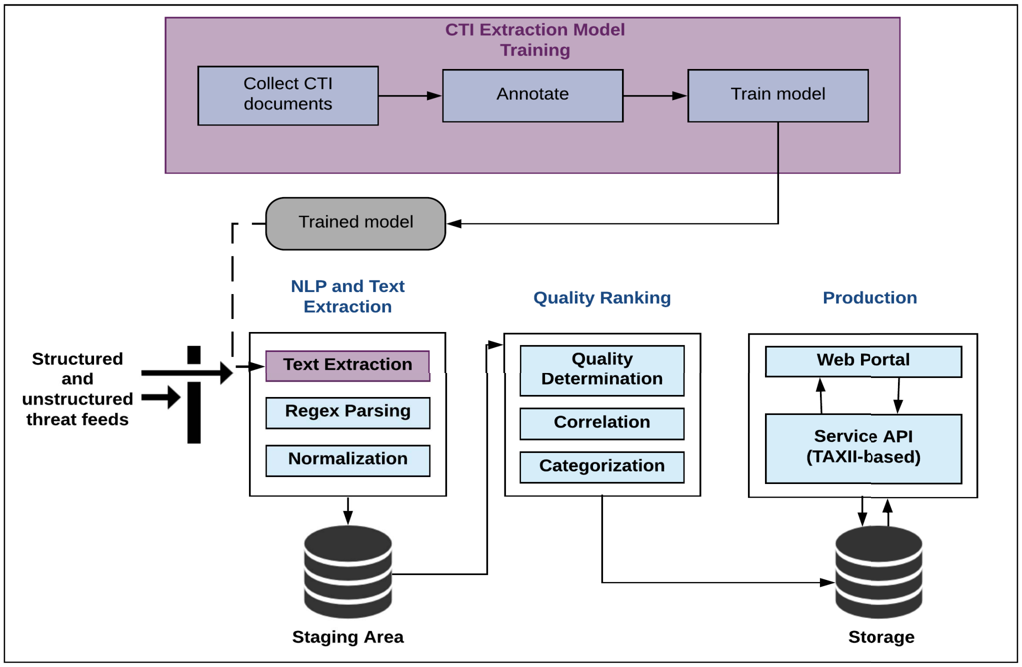


Fig. 2: Architecture of the solution

become a part of the training set. It first filters the incoming text to ensure that it does not contain any unicode characters. The next step in the process is to tokenize sentences, which returns a list of sentences in the given text. In a loop over the sentences, each sentence is tokenized such that every word is a separate token. The default tokenizer had to be customized to make sure it would tokenize registry keys and IP addresses, and other such indicators that contain special characters, properly. Then the tokenized terms are POS-tagged, which essentially means that the tokens are tagged as per the part of speech (POS) they represent in the sentence, eg. NN for noun, VB for verb, etc. For every tokenized word, there is a tag attached to it which tells the algorithm whether or not it needs to be recognized.

3) Training the CTI NER Model: We have chosen Stanford NLP not only because it is one of the most popular NLP libraries, but also because it has a powerful and well-tested CRF Classifier. We have used the Conditional Random Fields (CRFs) algorithm, which is a well-known method for pattern recognition. CRFs are used a lot in NLP because they take into account the context and produce sequential models. In other words, linear chain CRF predicts a label for a sample while taking into account the labels for neighboring samples, which is extremely important in NER because the label for the previous sample would help determine the label for the next one. To train the model, we provide the annotated training set and run it through the CRF Classifier algorithm. The resultant model is fed into the NLP component of the production system, which is used to extract the CTI terms.

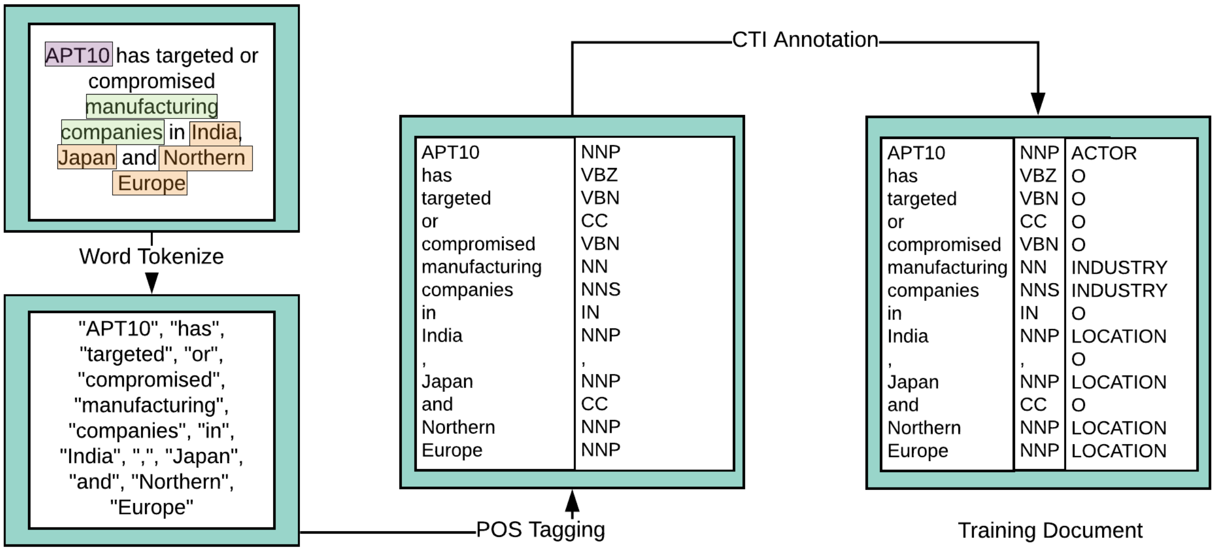
B. The Production System

Our production system is separated into three main components: the NLP component, the quality ranking component, and the production component, represented in Figure 2. We will describe them all in this subsection.

1) The Natural Language Processing Component: This component is responsible for making sense of the textual data and extracting CTI from the text based on the model. It also consists of a regex parsing portion for the CTI that we extract using custom regex patterns for IPs, hashes, etc. And finally we normalize that information into STIX format and store it for further processing.

2) Quality Ranking Component: Having extracted the data and normalized it, this component deduplicates the information we already have to remove redundancies and to aggregate the data regarding one specific attack and/or campaign. It is also responsible for ranking the sources, which is extremely important from the consumer's perspective. It will essentially allow consumers to see which sources are producing the latest data and how good the quality of the produced data is, based on the signal-to-noise ratio that we measure by counting the total words in a text and how many of those words are labelled as words from our vocabulary, as shown in 1. We finally tag the data and send it off to be indexed into Elasticsearch for easy retrieval.

$$Quality = 100 * (Labelled\ words\ /Total\ words) \quad (1)$$


APT10 has targeted or compromised manufacturing companies in India, Japan and Northern Europe
Word Tokenize
"APT10", "has", "targeted", "or", "compromised", "manufacturing", "companies", "in", "India", ",", "Japan", "and", "Northern", "Europe"
POS Tagging
CTI Annotation
Training Document

APT10	NNP
has	VBZ
targeted	VBN
or	CC
compromised	VBN
manufacturing	NN
companies	NNS
in	IN
India	NNP
,	,
Japan	NNP
and	CC
Northern	NNP
Europe	NNP

APT10	NNP	ACTOR
has	VBZ	O
targeted	VBN	O
or	CC	O
compromised	VBN	O
manufacturing	NN	INDUSTRY
companies	NNS	INDUSTRY
in	IN	O
India	NNP	LOCATION
,	,	O
Japan	NNP	LOCATION
and	CC	O
Northern	NNP	LOCATION
Europe	NNP	LOCATION

Fig. 3: Annotation Process Example

3) The Production Component: The production component deals with the dissemination of CTI information. Not only do we expose an API that will enable organizations to import the information and use it as required, we also provide a TAXII-based publisher-subscriber model so that STIX can be imported into any defensive or detective tool that supports TAXII and take advantage of the timely and effective intelligence.

IV. Evaluation

The main objective of the research is to be able to extract high-level indicators from textual content using a combination of NLP and machine learning. In information retrieval and extractions systems, evaluation is carried out by using Precision and recall method, which determines how relevant the information retrieved by the model actually is.

Precision, Recall and F-measure are defined by true positives, false positives and false negatives. Accurate labeling is a *True Positive (TP)*, inaccurate labeling is a *False Positive (FP)* and a missed label is a *False Negative (FN)*. Precision (P) is the number of TPs out of the sum of TPs and FPs. Recall (R) represents the number of TPs out of the total number of TPs and FNs. Whereas the F-measure (F1) or F1 score is a combination of precision and recall. Precision and Recall for a model may be high, despite the quality of the model; therefore, F1 is the representation of the overall quality of the model's performance.

We initially gathered 50 documents, the details of which are mentioned in Section III-A; 10% of these files were annotated for testing, while the other 90% were used for training. We have 7 major classes of text that we annotated and we wanted to recognize them from raw text, and we have evaluated the performance of the model against each of them, as can be seen in Table I.

TABLE I: Evaluation Results

Entity	P	R	F1
ACTOR	1	0.33	0.5
APP	0.33	0.25	0.29
EFFECT	1	0.71	0.83
INDUSTRY	1	0.67	0.8
LOCATION	0.5	1	0.67
TECHNIQUE	0.58	0.54	0.56
TOOL	0.5	0.5	0.5
Totals	**0.69**	**0.56**	**0.62**

We tested the trained model in two rounds, where we initially took 10% of the annotated documents as test documents, represented by blue and later moved the ratio to 15% by adding documents to the training set, as shown in red in Figure 4. We only show here the F1 score, which is a reliable evaluator of a classifier that takes into account both the precision and recall. While the number of TPs was satisfyingly high, the number of FPs was also higher than expected. So while the model has satisfactory precision, the recall is not as good as it could have been. The reason is that the confusion between certain terms is high — like Targeted Applications and Tools Used — and it is difficult for the model to have been adequately knowledgeable with the relatively small size of our training set compared to the thousands of documents it takes to perfect a model like Stanford's. Regardless, these results are sufficient to prove our hypothesis that it is indeed possible to extract TTPs using NER.

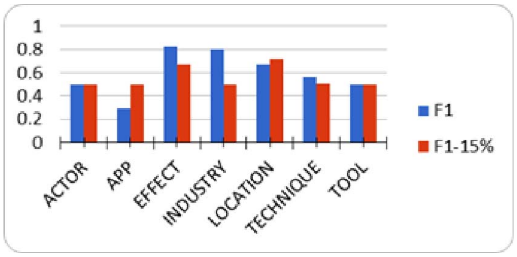


Fig. 4: F1 comparison

V. Conclusion and Future Work

The focus of this research has been primarily on extracting cyber threat intelligence concepts from textual data sources and using them to perform correlation and quality analysis. In order to do so, we trained a supervised model over a data corpus of unstructured texts from well-reputed security blogs. This would provide security analysts with enough context to make them capable of inferring relationships between seemingly disconnected cyber attacks and attributing attackers by similarity in strategies. Since this work is accompanied by stable code and the supervised model is open sourced [35], the work is not only reproducible but also extendable. The annotation tool can actually be used by security analysts all over the world so that they can help improve the precision and recall of the model. It can be used as an open portal for users to load threat intelligence documents where the system does a real time scan of the document and returns the appropriate tagged information in a structured format, reducing manual labor and allowing cyber security professionals to optimally configure security tools and ultimately provide optimal defense.

References

[1] "Massive locky ransomware attacks hit u.s. hospitals — healthcare it news," http://www.healthcareitnews.com/news/massive-locky-ransomware-attacks-hit-us-hospitals, (Accessed on 05/28/2017).

[2] "A closer look at the locky ransomware," https://blog.avast.com/a-closer-look-at-the-locky-ransomware, (Accessed on 05/28/2017).

[3] "Ransomware outbreak: Wannacry — redsocks security," https://www.redsocks.eu/news/ransomware-wannacry/, (Accessed on 05/28/2017).

[4] "Definition: Threat intelligence," https://www.gartner.com/doc/24872, (Accessed on 05/28/2017).

[5] D. Shackleford, "Cyber threat intelligence uses, successes and failures: The sans 2017 cti survey," SANS, Tech. Rep., 2017.

[6] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (stix)," *MITRE Corporation*, vol. 11, pp. 1–22, 2012.

[7] C. Harrington, "Sharing indicators of compromise: An overview of standards and formats," *EMC Critical Incident Response Center*, 2013.

[8] "hail a taxii," http://hailataxii.com/, (Accessed on 05/21/2017).

[9] *The Value of Threat Intelligence: The Second Annual Study of North American United Kingdom Companies*, 2017.

[10] "Information security is becoming a big data analytics problem," Mar 2012. [Online]. Available: https://www.gartner.com/doc/1960615/information-security-big-data-analytics

[11] "Ibm x-force exchange," https://exchange.xforce.ibmcloud.com/, (Accessed on 05/21/2017).

[12] "The third annual study on the cyber resilient organization," Mar 2018.

[13] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[14] M. Kantarcioglu and B. Xi, "Adversarial data mining: Big data meets cyber security," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1866–1867.

[15] B. Thuraisingham, M. Kantarcioglu, K. Hamlen, L. Khan, T. Finin, A. Joshi, T. Oates, and E. Bertino, "A data driven approach for the science of cyber security: Challenges and directions," in *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*. IEEE, 2016, pp. 1–10.

[16] Y. Harel, I. B. Gal, and Y. Elovici, "Cyber security and the role of intelligent systems in addressing its challenges," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 4, p. 49, 2017.

[17] C. Sabottke, O. Suciu, and T. Dumitras, "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits." in *USENIX Security*, vol. 15, 2015.

[18] B. L. Bullough, A. K. Yanchenko, C. L. Smith, and J. R. Zipkin, "Predicting exploitation of disclosed software vulnerabilities using open-source data," in *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*. ACM, 2017, pp. 45–53.

[19] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "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*. ACM, 2016, pp. 755–766.

[20] "Soltra edge." [Online]. Available: https://soltra.com/

[21] "Crits - of the community. by the community. for the community." [Online]. Available: https://crits.github.io/

[22] "Collective intelligence framework," http://csirtgadgets.org/collective-intelligence-framework/, (Accessed on 05/21/2017).

[23] "Misp - malware information sharing platform and threat sharing - open source tip," http://www.misp-project.org/, (Accessed on 05/21/2017).

[24] "Threat exchange - facebook for developers," https://developers.facebook.com/products/threat-exchange, (Accessed on 05/21/2017).

[25] "Threatconnect — security operations and analytics platform," https://www.threatconnect.com/, (Accessed on 05/21/2017).

[26] 2018. [Online]. Available: http://actortrackr.com/

[27] 1aN0rmus, "1an0rmus/tekdefense-automater," May 2016. [Online]. Available: https://github.com/1aN0rmus/TekDefense-Automater

[28] [Online]. Available: https://cse.google.com/cse/publicurl?cx=003248445720253387346:t

[29] Sroberts, "sroberts/cacador," Oct 2017. [Online]. Available: https://github.com/sroberts/cacador

[30] 2018. [Online]. Available: https://github.com/byt3smith/Forager

[31] 2018. [Online]. Available: https://github.com/sroberts/jager

[32] S. Dubey, N. Kirk, S. Miller, M. Berninger, and D. Pany, "Threat research," Jun 2018. [Online]. Available: https://www.fireeye.com/blog/threat-research.html

[33] A. V. Ivanov, A. Shadrin, A. Nikishin, V. Bogdanov, D. Legezo, S. Lurye, B. Stepanov, M. Vergelis, A. Kostin, D. Makrushin, and et al., "Securelist - english - global," Jun 2018. [Online]. Available: https://securelist.com/

[34] Kbandla, "kbandla/aptnotes," Jun 2017. [Online]. Available: https://github.com/kbandla/aptnotes/

[35] Yghazi, "yghazi/g4ti-nlp-processor," May 2017. [Online]. Available: https://github.com/yghazi/g4ti-nlp-processor